A GUIDE TO LEARNING MODULES IN A DYNAMIC NETWORK

Karthik Raghavan Ramaswamy

A GUIDE TO LEARNING MODULES IN A DYNAMIC NETWORK

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Eindhoven,
op gezag van de Rector Magnificus prof.dr.ir. F.P.T. Baaijens,
voor een commissie aangewezen door het College voor Promoties,
in het openbaar te verdedigen op

dinsdag 3 mei 2022 om 16:00 uur

door

Karthik Raghavan Ramaswamy

geboren te Chennai, India

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommisie is als volgt:

Voorzitter:	prof. dr. ir. A. M. J. Koonen
Promotoren:	prof. dr. ir. P. M. J. Van den Hof
	prof. dr. S. Weiland
Copromotor:	dr. G. Bottegal (ASML)
Leden:	prof. dr. H. Hjalmarsson (KTH Royal Institute of Technology)
	prof. dr. D. Materassi (University of Minnesota)
	prof. dr. ir. T. A. E. Oomen
	dr. A. G. Dankers (University of Calgary)

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

A guide to learning modules in a dynamic network

Karthik Raghavan Ramaswamy

This project has received funding from the European Research Council (ERC), Advanced Research Grant SYSDYNET, under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 694504).

This dissertation has been completed in fulfillment of the requirements of the Dutch Institute of Systems and Control (DISC) for graduate study.

A catalogue record is available from the Eindhoven University of Technology Library.
ISBN: 978-90-386-5500-0

Copyright © 2022 by Karthik Raghavan Ramaswamy.

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilised in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the copyright owner.

Cover design by Adithi Narayanaswamy
Printed by Ridderprint | www.ridderprint.nl

Dedicated to nature, time, love
and my family

Summary

A guide to learning modules in a dynamic network

Karthik Raghavan Ramaswamy

Complex interconnected systems are becoming increasingly ubiquitous, and data-driven model learning problems of large-scale interconnected systems, known as dynamic networks, are expected to become of paramount importance in different fields like robotics, smart grids, transportation systems, oil and gas reservoirs, autonomous vehicle platooning, biological systems. These networks can be considered as a set of measurable signals (the node signals) interconnected through dynamic systems and can be possibly driven by external excitation signals. The task is to learn a mathematical model of these dynamic systems, called modules, from measured signals (data). Either learning all the modules in the network or a subset of modules in the network, the task can be broken down into subproblems of estimating a single module embedded in a dynamic network (local module identification).

Typically, existing identification methods to identify modules in a dynamic network require restrictive assumptions to hold. For example, a typical standing assumption is that the non-measurable excitation signals (noise) entering the nodes of the dynamic network are uncorrelated with each other. However, in many situations, the noises can be correlated. In this situation, considering the assumption of uncorrelated noise in the identification procedure leads to an inaccurate model. Moreover, existing methods usually require the availability of specific measured nodes and certain sets of externally excited nodes in order to ensure an accurate model. However, in practical situations, these requirements cannot be met, and we need identification methods to deal with flexibility in selecting measured and excited node signals. Another important aspect is the complexity required to identify a module in large networks. Therefore, it is fundamental to develop effective scalable algorithms to address it. This thesis addresses the above problems, and a step-by-step guide to learning modules in a dynamic network is provided for a user.

In many practical cases, it is possible to have dynamic networks with disturbance signals that are correlated across measured nodes. In this situation, it is necessary to consider also the disturbance correlation structure during the estimation procedure. Also, the so-called confounding variables that lead to a lack of consistency need to be dealt with. A Local Direct Method (LDM) that provides asymptotically efficient estimates is developed to tackle this. It involves a constructive procedure for signal selection that tackles the effect of confounding variables and builds a MIMO identification setup that guarantees asymptotically efficient estimates when applying the LDM. All in all, a generally applicable theory is provided for the LDM that is independent of the particular node selection scheme selected.

Different sets of conditions are available in the literature for the set of node signals to be measured and the set of excitation signals needed to identify a module. These conditions have been derived from either an indirect identification approach, considering external excitation signals as inputs, or a direct identification approach, considering measured node signals as inputs. While both approaches lead to different sets of (sufficient) conditions, we extend the flexibility in the sufficient conditions for the selection of excitation and measured node signals by combining direct and indirect approaches. As a result, we introduce a Generalized method that offers flexibility in sensor selection and actuation requirements.

In order to achieve consistent estimates using the LDM or generalized method, data informativity conditions need to be satisfied. However, these conditions typically cannot be directly used by the experimenter since they rely on checking the positive definiteness condition on a spectrum of internal signals. Therefore, these conditions have been translated to path-based conditions that depend on paths from external signals to the internal node signals. The experimenter can now check the satisfaction of these path-based conditions using graphical algorithms and can easily verify and ensure data informativity.

A MISO or MIMO identification setup needs to be considered in a dynamic network setting to identify a single module. This leads to the estimation of a large number of parameters that are of no interest to the experimenter and requires model order selection for all the modules in the identification setup. While the former task poses the problem of estimating a large number of parameters that are of no interest to the experimenter, the latter task may result in computationally challenges in large-size networks. Regularized kernel-based methods are used to avoid these issues and increase the accuracy of the identified module of interest. An Empirical Bayes Direct Method (EBDM) is developed where the modules that are of no interest in the identification setup are modeled as zero-mean Gaussian processes with covariance matrix (kernel) given by the first-order stable spline kernel, thereby represented using only two hyperparameters for each module. Also, combining the approach with approximate inference methods, the situation of missing node measurements is handled by employing a Markov-chain Monte Carlo technique to reconstruct the unknown missing node measurements and the network dynamics.

Finally, the identification approaches require prior knowledge of network and noise topology. A new approach that incorporates the estimation of this prior information into the identification, leading to a fully data-driven approach for estimating the dynamics of a local module, is presented. The developed algorithm uses a non-causal Wiener filtering technique that involves a series of analytical solutions with parallel computation capabilities to estimate the topology. The regularized kernel-based method with attractive statistical properties and scalable to handle a larger-scale network is then employed to estimate the target module. As an alternative, a scalable multi-step least-squares algorithm that admits only explicit solutions for topology and module estimation is introduced, making it computationally efficient. Consistency proof with path-based data informativity conditions, which can indicate where excitation signals must be allocated, is also formulated.

Contents

Appendices

Appendices 227

Appendices 255

1 Chapter

Introduction

The advancements in technology have made real-life systems increasingly complex and interconnected. Large-scale interconnected systems, known as dynamic networks, are becoming increasingly ubiquitous. The data-driven model learning problem of these systems has become of paramount importance in various domains. Despite the availability of several identification techniques for dynamic networks developed over the past decade, many open problems still need to be solved. The available identification methods are applicable only for dynamic networks that satisfy restrictive conditions/assumptions that are mostly impractical. Also, as the size of the network grows, the complexity of applying these methods increases. This thesis provides approaches to identify modules (subsystems) embedded in a dynamic network by relaxing the restrictions and reducing the complexity. This objective involves exploiting the freedom in the signal selection and actuation schemes, reconstructing missing signals, handling networks with correlated and rank-reduced noise, translating the data-informativity condition to a topology-based condition, handling the unavailability of topology information, and reducing the complexity with algorithms using kernel-based methods and multi-step least squares method. As a result, we aim to guide an experimenter and provide the experimenter with the needed tools to identify a module embedded in a dynamic network accurately. The first chapter summarizes the motivation and challenges in identifying modules embedded in a dynamic network and surveys the problems considered in this thesis.

1.1 Motivation

The field of systems and control deals with the analysis, modeling, and control of systems. According to the definition from Oxford languages," a system is a set of things working together as parts of a mechanism or an interconnecting network; a complex whole." In an abstract sense, a system is an object in which variables of different kinds interact and produce observable signals [77]. Many engineering and physical systems can have *dynamics*, i.e., the present behavior of a system can affect its future behavior. For example, the acceleration of a body in the past and present can affect its speed and position in the future. As considered in the field of *system identification*, these systems are called *dynamical systems*, i.e., systems that have the memory of the past.

Due to the advancements in technology, systems in real life are becoming more complex and large-scale. Examples can be found in various domains, such as biology [34, 88], Telecommunication systems [96], Economics [81], Brain and cognitive sciences [46, 97], Power systems [24, 75], Autonomous vehicles [108], Geology [5], Windfarms [69, 121]. Large-scale complex systems have become ubiquitous and a part of our day-to-day life. These large-scale complex systems can be described as interconnected networks of simpler subsystems. This representation makes it important for us to model these interconnected systems to study, analyze, monitor, and control the large-scale complex systems. For example, with the uprising of smart grids (see figure 1.1), renewable energy sources, increasing power demand, and modern technologies like plug-in electric vehicles, there will be a significant change in the power system. Therefore it will be quintessential to model the interconnected systems for analysis and better control and optimization of power grids. Similarly, it is necessary to model the causal influences among neuronal populations in brain networks for estimating the effective connectivity between different brain areas. Based on [148], any system is an interconnected system that consists of interacting simpler subsystems modeled by *tearing, zooming* and *linking*. Therefore, by modeling the interconnected system, we can understand any system, like which subsystems are interconnected with other subsystems and how they interact. Also, we can focus our modeling efforts on particular subsystems.

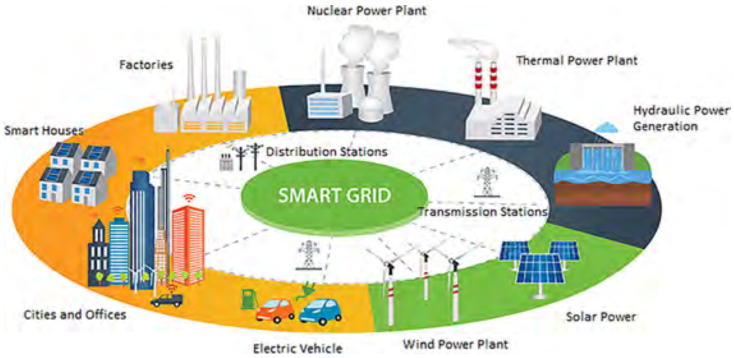


Figure 1.1: Smart grids (Eolas Magazine)

Also, due to the growth of sensing technology, sensors are becoming cheaper with improved quality and accuracy. In addition, they can measure different variables like pressure, flow, temperature, voltage, current, phase, frequency. The ability to measure many variables and improvement in sensing technology has increased the possibility of measuring various relevant process variables in many complex systems. For example, sensors can now be placed between the reservoir and the wellbore in reservoir engineering to measure quantities like pressure and flow rate. Similarly, many piezoelectric elements can be placed underneath the wafer to measure the deformations at different points in wafers during the exposure phase of lithography. Thus, the current technological advancements have led to the availability of lots of data for many interrelated variables. This data availability for many interrelated variables for complex systems paves the way for modeling interconnected systems through data.

1.2 Need for models

A model describes the relationship between the variables interacting with a system. It is an abstraction of reality that tries to capture the essential aspects. Therefore the aim is to describe the essential behavior and discard the irrelevant features. What is essential and what is irrelevant depends on the intended use of the model [77].

Models can be of many forms, namely *mental models, graphical models, physical models* or *mathematical (or analytical) models*. Many advanced applications in engineering and physics use these mathematical models, which describe the relationship among the system variables using mathematical equations like differential or difference equations. Mathematical models can further be classified using many attributes like *static or dynamic, linear or non-linear, deterministic or stochastic, lumped or distributed*. *Dynamic models* are models which describe the behavior of a dynamical system over time. Time plays an important factor in these models. The field of control theory depends on these *dynamic models* to control the behavior of the system over time using controllers. Using mathematical models, one can gain physical insights into the system. Many fields use mathematical models for various purposes. Below we list some of the purposes: [77]

1. *Prediction:* A model of a system can be used to predict the future behavior of a system. For example, predicting the stock price in stock markets or predicting weather conditions in a particular region.

2. *Simulation:* A model can be used to simulate the behavior of a system under different operating conditions and experimental conditions. For example, simulators using reservoir models are used by most hydrocarbon industries for many insights like designing the operations of the hydrocarbon extraction process to get maximum productivity [106].

3. *System design:* A model can be used to analyze a system and design it to achieve the desired performance. For example, system re-designing can be done using the model of the system such that the desired behavior is achieved. Another example is the design of a controller for a system based on its model.

4. *Monitoring:* A model can be used to estimate an unmeasured variable. For example, a State-of-Charge (SOC) estimator can be built for Battery Management System (BMS) using voltage and current measurements.

5. *Diagnosis:* A model can be used for fault detection. The model's behavior can be compared and checked for any deviation from the system's behavior to detect any fault and the location of the fault. For example, in power or water, or gas distribution systems, any fault or leak and its location can be detected.

The classical methods in system identification literature deal with relatively simple configurations like open-loop and closed-loop systems [77, 95, 120]. However, in this thesis, we move beyond these configurations and deal with increasingly complex systems, which typically consist of many subsystems interacting with each other. Consequently, these networked systems are large-scale in nature. Moreover, they have complex spatial interconnectivity, requiring local and global models and tools for local and global decision-making. These networked systems, known as *dynamic networks*, require dedicated modeling tools that can tackle their complexities.

1.3 Dynamic network model

As already discussed, any system we encounter in our day-to-day life is nothing but an interconnected network of simple subsystems. Take, for example, the recent trend and enabling technology of current engineering innovations, the *Cyber-physical systems*. They are interconnected systems at various layers as seen in figure 1.2. The physical layer consists of subsystems linked together by physical interactions and the cyber layer with the networked controllers. The controllers interact with each other and with the systems in the physical layer. This representation is similar to the distributed [25] or decentralized [67] control scheme used to optimize and control large-scale systems like industrial plants or

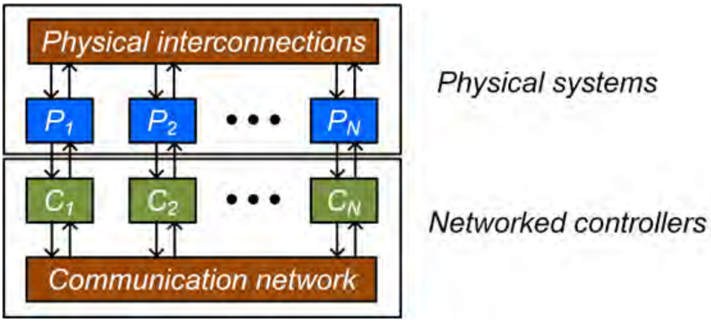


Figure 1.2: Cyber-physical system as interconnections of several interaction systems at different layers.

power plants, with the exception that in distributed control, the controllers will not communicate with each other. Here, local subsystems are controlled using local controllers. It is important to know which subsystems are connected to other subsystems and their interactions to design the control scheme and decide which controller communicates with which controllers to attain the desired performance. Therefore, to study, analyze, monitor, control or make decisions on these systems, it is essential to know about two attributes - how subsystems are interconnected and how they interact. Therefore, a *dynamic network model* that encompasses both these attributes and represents the interconnected systems is required.

Generally, a dynamic network model has two components - a graphical representation and a set of mathematical models. The graphical representation (see figure 1.3) contains the vertices (nodes) connected by directed or undirected edges (links). It encodes the information about the interconnection structure of the subsystems. Various network models are available in the literature like probabilistic graphical models, behavioral models, state-space models, Structural Equation Models (SEM). A brief overview of different network models is provided in [146] and [115]. Depending on the network model, the nodes and edges in the graph can represent different objects. The mathematical model that describes the interactions also changes based on the network model. For example, the vertices can represent a random variable, time-series, or subsystem dynamics based on the model. In probabilistic graphical models [73], the nodes represent the random variables and the graph structure using links incorporates the conditional independence information of the joint distribution of all the random variables. In state-space models where the system's behavior is represented by a first-order differential or difference equation, the nodes represent the states and the relationship between states are captured through weighted links in the graph.

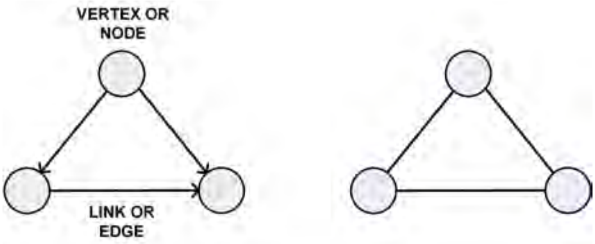


Figure 1.3: Graphical representation of a dynamic network with directed edges (left) and undirected edges (right).

1.3.1 Module networks

Another type of dynamic network model is the *module network*. It is an extension of the classical closed-loop architecture in the field of systems and control. In this network model, a vertex or node represents a time series, and a directed link (edge) represents the dynamics between nodes.

The *module network* consists of the following (see figure 1.4):

1. **Nodes or internal variables:** These represent the time series of quantities that can possibly be measured using measurement devices or sensors. Examples are voltage, current, pressure, flow rate, Blood Oxygenation Level Dependent (BOLD) signal from functional magnetic resonance imaging (fMRI), position or distance, velocity, force, temperature. In figure 1.4, these are represented by signals $w_1, \ldots, w_8$.

2. **Modules:** The internal variables are dynamically related to one another in the network. This relation is represented by the modules that describe the dynamic behavior, which describes the interconnection and interaction between different nodes. For example, modules can represent the dynamic behavior of different systems in power grids (like load, generator, controller), dynamic behavior of different self-driving vehicles in cooperative vehicles. In figure 1.4, these are represented in boxes like $G_{21}, G_{32}, \ldots$.

3. **Excitations:** These represent the *measured external variables* that can be directly manipulated and affects the internal variables. For example, the velocity of a leader in vehicle platooning that can be manipulated by an operator, the activation of a test subject's brain using music or sleep. In figure 1.4, these are represented by signals r_1, r_4, r_5, r_8.

4. **Disturbances:** These represent the *unmeasured external variables* that affects the internal variables. For example, the wind affecting different wind turbines in a wind farm, the thermal noise in a electric circuit. In figure 1.4, these are represented by signals $v_1, \ldots, v_8$.

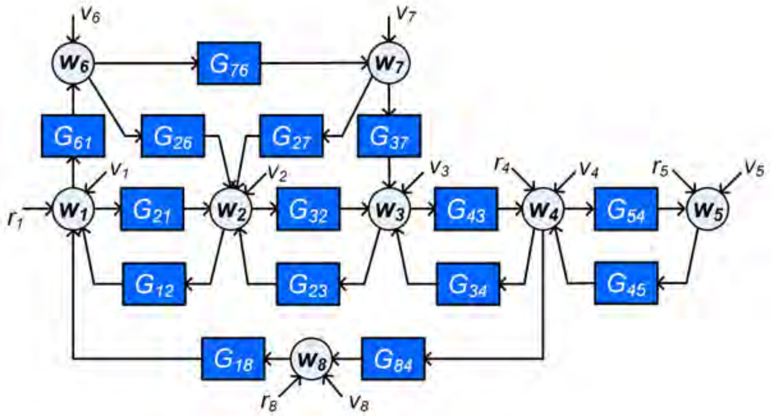


Figure 1.4: Module network framework for dynamic networks.

The nodes can be an input of a module and act as an output of another module. The network's topology is described by the existence of a module between two nodes, i.e., the interconnection structure between nodes through non-zero dynamics. A formal definition of the dynamic network setup will be given in Chapter 2. Nevertheless, for now, we will see some examples of dynamic networks using this modeling framework.

1.3.2 Vehicle platoons

Autonomous vehicles are a hot topic in the domain of automotive technology. Modern cars are equipped with sensors that sense the environment. This assists the driver and, if autonomous, helps in the self-driving operation of the vehicle. Therefore, each car will communicate with each other in cooperative driving and work together as a dynamic network. One classical problem in cooperative vehicles is the distance control in vehicle platooning. The concept of vehicle platooning is to form a convoy of vehicles driving close behind each other to increase the freeways traffic throughput and also to reduce fuel consumption for the follower vehicles, transportation costs, and greenhouse gas emissions [76].

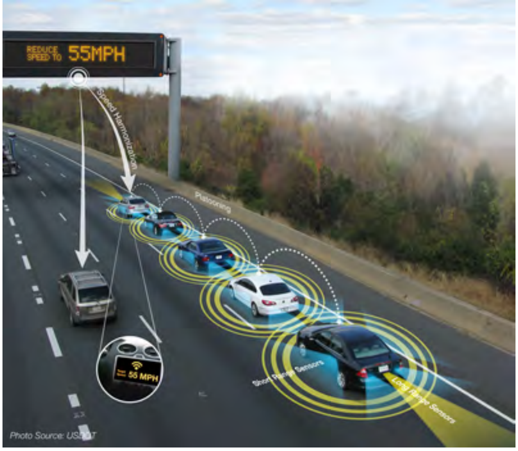


Photo Source: USDOT

Figure 1.5: Cooperative vehicles and vehicle platooning (Source : www.its.dot.gov).

In order to achieve fuel efficiency, it is essential to obtain small spacing between vehicles which increases the risk of an accident. The desired spacing policy can be achieved by forming autonomous vehicle platoons and using a distributed control scheme [76].

Interconnected vehicle platoons can be represented as a dynamic network. Here, one vehicle influences the dynamic behavior of the other vehicle, operating in a distributed control scheme. Therefore, the dynamic behavior of each vehicle constitutes the modules. The nodes represent the velocity of each vehicle and the relative distance between two vehicles. The setpoint or the velocity profile given to the leader (i.e., the first vehicle in the platoon) is the external excitation given to the dynamic network. A dynamic network model of vehicle platooning helps the engineers operate, maintain, design, and control the vehicles and obtain the desired fuel efficiency and performance. These dynamic network models are also used to infer parameters like air resistance coefficients of the vehicles using the observed data like velocity and relative distance between the vehicles.

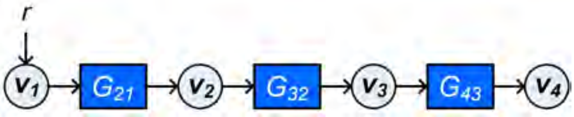


Figure 1.6: Vehicle-based dynamic network representation as described in [94]. G_{ij} represents the dynamics (transfer function) from the j^{th} vehicle velocity to the i^{th} vehicle velocity. The variables v_i represent the velocity of the i^{th} vehicle. The variable r refers to the excitation given to the leading vehicle, the velocity profile.

1.3.3 Industrial plants

Any industrial plant, including chemical or power plants, has many controlled subsystems with interacting dynamics. They operate in a distributed or decentralized process control scheme [58]. Different units (subsystems) in the plant have their local controllers designed and implemented for each subsystem by looking at their local cause and effect relationships and the effect of interactions from other subsystems (units). Therefore, the plants comprising of all the interacting units and controllers together constitute a dynamic network as seen in figure 1.7.

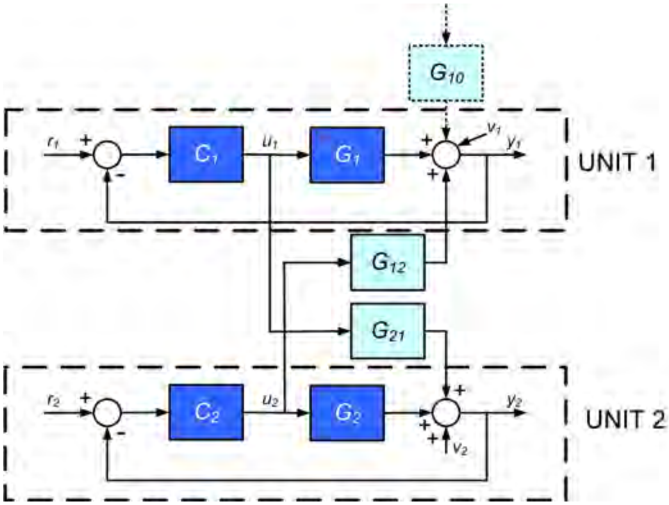


Figure 1.7: Two units of a plant being controlled with a local controller as a dynamic network. G_{ij} represent the interaction dynamics between the units [125].

Let us consider the example in [58] represented using Fig. 1.7, in which the problem of identifying interaction dynamics (i.e. G_{21} or G_{12}) that exist between units operating in a decentralized control scheme has been addressed. Normally these interactions between the units are not perceivable. Identification of such interaction relationships is crucial to the deployment of coordinated decentralized control and achieving the desired performance of the plant. The dynamic network identification facilitates the identification of the interacting dynamics.

1.3.4 Automotives

Electric and hybrid vehicles have entered the automotive market. They consist of various subsystems like the battery, internal combustion engine, electric motors, heat ventilation, air conditioning (HVAC) system, mechanical compressors. The energy in the vehicle is transformed between different physical domains. Therefore, they can be considered a dynamic network representing the power interaction in the vehicle with power as node signals connecting many subsystems, including energy storage devices and energy converters.

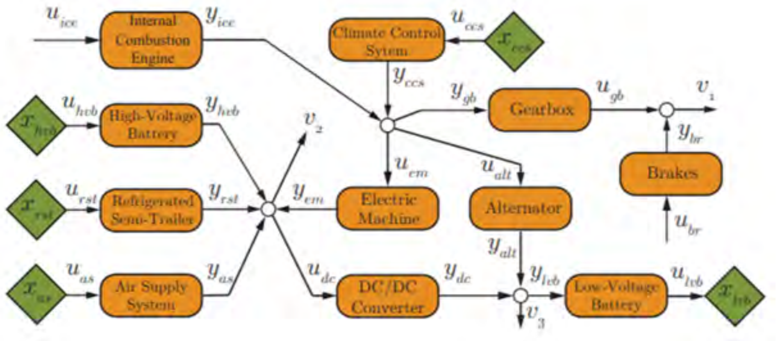


Figure 1.8: Power network of a hybrid truck. The topology and the arrows indicate the flow of power [111].

The dynamic network model can be used for Complete Vehicle Energy Management (CVEM) strategies [111] to reduce energy consumption. CVEM revolves around optimizing the energy consumption of all the interconnected subsystems in the vehicle by considering the interactions between them. This requires solving an optimal control problem for which the mathematical models of interconnected systems are required, which can be provided by the dynamic network models.

1.3.5 What model are we looking into?

In general, most of the systems in the real world are non-linear systems. Having said so, modeling non-linear systems using data and incorporating the time-varying behavior involves additional complexities. However, modeling linear time-invariant (LTI) systems is less complex than non-linear and time-varying systems. Moreover, there are well-advanced and extensive theories for the data-driven modeling of LTI systems in simple open-loop and closed-loop configurations. Therefore, it is a natural choice to extend these theories to a dynamic network setup since the theory of data-driven modeling of LTI dynamic networks is not advanced. This will serve as a basis to develop theories for non-linear dynamic networks. Also, many real-world systems can be approximated by a linear system when operating around a working point. Considering all these, in this thesis, we will consider linear dynamic networks where the modules have linear dynamics. Indeed, extending to non-linear or time-varying models will be the work of future scope.

Different LTI dynamic network models suitable for dynamic networks exist in the literature, as discussed in this section. For example, the state space models, probabilistic models, behavioral models. In this thesis, we choose a particular type of module network known as *transfer function network models* where the modules are described using LTI transfer functions. In essence, the modules $G_{21}, G_{32}, \ldots$ in figure 1.4 are LTI transfer functions. Transfer functions are extensively used in many engineering subfields like signal processing, time-series analysis, vibration analysis. They encode a causal relationship between measured signals. In addition, there are established links between transfer functions, state-space models, and models based on differential or difference equations.

The transfer function dynamic network has been presented as Dynamic Structure Functions (DSF) in [57], where there are no disturbances on the network. This representation has been extended in [124] with unknown disturbances. Similarly, dynamic networks with impulse response modules (also can be interpreted as a type of transfer function) and Wiener filters (can be related to transfer functions) have been presented in [23] and [85].

1.4 Why data-driven model learning?

We discussed how many large-scale, complex, and interconnected systems can be seen as dynamic networks and how a dynamic network model helps different applications. However, dynamic network models are generally not available. This raises the following question: *How to obtain the dynamic network models?*

There are two types of approaches to obtain mathematical models, namely *first principles modeling* and *data-driven modeling*. In the former, the equations using laws of physics or other sciences (like the law of conservation of mass, momentum) are used to find the relations between different variables. However, constructing models for dynamic networks only through first principles is sometimes costly or infeasible. In this case, using the measured data collected during the operation of the networked system can aid the modeling procedure. The data-driven modeling approach uses the measurement data from a system to infer a model of the system. This is based on the fact that the measurements contain information about the behavior of the system. Data-driven modeling approaches can be used to obtain the parameters of physical models (grey-box identification) and also to obtain models without any knowledge of physical interpretation of the system (black-box identification). With access to lots of data in this current world with advanced technologies, data-driven modeling approaches is an attractive choice to model dynamic networks.

The field of *system identification* deals with data-driven modeling problems of dynamical systems. According to [77, 120], in the classical approach of *system identification*, a user selects a *model set* containing the candidate models with the parameters to be determined. Then a model is selected from the candidate models through a rule called *criterion* that discriminates the candidate models from the best model using the *measured data set*. In practical situations, the data is corrupted by disturbances. Therefore, the criterion must take this into account and choose the best possible model. An accurate model that mimics the system's behavior is obtained when (1) the data is informative, (2) the set of candidate models contains an accurate system approximation, and (3) with a proper selection criterion. Data-driven modeling techniques are well known for simple open-loop, closed-loop, and MIMO unstructured systems. However, the data-driven modeling of a transfer function network that incorporates structure poses many interesting problems.

1.5 The research trends

Various problems in the data-driven modeling of dynamic networks can be formulated. In this section, we discuss four current research trends in the data-driven modeling of dynamic networks.

1.5.1 Topology identification

Topology identification or *topology detection* in a dynamic network deals with finding the interconnection structure of the dynamic network. The topology detection is a binary question, i.e., determining whether there exists a link between two nodes or not? Topology identification is a critical problem in many applications, e.g. neuroscience [119] and systems biology [64]. Many approaches for topology detection involve both topology detection and identifying the network dynamics simultaneously [23, 63, 152]. In [112], a compressive sensing problem is solved using linear regression models to find the interconnection between nodes of a dynamic network. In [85], topology identification based on Wiener filtering has been presented. In [72], efficient reconstruction of networks using Granger causality concept has been presented. In [23, 154], a Bayesian approach to find the interconnection structure of a dynamic network using kernel-based methods and sparsity inducing priors has been presented. In [135], a hyperparameter tuning-free sparse topology estimation has been introduced using covariance matching.

1.5.2 Full network identification

Full network identification deals with the identification of the full network dynamics from measured data. Normally, in this case, it is assumed that the interconnection structure between nodes is known. Identification of a full network by modeling it as a state-space model has been addressed in [60]. An identification method that can consistently identify all modules in a linear dynamic network with algebraic loops is presented in [142]. Identification of a full network in case of rank-reduced noise has been addressed in [144]. However, the method in [144] involves solving non-convex optimization problems. A sequential least-squares algorithm for full-network identification of ARMAX networks has been introduced in [140]. In [43], a method that involves only convex optimizations to solve the full network identification problem using Weighted Null-space fitting (WNSF) [49] has been presented for dynamic networks with white noise disturbances.

1.5.3 Single module identification

Single module identification or *local module identification* deals with the identification of a specific module (system) in the network. A local part of the dynamic network is of interest and needs to be identified in many situations. Consider the examples presented in Section 1.3. The example of identifying the air resistance coefficient of a vehicle among the vehicle platoons is a single module identification problem [94]. Similarly, identifying the interaction dynamics among different units in a large-scale industrial plant required for decentralized or distributed control [58] is a local module identification problem. In [79], an estimate of the reservoir thickness and permeability is obtained by identifying a single module in the dynamic network model (bilaterally coupled reservoir and wellbore model). Therefore, there is a strong need to develop a framework and theory for local module identification in large-scale interconnected dynamic networks.

For single module identification, the classical closed-loop prediction error identification techniques (direct and two-stage method) have been extended to a dynamic network setup with uncorrelated noise in [29, 124]. Considering the effect of sensor noise in the measurements, the setting mentioned above has been generalized in [27]. Identification of an individual transfer function in a dynamic network using wiener filters has been provided in [83]. In [54], the classical indirect identification method has been generalized to the dynamic network framework. A *simultaneous minimization of the prediction error* (SMPE) approach is introduced in [59] for identifying the target module (i.e. the module of interest that needs to be identified) in a dynamic network with only sensor noise. This method has been extended to a Bayesian setting in [38], where regularized kernel-based methods are used to decrease the variance of the estimated module. Also, an indirect identification method that involves only convex optimization has been presented in [48].

It is implicit that we can identify the full network by identifying each module in the network using single module identification. Similarly, by identifying the full network, we can identify each module. However, for identifying a single module, we do not want to identify the whole network. We require more measurements, excitation, and computational effort to identify the full network than those required to identify a single module. For instance, we do not need to measure all the nodes, have excitations be present on all nodes, and allow a correlation between noises at some nodes. The emphasis is to identify a module with local measurements and exploit the freedom in the signal selection. For example, in order to identify a module G_{21} in figure 1.4, we would want to use the local measurements w_1, w_2, w_3, w_6, w_7 and if possible have flexibility in selection among the signals.

1.5.4 Network identifiability

Identifiability of dynamic networks is a different problem from identification. It is a theoretical problem determining whether a unique network model exists that represents the stochastic properties of the measured signals. These properties are typically second-order properties like mean and power spectral density. The network identifiability aspects for an entire network have been addressed in many works like [7, 22, 57, 65, 133, 143]. Exploiting the concept of generic identfiability, path-based identifiability conditions based on pre-specified network topology are provided in [7, 65]. The problem to determine where to allocate excitation signal to guarantee network identifiability has been addressed in [6, 22]. Instead of the full network, identifiability works focussing on single module embedded in a dynamic network or sub-networks are addressed in [54, 117, 118, 145].

1.6 Open problems

There have been some works on data-driven modeling of a module in a dynamic network in the last ten years. However, the works involved restrictive assumptions and conditions to be satisfied that can be practically impossible in large-scale networks. This has opened many questions in the field of data-driven modeling in dynamic networks.

1.6.1 Restrictive assumption on disturbance correlation and confounding variables

The available single module identification methods can be broadly classified into two methods: the *direct methods* and the *indirect methods*. Indirect methods mostly rely on external excitations as inputs rather than the internal node signals. The required transfer function is not directly identified from the measurements. Closed-loop transfer functions are identified first and the required transfer function is estimated from it. We call this step as *post-processing*. The *two-stage method* in [29, 124], Errors-in-variables (EIV) method in [27, 120], SMPE method in [59] and the method in [54] represent particular forms of indirect methods. The direct method for single module identification like [29, 30, 124] uses the internal node signals as input and directly identifies the required transfer function.

The available direct identification methods for dynamic networks impose a restrictive assumption on the disturbances acting on the dynamic network. The common restriction in the literature is that there are no confounding variables for the estimation problem since its presence leads to biased estimates. A *confounding variable* is an unmeasured variable (like the disturbances) that affects both the input and output of the estimation problem. Take, for example, the dynamic network in figure 1.4. For a MISO estimation problem with inputs $\{w_1, w_3, w_7\}$ and output w_2, if v_1 and v_2 are correlated, then v_1 or v_2 directly affect both the input w_1 and output w_2. These correlated disturbances are *direct confounding variables*, and, realistically, there will be a correlation in the noise sources (disturbances) existing in a dynamic network. For example, when there exists spatial correlation of disturbances in a local area of the network, like a wind disturbance affecting multiple position measurements in a dynamic network. For the same example network and MISO estimation problem, the disturbance v_6 affects both the input w_1 and output w_2 through a node w_6 that is not included in the MISO estimation problem. Such unmeasured variables that affect through nodes in a network that are not included in the estimation problem are *indirect confounding variables*. Indirect confounding variables are common in the single module identification problem since we will only use local node measurements for our estimation.

The advantage of the indirect methods is that they can handle correlated process noise and confounding variables and achieve consistent estimates. However, they have restrictive requirements on excitations and also lead to estimates with high variance. On the contrary, the direct methods lead to consistent estimates with asymptotically minimum variance. However, the direct method requires a restrictive assumption on the absence of confounding variables. In [128], it is shown that correlated noise networks can be consistently identified moving from a MISO identification setup to a MIMO identification setup for a two-node network. However, there is no theory on handling correlated noise networks and confounding variables to identify a single module using direct methods.

1.6.2 Restrictive assumption on sensor noise and rank-reduced process noise

Many local module identification methods have been addressed, assuming that the data is measured with no noise (i.e., no sensor noise). However, in practice, this might not be the case. For example, in the network in figure 1.9, due to the presence of sensor noise, we do not measure the actual value of the node signal at time instant t (i.e $w_i(t), i = \{1, 2, 3\}$), but instead we measure $\tilde{w}_i(t) = w_i(t) + s_i(t)$ where $s_i(t)$ is the sensor noise disturbing the measurement of internal variable $w_i(t)$. These noise sources do not enter the dynamic network but disturb only the measurements. Therefore, we end up in an errors-in-variables problem. The *indirect methods* handles the presence of sensor noise and provides a consistent estimate. There are results considering only dynamic networks with sensor noise and no process noise [38, 48, 49]. Nevertheless, being indirect methods, they have restrictive requirements on excitations and lead to high variance estimates. Therefore the question here is: Are there alternative local module identification methods that can provide efficient estimates in the presence of sensor noise? This question remains open.

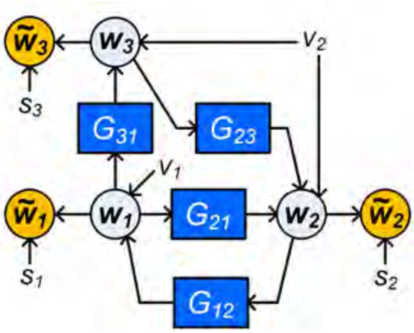


Figure 1.9: Example of a 3-node network with sensor noise and rank-reduced noise. Signals s_i are the sensor noise and v_i are the process noise signals or disturbances. Node signals are w_i and $\tilde{w}_i$ are the measurement of node signals with sensor noise.

Another restrictive assumption for direct identification methods that may not be applicable in practice is that a disturbance must enter at every node in the network or the situations where few sources dominate the disturbances in the network. This latter situation is depicted in figure 1.9 where the process noises on nodes 2 and 3 are the same (i.e. v_2). Handling such dynamic networks with rank-reduce noise has been addressed in [144]. In [144], the full network is identified using the *joint-direct method*, which is a prediction error approach with a Constrained Least Square (CLS) identification criterion. Having an established identification method for full network dynamics, identifying a specific module in the case of dynamic networks with rank-reduced noise is an unexplored area. The important challenge here is the signal selection and formulating a predictor model for the identification method that leads to unbiased estimates.

1.6.3 Limited focus towards quality measure for estimates other than consistency

The local module identification methods focus on consistency aspects of the estimation and the variance aspect has not been explored much. For consistent estimators, the variance of an estimator is bounded by the Cramér-Rao Lower Bound (CRLB). This means that no consistent estimators can achieve a consistent estimate with a variance smaller than the CRLB for the given model set and data set. This can be equated to Maximum Likelihood properties (both consistency and variance at the CRLB). However, developing single module identification methods that can achieve estimates with asymptotic efficiency is still an open question.

Some literature deals with reducing the variance aspect of indirect methods. In [59], the SMPE method is introduced to identify a specific module in a dynamic network with only sensor noise, reducing the estimates' variance. Variance analysis for networks with simple structures like cascaded modules are presented in [138], [41], [42] and Single-Input-Multiple-output (SIMO) structures in [39, 40].

Modern system identification methods focus on the aspect of achieving reduced Mean-squared Error (MSE) estimates. MSE includes the aspect of both bias and variance. So by allowing some bias in the estimates, it is possible to reduce the variance and the MSE using Bayesian methods and regularization. We can also deviate from consistency, an asymptotic property on data, to finite data identification approaches. The indirect SMPE method is extended to a Bayesian setting in [38], where regularized kernel-based methods are used to reduce the MSE of the identified closed-loop transfers. These are then used to identify the target module. The method considers only the presence of sensor noise and does not consider the presence of process noise. Therefore local module identification methods need to be developed incorporating different noise and identification frameworks focusing on reduced MSE of the estimates.

1.6.4 Informative data conditions

The identification methods that offer consistent estimates require a condition that the data should be informative. For example, consistency using the *direct method* in [124] requires sufficient conditions to be satisfied like the power spectrum of $\kappa(t)$ to be positive definite i.e. $\Phi_{\kappa(t)} > 0$ for sufficient number of frequencies ω, where $\kappa(t)$ is a vector of internal node signals. The challenging part is that the data mentioned here are the internal signals $w(t)$ in the dynamic network, which the user cannot directly manipulate. Therefore, it becomes hard to ensure data informativity and to design experiments that satisfy it. However, there is a possibility to translate the spectrum condition on internal signals to a condition on external signals that govern the dynamic network. This may require the use of the topology of the network as well. In this way, informative experiments can be designed for identification. Sufficient richness conditions on the external signals for the consistent identification of the desired module need to be developed. This is still an open problem in the field of system identification in dynamic networks.

1.6.5 Restrictive sensor location and actuation schemes

Both the direct methods like [29, 30] and the indirect methods like [54, 59] for single module identification requires a path-based condition called *parallel* *path/loop* condition to be satisfied in order to obtain consistent estimates. This condition warrants that every parallel path from the target module's input to the target module's output and every loop through the output passes through a measured node. For example, consider the network in figure 1.4 with G_{21} to be identified. There are three parallel paths from input of the target module w_1 to the output of the target module w_2 i.e. $(w_1 \rightarrow w_6 \rightarrow w_2)$, $(w_1 \rightarrow w_6 \rightarrow w_7 \rightarrow w_2)$, $(w_1 \rightarrow w_6 \rightarrow w_7 \rightarrow w_3 \rightarrow w_2)$ and two loops through output w_2 i.e. $(w_2 \rightarrow w_3 \rightarrow w_2)$, $(w_2 \rightarrow w_1 \rightarrow w_2)$. Therefore, the condition warrants w_3, w_6 to be measured along with input w_1 and ouput w_2. This imposes restrictions on the set of nodes to be measured for both methods. For indirect methods, it also imposes restrictions on the set of excitation that needs to be present. However, it might not be possible to measure certain nodes using sensors in practical situations, and it might also be impossible to actuate certain nodes. In these cases, the restriction inhibits our objective to identify the target module. Therefore, we need methods that can relax the restrictions and increase flexibility in sensing and actuation.

1.6.6 Assumptions on an apriori known topology

The single module identification or the full network identification methods in the available literature assume that the network topology is known apriori. Network topology estimation is seen as a separate problem in the literature of dynamic network identification. We might assume that we know the network's topology in some instances, and we would like to identify the modules. However, there can be cases where the network's topology is not known. Furthermore, for identifying local modules, we might need only the information on the local topology. Therefore, there are possibilities to develop methods that integrate the topology estimation and estimation of modules into a single identification task. Also, less attention has been paid to estimating the disturbance topology, i.e., the (spatial) noise correlation structure and the noise rank in the disturbance signal's filtered white noise representation. Therefore, identifying modules in the network with no prior information on noise topology is still a problem to be explored.

1.6.7 Algorithms for large-scale networks

Dynamic network identification methods are developed to be applied in practice. However, the focus is on identification methods to provide theoretical properties like consistency but not on how these methods can be practically applied to large-scale networks with scalable algorithms. Local module identification using prediction error methods (PEM) requires that we have to formulate a multi-input single-output (MISO) identification problem to identify a given module of interest. This implies that to avoid possible bias in the parameter estimates, one must identify all the modules constituting the MISO structure. It is bringing in the problem of a possibly very high number of parameters to be estimated that are of no interest to the user. In addition, it may be required to select the model order of each of these additional modules using model selection criteria such as

AIC, BIC, or cross-validation [77]. If the number of modules is high, one may have to test a huge combination of candidate model orders, making model order selection a computationally infeasible step (e.g., for five modules with FIR model structure and orders from 1 to 5, one has to test 5^5 possible combinations). Therefore it is necessary to avoid model order selection issues and reduce the number of nuisance parameters in single module identification. This is a non-trivial problem in a large-scale interconnected dynamic network since the MISO structure could have many modules.

Also, the available prediction error methods involve solving non-convex optimization problems that might run into local minima and achieve sub-optimal results. Also, for solving large-scale network identification problems, we would like to split the optimization problem and solve it in parallel to make it computationally effective. Therefore, developing effective algorithms plays a vital role and remains an open question.

1.7 Research Question

Many open problems related to identifying a module in a dynamic network have been discussed in the previous section. This thesis aims to guide users who wants to identify a module in a dynamic network and provide them the tools needed to achieve his objective of accurately identifying a module embedded in a dynamic network. In order to achieve this, the discussed open problems need to be addressed. This ultimately raises the following research question:

> How to effectively identify a module embedded in a dynamic network and obtain accurate estimates?

Let us now examine the important keywords in the research question.

- **Identify a module:** In this thesis, the main objective deals with identifying a single module embedded in a dynamic network. However, the full network identification or part of the network identification can be treated as an extension of the single module identification. We will touch upon that as well in this thesis. However, the primary focus will be on identifying a module in a dynamic network.

- **Effectively:** Effectiveness refers to flexibility in sensing and actuation, i.e., flexibility in choosing measured node signals and excitations. Also, it refers to developing algorithms that are scalable and less complex for large-scale dynamic networks.

- **Accurate:** Accuracy refers to the quality of the estimates. We focus on two quantities for accuracy of estimates (1) asymptotic efficiency, i.e., consistent estimates with variance at CRLB, and (2) reduced MSE.

1.8 Subquestions and the approach to solve them

We will now address the answer to the research question by formulating several sub-questions and answering them.

1.8.1 Handling correlated process noise and confounding variables

This sub-problem represents the problem discussed in Section 1.6.1. As already mentioned, correlated process noise is practically common in dynamic networks. It can be seen as a direct confounding variable. When we identify a module in a dynamic network with local measurements, we will also encounter indirect confounding variables. Confounding variables in an estimation problem lead to biased estimates and needs to be handled appropriately. Also, we would like to go beyond consistency and achieve minimum variance estimates and consistency, i.e., maximum likelihood properties. These lead to the following question:

How to handle confounding variables and identify a module embedded in a dynamic network and obtain maximum likelihood estimates?

The answer to this question is given in **Chapter 3**. Prediction error methods using a direct approach can be related to Maximum likelihood properties [77]. However, we need to handle confounding variables by suitably selecting the available node signals to incorporate in the identification experiment. For example, if not appropriately addressed, correlated noise in the dynamic network can lead to biased estimates. This can be avoided by adding nodes affected by the correlated noise as predicted variables in the identification setup. By doing so, we model the noise correlations and prevent bias in the estimates. Other situations, including the so-called indirect confounding variables [30], requires careful analysis, which is detailed in **Chapter 3**. We can end up in multiple node selection schemes that guarantee the objective. Therefore, a generally applicable theory is provided that is independent of the particular node selection scheme selected, and we call it as Local Direct Method (LDM).

- This chapter is based on the following publication:

  —— K.R. Ramaswamy, P.M.J. Van den Hof (2021). A local direct method for module identification in dynamic networks with correlated noise. *IEEE Transactions on Automatic Control, Vol. 66, no. 11, pp. 5237-5252,* DOI:10.1109/TAC.2020.3035634.

  whose preliminary work has been published in:

  —— P.M.J. Van den Hof, K.R. Ramaswamy, A.G. Dankers, G. Bottegal (2019). Local module identification in dynamic networks with correlated noise: the full input case. *Proc. 58th IEEE Conf. Decision and Control, Nice, France*, pp. 5494-5499, DOI: 10.1109/CDC40024.2019.9029448.

1.8.2 An alternative to direct and indirect methods

This section relates to the problem discussed in Section 1.6.5. The objective is to consistently identify a module in a dynamic network. There are already available direct and indirect methods that can solve this objective. However, both the available direct and indirect identification methods require restrictive conditions (i.e., parallel path/loop condition) on certain nodes to be measured and certain nodes to be excited. However, it might not be possible to measure certain nodes using sensors in practical situations, and it might also be impossible to actuate certain nodes. This may raise to situations when both direct and indirect methods cannot be used. Therefore we need a method that can provide flexibility in sensing and actuation, and solve the objective of identification when direct and indirect methods do not work. This raises the following question:

> How to increase the flexibility in selecting the measured node and excitation signals for consistent target module identification in a dynamic network?

The answer to this question is given in **Chapter 4**. Philosophically, the approach for the solution to the problem is simple. The indirect method uses only known external excitation signals as inputs which increase its requirement on actuations. However, it allows the flexibility to *post-process* the identified modules to extract the target module. On the other hand, the direct method uses both the known external signals and node signals as inputs which alleviates the requirement on actuation by using disturbances as excitation. However, it does not have a post-processing step. Therefore, we combine the elements of both direct and indirect methods. In this sense, we develop an identification method that can use both the external excitation signals and node signals as inputs and allow the post-processing of identified modules. We call this method the Generalized Method.

- This chapter is based on the following publication:

  —— K.R. Ramaswamy, P.M.J. Van den Hof, A.G. Dankers (2019). Generalized sensing and actuation schemes for local module identification in dynamic networks. *Proc. 58th IEEE Conf. Decision and Control, Nice, France*, pp. 5519-5524, DOI: 10.1109/CDC40024.2019.9029338.

1.8.3 Data-informativity conditions

This section relates to the problem discussed in Section 1.6.4. The experimenter would like to easily ensure data-informativity for any identification method and design experiments for identification. Therefore, the objective is to translate the data-informativity spectrum condition on internal signals that the user cannot manipulate to a condition on external signals. This raises the following question:

How to translate the data-informativity condition in local module identification methods to an experimenter-friendly condition?

The answer to this question is given in **Chapter 5**. This problem is approached by using the topological aspects of the dynamic network and coming up with generic (i.e., that do not depend on the numerical value of the module) path-based conditions that depend on paths from external signals to the internal node signals. In this way, if we know the topology of the dynamic network, we can verify and ensure data-informativity for a particular identification setup by checking the graph of the network. Also, this paves the way to a synthesis problem of allocating excitation to guarantee data-informativity using graphical conditions.

- This chapter is based on the following publication:

  —— P.M.J. Van den Hof and K.R. Ramaswamy (2020). Path-based data-informativity conditions for single module identification in dynamic networks. *Proc. 59th IEEE Conf. Decision and Control, Jeju Island, Republic of Korea*, pp. 4354-4359, DOI: 10.1109/CDC42340.2020.9304263.

1.8.4 Algorithm for consistent identification of modules without prior disturbance topology and rank information

This section relates to the problem discussed in Section 1.6.7 and 1.6.6 on the need for algorithms that can be practically applicable to large-scale networks. Identification methods for dynamic networks typically require prior knowledge of the network and disturbance topology. They often rely on solving poorly scalable non-convex optimization problems. While methods for estimating network topology are available in the literature, and sometimes the network topology is already known, less attention has been paid to estimating the disturbance topology and the noise rank. Therefore, we need to integrate the disturbance topology estimation in the network identification. This raises the following question:

How to effectively identify modules in a dynamic network with unknown disturbance topology and rank information, and obtain consistent estimates?

The answer to this question is given in **Chapter 6**. To answer this, we extend the multi-step Sequential Linear Regression [31] and Weighted Null Space Fitting methods [51] to deal with reduced rank noise, and use these methods to estimate the disturbance topology and the network dynamics. As a result, we provide a multi-step least squares algorithm with parallel computation capabilities that rely only on explicit analytical solutions. Thereby no non-convex optimizations are involved. Consequently, we consistently estimate dynamic networks of Box-Jenkins model structure while keeping the computational burden low.

- This chapter is based on the following publication:

  —— S.J.M. Fonken, K.R. Ramaswamy, P.M.J. Van den Hof (2022). A scalable multi-step least squares method for network identification with unknown disturbance topology. *To appear in Automatica, July 2022. ArXiv:2106.07548*.

1.8.5 Regularized kernel-based methods for reduced MSE

First, we look into the problem discussed in Section 1.6.7. In order to identify a module, we end up in a MISO/MIMO identification problem that is solved using PEM. This leads to a large number of parameters to be estimated, which increases variance and involves a model order selection step that is complex for large networks. This raises the following question:

> How to effectively identify a module in a large dynamic network and obtain estimates with reduced MSE?

This question is approached using regularized nonparametric kernel-based methods (see [93] for a survey on this subject). Keeping a parametric model for the module of interest, we can model the impulse response of the remaining modules in the MISO/MIMO structure as zero mean Gaussian processes with prior covariance matrix (kernel). By using this Bayesian framework, it is possible to incorporate prior knowledge. For example, by using the first-order stable spline kernel [21, 92] we can encode the stability and smoothness of an impulse response. The kernels are dependent on hyperparameters. By tuning the hyperparameters, it is possible to reduce the MSE of the estimates through *bias-variance trade off* [10]. Therefore, we need to tune the hyperparameters such that the MSE is reduced. We find the kernel hyperparameters and the target module parameters using the Empirical Bayes (EB) approach [80]. Therefore, we deviate from PEM that requires an enormous amount of data to achieve accurate results. We follow a Bayesian approach incorporating regularization to achieve reduced MSE that can perform well with limited data.

The answer to the question for uncorrelated noise networks is given in **Chapter 7**.

- This chapter is based on the following publication:

  —— K.R. Ramaswamy, G. Bottegal and P.M.J. Van den Hof (2021). Learning linear modules in a dynamic network using regularized kernel-based methods. *Automatica, Vol. 129, Article 109591, July 2021*, DOI: 10.1016/j.automatica.2021.109591.

  whose preliminary work has been published in:

  —— K.R. Ramaswamy, G. Bottegal and P.M.J. Van den Hof (2018). Local module identification in dynamic networks using regularized kernel-based methods. *Proc. 57th IEEE Conf. Decision and Control, Miami Beach, FL*, pp. 4713-4718, DOI: 10.1109/CDC.2018.8619436.

The answer to the question for correlated noise networks which requires handling a MIMO identification framework is given in **Chapter 8**.

- This chapter is based on the following publication:

—— V.C. Rajagopal, K.R. Ramaswamy and P.M.J. Van den Hof (2020). A regularized kernel-based method for learning a module in a dynamic network with correlated noise. *59th IEEE Conf. Decision and Control, Jeju Island, Republic of Korea*, pp. 4348-4353, DOI: 10.1109/CDC42340.2020.9303879.

1.8.6 Integration of local topology estimation

This sub-problem represents the problem discussed in Section 1.6.6. In order to select an appropriate predictor model for single module identification, one typically needs prior knowledge on the topology (interconnection structure) of the dynamic network and the correlation structure of the process disturbances. The objective is to integrate the estimation of this prior information into the identification algorithm and develop an entirely data-driven approach for learning the dynamics of a single module. Since the objective is to identify a single module, we are looking for local topology here. This raises the following question:

How to effectively identify a module in a large dynamic network with unknown prior topology?

The answer to the question is given in **Chapter 9**. We first find the nodes that do not affect the output of the target module and eliminate these nodes in the network to estimate the local topology. Next, the local topology is used to build the appropriate input/output setting for a predictor model in the local direct method under correlated process noise. Then the algorithm developed in Chapter 8 can be used to solve the identification problem. This leads to an identification algorithm with attractive statistical properties that is scalable to handle larger-scale networks too.

- This chapter is based on the following publication:

—— V.C. Rajagopal, K.R. Ramaswamy and P.M.J. Van den Hof (2021). Learning local modules in dynamic networks without prior topology information. *Proc. 60th IEEE Conf. Decision and Control, December 13-15, 2021, Austin, TX, USA*, pp. 840-845.

1.8.7 Handling missing node observations

In order to identify modules in a dynamic network using different identification methods, it is important to have measurements of certain nodes available. For example, we need node measurements to satisfy the parallel path/loop condition and handle confounding variables. We need these measurements to have reduced bias in the estimated target module. However, there may be situations where we cannot measure certain nodes. This raises the following question:

How to identify modules under missing node observations and obtain reduced MSE estimates?

The answer to the question is given in **Chapter 10**. Here, the objective is to identify a parametric model of the target module. We approach the problem by using a Bayesian approach and data augmentation strategy. We aim to reconstruct the missing node measurements and increase the accuracy of the estimated target module. To this end, we use regularized kernel-based methods coupled with approximate inference methods [10].

- This chapter is based on the following publication that is to be submitted:

—— K.R. Ramaswamy, G. Bottegal and P.M.J. Van den Hof (2022). Learning linear modules in a dynamic network with missing node observations. *To be Submitted to Automatica*.

1.9 Overview of contents

1.9.1 Chapter 2

A detailed definition of the type of dynamic network model class is provided. This thesis uses the classical PEM as well as Bayesian approaches. So, we provide details of the state-of-the-art PEM for dynamic network identification and the details about the Bayesian learning techniques.

1.9.2 Other chapters

The overview of other chapters has already been discussed in the previous section.

1.9.3 The Guide

This thesis aims to provide a guide to identify a module in a dynamic network. The contributions of each chapter in this thesis are the pieces of the puzzle in order to answer the research question of how to effectively estimate a module in a dynamic network and obtain accurate estimates. Assembling the puzzle pieces (i.e. the contributions of each chapter) leads to the decision flow-chart in figure 1.10, which guides the user to effectively learn a module in a dynamic network.

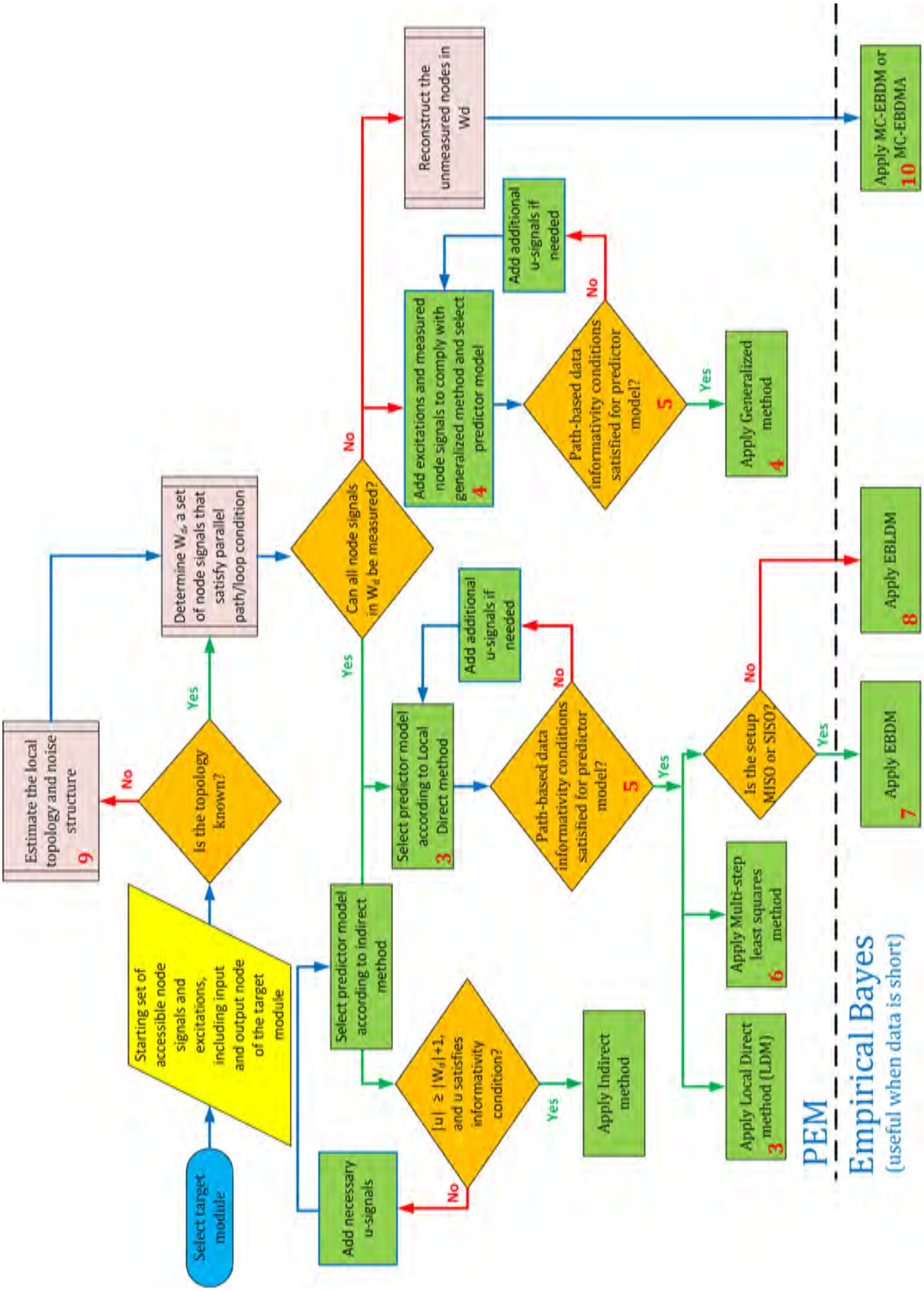

Select target module
Starting set of accessible node signals and excitations, including input and output node of the target module
Is the topology known?
No
Estimate the local topology and noise structure 9
Yes
Determine W_d, a set of node signals that satisfy parallel path/loop condition
Can all node signals in W_d be measured?
No
Reconstruct the unmeasured nodes in Wd
Add excitations and measured node signals to comply with generalized method and select predictor model 4
Path-based data informativity conditions satisfied for predictor model? 5
No
Add additional u-signals if needed
Yes
Apply Generalized method 4
Apply MC-EBDM or MC-EBDMA 10
Yes
Select predictor model according to indirect method
$|u| \geq |W_d|+1$, and u satisfies informativity condition?
No
Add necessary u-signals
Yes
Apply Indirect method
Select predictor model according to Local Direct method 3
Path-based data informativity conditions satisfied for predictor model? 5
No
Add additional u-signals if needed
Yes
Is the setup MISO or SISO?
No
Apply EBLDM 8
Yes
Apply EBDM 7
Apply Local Direct method (LDM) 3
Apply Multi-step least squares method 6
PEM
Empirical Bayes
(useful when data is short)

Figure 1.10: A guide for learning modules in a dynamic network. The number in red specifies the chapter of this thesis that contributes to the decision chart.

1.10 Video presentations with clickable link

ECC 2021 - Tutorial Session on Data-Driven Modeling in Dynamic Networks

IFAC 2020 [129, 130] - Single module identification - current status

CDC 2020 [127] - Path-based data informativity conditions

CDC 2020 [99] - Regularized kernel-based method for learning a module

CDC 2021 [100] - Learning local modules without prior topology information

1.11 Other publications

The other publications that are not included in this thesis:

—— K.R. Ramaswamy, P.Z. Csurcsia, J. Schoukens and P.M.J. Van den Hof (2021). A frequency domain approach for local module identification in dynamic networks. *To appear in Automatica, November 2022. ArXiv preprint ArXiv:2105.10901.*

—— P.M.J. Van den Hof and K.R. Ramaswamy (2021). Learning local modules in dynamic networks. *In Proceedings of the 3rd Conference on Learning for Dynamics and Control, volume 144 of Proceedings of Machine Learning Research*, pages 176–188. PMLR, ETH Zurich, Switzerland.

—— P.M.J. Van den Hof and K.R. Ramaswamy (2020). Single module identification in dynamic networks - the current status. *Extended abstract, Prepr. 21st IFAC World Congress, Berlin, Germany.*

—— K.R. Ramaswamy, O. Leeuwenburgh, R.M. Fonseca, M.M. Siraj and P.M.J. Van den Hof. Improved sampling strategies for ensemble-based optimization. *Computational Geosciences, Vol. 24, May 2020*, DOI: 10.1007/s10596-019-09914-8.

2 CHAPTER

Background

This chapter provides the necessary background for the research in this thesis. First, we define the setup of the linear dynamic network that we consider in this thesis in detail. This thesis deals with learning a module in a dynamic network from data. To achieve this, we follow two different approaches. One involves a classical system identification approach, namely the prediction error identification approach [77]. We provide state-of-the-art system identification approaches using the prediction error method for local module identification. The other direction of work of this thesis involves using the Bayesian kernel-based approach. To this end, we introduce the Bayesian estimation and tools from machine learning literature used in the second part of this thesis. At the end of this chapter, we lay the solid foundation for the remainder of this thesis.

2.1 Dynamic network setup

Dynamic networks are typically thought of as a set of signals (the node signals) interconnected through linear dynamic systems (the modules), possibly driven by known external excitations (the reference signals) and unknown external excitations (the disturbances). The dynamic network setup that we consider in this thesis has its origin from the *Dynamic Structure Functions* [57]. The basic setting of *Dynamic Structure Functions* that was introduced in [56], was generalized to a stochastic estimation and identification setting in [124], and has been adopted by several different authors as the setup of *dynamic networks*. We follow this setup in our thesis. In this setting, a dynamic network is built up out of L scalar *internal variables* or *nodes* w_j, $j \in \mathcal{L}$, and K *external variables* $r_k, k \in \mathcal{R}$. Here, $\mathcal{L}$ and $\mathcal{R}$ are sets with cardinality L and K respectively, where $\mathcal{L} = \{1, \ldots, L\}$ and $\mathcal{R} = \{1, \ldots K\}$. Each internal variable is described as:

$$w_j(t) = \sum_{\substack{l=1 \\ l \neq j}}^{L} G^0_{jl}(q) w_l(t) + u_j(t) + v_j(t) \tag{2.1}$$

where q^{-1} is the delay operator, i.e. $q^{-1} w_j(t) = w_j(t-1)$;

- G^0_{jl} are proper rational transfer functions, referred to as *modules*;
- There are no self-loops in the network, i.e. nodes are not directly connected to themselves $G^0_{jj} = 0$;
- u_j is an input signal, $u_j(t) = \sum_{k=1}^{K} R^0_{jk}(q) r_k(t)$ with r_k being the *known external variables* that can directly be manipulated by the user and $R^0_{jk}(q)$ is a known stable proper rational transfer function. It can be zero. We have the vector $u = [u_1 \cdots u_L]^T$.
- v_j is *unmeasured process noise*, where the vector process $v = [v_1 \cdots v_L]^T$ is modelled as a stationary stochastic process with rational spectral density $\Phi_v(\omega)$, such that there exists a white noise process $e := [e_1 \cdots e_p]^T$, $p \leqslant L$, with covariance matrix $\Lambda^0 > 0$ such that $v(t) = H^0(q)e(t)$, where $H^0(q)$ is a rational transfer function matrix. The noise model will be further specified in detail in Section 2.1.1.

Combining the L node signals we arrive at the full network expression

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} = \begin{bmatrix} 0 & G^0_{12} & \cdots & G^0_{1L} \\ G^0_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & G^0_{L-1\ L} \\ G^0_{L1} & \cdots & G^0_{L\ L-1} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} + R^0 \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_K \end{bmatrix} + H^0 \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_p \end{bmatrix}$$

which results in the matrix equation of the *dynamic network*:

$$w = G^0(q)w + R^0(q)r + H^0(q)e, \tag{2.2}$$

$$w = (I - G^0)^{-1}(R^0 r + H^0 e), \tag{2.3}$$

where by construction the matrix G is hollow, i.e. it has diagonal entries 0, while it encodes the topology of the network, i.e. $G^0_{j\ell}(q) \neq 0$ if and only if $\ell \in \mathcal{N}_j$ where $\mathcal{N}_j$ is the set of indices of node signals with direct causal connection to node w_j in the network. Also, $R^0_{j\ell}(q) \neq 0$ if and only if $\ell \in \mathcal{R}_j$, where $\mathcal{R}_j$ is the set of indices of external variables with direct causal connection to node w_j. A dynamic network can be graphically represented. An example of the above-defined dynamic network setup is provided in figure 2.1.

We make the following assumption for the dynamic network considered in this thesis.

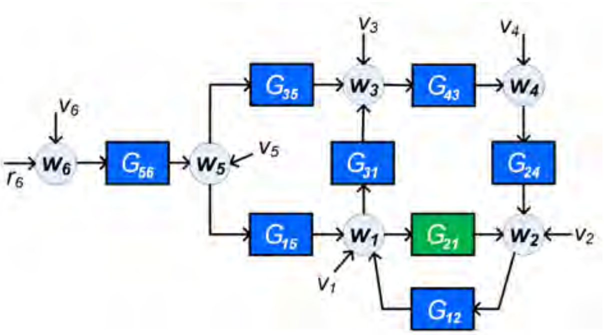


Figure 2.1: Example dynamic network with $u_6 = r_6$ and the green module G_{21} being the target module for identification.

Assumption 2.1 *The dynamic network is stable, i.e.* $(I - G^0)^{-1}$ *consists only of stable transfer functions (transfer functions with poles inside the unit circle). Also, the dynamic network is well-posed, i.e., all principal minors of* $\lim_{z\to\infty}(I - G^0(z))$ *are non-zero (see [26] for details).*

2.1.1 The noise model

In this section, we will further specify the noise model $v(t) = H^0(q)e(t)$ in detail. In order to evaluate the spectral contents of signals a definition of the power spectral density is required. The cross power spectral density of vector signals $a(t)$ and $b(t)$ is defined as[1]

$$\Phi_{ab}(\omega) := \mathcal{F}\{\bar{\mathbb{E}}[a(t)b^\top(t-\tau)]\},$$

where $\mathcal{F}$ is the discrete-time Fourier transform. The auto power spectral density of signal $a(t)$ is defined as

$$\Phi_{a}(\omega) := \mathcal{F}\{\bar{\mathbb{E}}[a(t)a^\top(t-\tau)]\}.$$

We denote the spectrum of the disturbance v as $\Phi_v(\omega)$. When the noise spectrum $\Phi_v(\omega)$ is full rank, it is called *full rank spectrum*. In this case, $\Phi_v(\omega)$ need not be necessarily diagonal, i.e., with uncorrelated noise components. However, in cases where the node signals can be noise-free or when the disturbances are related to each other through a linear filter, the noise spectrum can be singular. This is called *singular or rank-reduced spectrum*. For both these cases of the noise spectrum, we can represent the noise model using spectral factorization [150]. We will do that now using the result from [143] concerning unique representations of the reduced rank spectrum; full rank spectrum can be seen as a special case of the result.

[1] $\bar{\mathbb{E}}$ refers to $\lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N}\mathbb{E}$, and $\mathbb{E}$ the expected value operator.

Lemma 2.1 (from [144]) *Consider an L-dimensional disturbance process v with spectral rank p. Also, consider that the node signals and in turn the disturbance signals in v are reordered in such a way that $v = \begin{bmatrix} v_a^\top & v_b^\top \end{bmatrix}^\top$ where v_a is a full-rank process with rank p. Then the following unique representations result:*

$$\begin{aligned} &\begin{bmatrix} v_a \\ v_b \end{bmatrix} = H^0 e = \breve{H}^0 \breve{e} \quad \textit{with} \\ &H^0 = \begin{bmatrix} H_a^0 \\ H_b^0 \end{bmatrix}, \; \breve{H}^0 = \begin{bmatrix} H_a^0 & 0 \\ H_b^0 - \Gamma^0 & I \end{bmatrix}, \; \breve{e} = \begin{bmatrix} \breve{e}_a \\ \breve{e}_b \end{bmatrix} = \begin{bmatrix} e \\ \Gamma^0 e \end{bmatrix} \\ &\textit{and } \Gamma^0 = \lim_{z\to\infty} H_b^0(z) \end{aligned} \tag{2.4}$$

such that

- *$H_a^0 \in \mathbb{R}^{p\times p}(q)$ is a monic full rank rational transfer function matrix;*
- *$H_b^0 \in \mathbb{R}^{(L-p)\times p}(q)$ is a stable proper rational transfer function;*
- *$H^0 \in \mathbb{R}^{L\times p}(q)$ is stable and has a stable left inverse $H^\dagger$, that satisfies $H^\dagger H = I \in \mathbb{R}^{p\times p}$;*
- *e and $\breve{e}$ are white noise signals with dimensions p and L respectively;*
- *The covariance matrix of $\breve{e}$ is given by,*

$$\breve{\Lambda}^0 = \begin{bmatrix} I \\ \Gamma^0 \end{bmatrix} \Lambda^0 \begin{bmatrix} I \\ \Gamma^0 \end{bmatrix}^\top = \begin{bmatrix} \Lambda^0 & \Lambda^0 {\Gamma^0}^\top \\ \Gamma^0 \Lambda^0 & \Gamma^0 \Lambda^0 {\Gamma^0}^\top \end{bmatrix}, \tag{2.5}$$

  where $cov(e) = \Lambda^0 \in \mathbb{R}^{p\times p}$ has rank p,

- *If additionally H_a^0 is minimum phase then $\breve{H}^0 \in \mathbb{R}^{L\times L}(q)$ is a square monic rational transfer function that is stable and minimum-phase.*[a] □

[a] It has recently been pointed out in [18] that this excludes the situation where the (deterministic) mapping from v_a to v_b is unstable.

As a consequence of above lemma, for the *rank-reduced noise or singular noise*, $p < L$, we have two different representations for the noise model. That is, we can write

$$v = H^0 e = \breve{H}^0 \breve{e}.$$

In case of *full-rank noise*, $p = L$, and both the representations are the same (since v_b will be void).

2.2 Basic graph concepts

A dynamic network can be represented as a directed graph where the vertices correspond to the node signals and the links/edges constitute the modules. As already defined, $\mathcal{N}_j$ is the set of node indices k such that $G^0_{jk} \neq 0$, i.e., the node signals in $\mathcal{N}_j$ are called the *w-in-neighbors of the node signal w_j*. It is important to note that, in many works like [54, 84], these are mentioned with notation $\mathcal{N}_j^-$ to differentiate between the set of *w-out neighbors* which are represented with set $\mathcal{N}_j^+$. For example, w_j is a w-out neighbor of nodes in $w_k, k \in \mathcal{N}_j$. In the example network in Figure 2.1, w_4 is a w-in-neighbor of w_2 and w_3 is a w-out-neighbor of w_5.

Other graphical concepts that will be frequently used in this thesis are *paths and loops*. A *path* from a node (say w_{k1}) to a node (say w_{kn}) in the network is a sequence of directed edges that share at least a node, starting from node w_{k1} and ending at node w_{kn}, such that the edges are oriented in the same direction and no nodes are repeated. In graph theory, this is analogous to a *dipath* or *chain* [36]. A *loop* is a path that starts and ends at the same node i.e. a path with $w_{kn} = w_{k1}$. For example, in the example network in Figure 2.1, there is a path from w_6 to w_2 (i.e. $w_6 \rightarrow w_5 \rightarrow w_1 \rightarrow w_2$) and there is a loop through w_1 (i.e., $w_1 \rightarrow w_2 \rightarrow w_1$). Similarly, there is no path from w_1 to w_5 since the directed edges are not in the same orientation (i.e., $w_1 \rightarrow w_3 \leftarrow w_5$).

A *direct path* (not to be confused with directed path) from w_{k1} to w_{k2} is a path from w_{k1} to w_{k2} with no other nodes in the path; for example the path $w_6 \rightarrow w_5$ in Figure 2.1. A *simultaneous path* from w_{k1} to w_{k2} and w_{k3} indicates that there exists a path from w_{k1} to w_{k2} as well as from w_{k1} to w_{k3}. For example, the network in Figure 2.1 has a simultaneous path from w_1 to w_2 and w_3. Another important concept that we will be using is the *parallel path*. A *parallel path* from w_i to w_j is a path from w_i to w_j, excluding the direct path from w_i to w_j. The graph can include the signals in e and r as nodes as well. For example, in the example network in Figure 2.1, there exists a path from v_6 to w_5 (i.e. $v_6 \rightarrow w_6 \rightarrow w_5$) and a path from r_6 to w_3 (i.e. $r_6 \rightarrow w_6 \rightarrow w_5 \rightarrow w_3$).

There are a few more graph concepts used in this thesis. We will introduce it during this thesis.

2.3 Immersion

Immersion is the operation on a dynamic network that involves the removal of a set of nodes or internal variables in a dynamic network and representing the dynamic network with the remaining node signals [29]. In [29], this operation has been used to remove the unmeasured nodes in the network and form an *immersed network*. This immersed network has been used to find the dynamic relationships between the measured nodes. Immersion is a graph-based algorithm based on the lifting technique. The lifting technique works as follows. Consider a path $w_1 \rightarrow w_2 \rightarrow w_3$ and modules between the paths as in Figure 2.2 (left). If we are removing the node w_2, each path that passes through w_2 is lifted and the dynamics G_{32}^0 is removed and replaced by the dynamics $G_{32}^0 G_{21}^0$ between the path $w_1 \rightarrow w_2$ (see Figure 2.2 (right)). Therefore, there is no longer a path that passes through w_2 and it is important to note that the behavior of the remaining nodes (w_1 and w_2) remains unchanged. We will now consider an example.

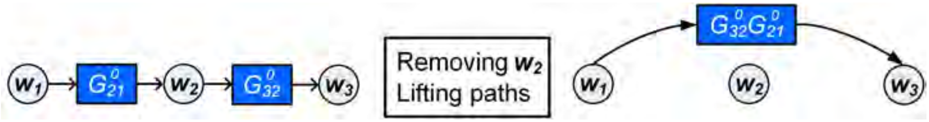


Figure 2.2: Illustration of lifting a path.

Example 2.1 *Consider a dynamic network as represented in Figure 2.3 with all noises in v uncorrelated with each other. Here w_4 and w_5 need to be immersed (or removed). First we will focus on node w_4. The path from w_3 to w_2 through w_4 and from external signal v_4 need to be lifted. When lifting technique is applied to the path $w_3 \rightarrow w_4 \rightarrow w_2$, we get a module $G_{23} + G_{24}G_{43}$ between w_3 and w_2, see Figure 2.4 (left). The term G_{23} is added since there was an already existing dynamics G_{23} in the path $w_3 \rightarrow w_2$. Similarly, we apply the lifting technique for the path from external signal v_4 through w_4. As a result we get $\tilde{v}_2 = v_2 + G_{24}v_4$. Now we focus on removing node w_5. Lifting the path $w_2 \rightarrow w_5 \rightarrow w_2$ will lead to a self-loop around node w_2, i.e., node w_2 is an input to itself. This is not allowed according to our network setup defined in Section 2.1. If we write the equation for node w_2 based on the network in Figure 2.4 (left), we have*

$$w_2 = (G_{23} + G_{24}G_{43})w_3 + G_{21}w_1 + G_{25}w_5 + r_2 + \tilde{v}_2, \tag{2.6}$$

where $w_5 = G_{52}w_2 + v_5$. The self-loop can be removed to obtain a network description that matches our definition, by moving the w_2 terms to the left-hand side, and normalizing by multiplication of the equation with $S = (1 - G_{25}G_{52})^{-1}$. This will lead to the immersed network in Figure 2.4 (right).

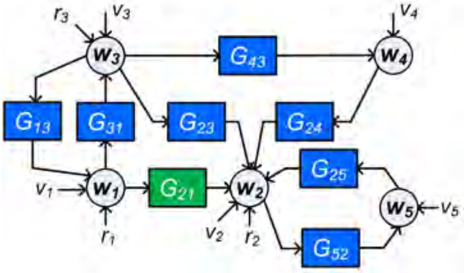


Figure 2.3: Example network with $u_1 = r_1, u_2 = r_2$ and $u_3 = r_3$.

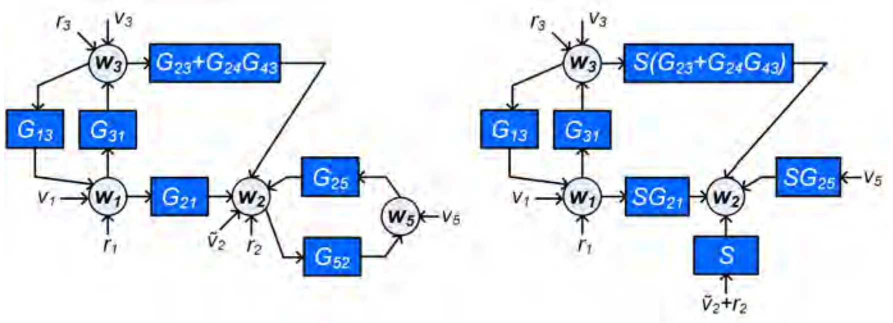


Figure 2.4: (Left) Immersed network of network in Figure 2.3 where the node w_4 has been immersed, where $\tilde{v}_2 = (v_2 + G_{24}v_4)$. (Right) Immersed network of network in Figure 2.3 where the nodes w_4 and w_5 have been immersed, where $S = (1 - G_{25}G_{52})^{-1}$.

2.4 The single module identification problem

The single module identification problem is the problem of identifying one particular module $G^0_{ji}(q)$ on the basis of measured time-series of a subset of variables in w, and possibly r. This is illustrated in the network depicted in Figure 2.1.

It may be clear that simply measuring the input and output of the target module and estimating a model based on these signals will generally not lead to accurate results because of the signal correlations that are induced by the remaining part of the network. For example, in the situation of Figure 2.1, with G_{21} being the target module for estimation, estimating the dynamics on the basis of input w_1 and output w_2 only will provide an estimated model that includes the dynamics of the "parallel path" $G_{24}G_{43}G_{31}$. On the other hand, performing a full network identification to identify a single module requires expensive experiments and measurements of many nodes to identify all the modules in the network. Therefore, we need to explore single module identification methods that can learn the dynamics of a module using local measurements and excitations.

In the next section we will make a general classification of the available state-of-the-art single module identification approaches. The classical closed-loop system identification prediction error methods [77] are extended to a dynamic network setting in these approaches. For more details on identification with prediction error methods, we refer to [77].

2.5 Main approaches

We can distinguish the PEM approaches for addressing the single module identification problem in dynamic networks broadly into two categories, namely the *direct* and the *indirect* method. First, we will provide the background for the two categories and then paint an overall picture of these two main approaches for single module identification. The target module is indicated by G_{ji}^0 i.e. the module between w_i and w_j.

2.5.1 The MISO direct method

The MISO direct method is an extension of the direct method for closed-loop identification [77]. The objective is to estimate a particular transfer function embedded in the network. This transfer function is denoted by G_{ji}. This method is a prediction error method based on the concept of prediction, i.e., a good model of the node $w_j(t)$ should predict $w_j(t)$ based on the available present or past signals. To this end, the method uses the *one-step-ahead predictor*[2] $\hat{w}_j(t|t-1) := \bar{\mathbb{E}}\{w_j(t) \mid w_j^{t-1}, w_{\mathcal{D}_j}^t, u_j(t)\}$ ([77]) and build a predictor model

$$\hat{w}_j(t|t-1) = (1 - H_j(q)^{-1})w_j(t) + H_j(q)^{-1}[(\sum_{k\in\mathcal{D}_j} G_{jk}(q)w_k(t) + u_j(t)]. \tag{2.7}$$

with predicted output w_j, and predictor inputs u_j, w_k where $k \in \mathcal{D}_j$. The set $\mathcal{D}_j$ is left unspecified at the moment and will be specified later. The signals in the network that are not in the predictor model are discarded from the estimation. Here, $\hat{w}_j(t|t-1)$ is a part of $w_j(t)$ that can be predicted using the present and past values of signals, and the part that cannot be predicted is the innovation $e_j(t)$, i.e., $w_j(t) = \hat{w}_j(t|t-1) + e_j(t)$ with the innovation being uncorrelated to the predictor $\hat{w}_j(t|t-1)$.

As already defined $\mathcal{N}_j$ is the set of node indices k such that $G_{jk}^0 \neq 0$, i.e. the node signals in $\mathcal{N}_j$ are the *w-in-neighbors of the node signal w_j*. Let $\mathcal{D}_j$ denote the set of indices of the internal variables that are chosen as predictor inputs. It seems most obvious to have $\mathcal{D}_j \subseteq \mathcal{N}_j$, but this is not necessary, as will be shown later in the next section. Let $\mathcal{V}_j$ denote the set of node indices k such that v_k has a path to w_j. Let $\mathcal{Z}_j$ denote the set of indices not in $\{j\} \cup \mathcal{D}_j$, i.e. $\mathcal{Z}_j = \{1, \ldots, L\} \backslash \{\{j\} \cup \mathcal{D}_j\}$, reflecting the node signals that are discarded in the prediction/identification. Let $w_{\mathcal{D}}$ denote the vector $[w_{k_1} \ \cdots \ w_{k_n}]^T$, where $\{k_1, \ldots, k_n\} = \mathcal{D}_j$. Let $u_{\mathcal{D}}$ denote the vector $[u_{k_1} \ \cdots \ u_{k_n}]^T$, where $\{k_1, \ldots, k_n\} = \mathcal{D}_j$. The vectors $w_{\mathcal{Z}}$, $v_{\mathcal{D}}$, $v_{\mathcal{Z}}$ and $u_{\mathcal{Z}}$ are defined analogously. The ordering of the elements of $w_{\mathcal{D}}$, $v_{\mathcal{D}}$, and $u_{\mathcal{D}}$ is not important, as long as it is the same for all vectors. The transfer function matrix between $w_{\mathcal{D}}$ and w_j is denoted $G_{j\mathcal{D}}$. The other transfer function matrices are defined analogously.

To illustrate the notation, consider the network sketched in Figure 2.5, and let module G_{21}^0 be the target module for identification. Then $j = 2, i = 1; \mathcal{N}_j = \{1, 4\}$ i.e. indicates the indices of w-in neighbors of $w_j = w_2$. If we choose the set of predictor inputs as $\mathcal{D}_j = \mathcal{N}_j$, then the set of remaining (nonmeasured) signals, becomes $\mathcal{Z}_j = \{3, 5, 6\}$.

[2] $\bar{\mathbb{E}}$ refers to $\lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} \mathbb{E}$, and w_j^ℓ and $w_{\mathcal{D}_j}^\ell$ refer to signal samples $w_j(\tau)$ and $w_k(\tau)$, $k \in \mathcal{D}_j$, respectively, for all $\tau \leqslant \ell$.

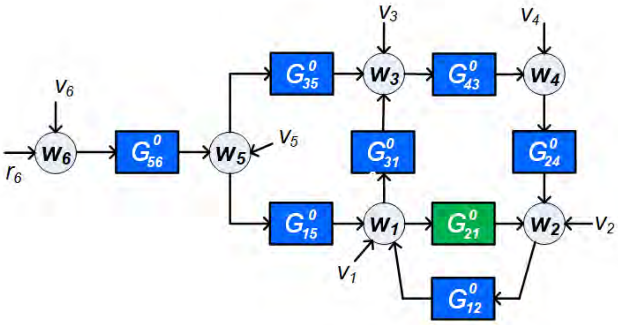


Figure 2.5: Example network with target module G^0_{21} (in green).

By this notation, the network equation (2.2) is rewritten as:

$$\begin{bmatrix} w_j \\ w_{\mathcal{D}} \\ w_{\mathcal{Z}} \end{bmatrix} = \begin{bmatrix} 0 & G_{j\mathcal{D}} & G_{j\mathcal{Z}} \\ G_{\mathcal{D}j} & G_{\mathcal{D}\mathcal{D}} & G_{\mathcal{D}\mathcal{Z}} \\ G_{\mathcal{Z}j} & G_{\mathcal{Z}\mathcal{D}} & G_{\mathcal{Z}\mathcal{Z}} \end{bmatrix} \begin{bmatrix} w_j \\ w_{\mathcal{D}} \\ w_{\mathcal{Z}} \end{bmatrix} + \begin{bmatrix} v_j \\ v_{\mathcal{D}} \\ v_{\mathcal{Z}} \end{bmatrix} + \begin{bmatrix} u_j \\ u_{\mathcal{D}} \\ u_{\mathcal{Z}} \end{bmatrix}, \tag{2.8}$$

where $G_{\mathcal{D}\mathcal{D}}$ and $G_{\mathcal{Z}\mathcal{Z}}$ have zeros on the diagonal.

For identification of module G_{ji} we select $\mathcal{D}_j$ such that $i \in \mathcal{D}_j$, and subsequently estimate a multiple-input single-output (MISO) model for the transfer functions in $G_{j\mathcal{D}}$, by considering the *one-step-ahead predictor* $\hat{w}_j(t|t-1;\theta) := \bar{\mathbb{E}}\{w_j(t) \mid w_j^{t-1}, w_{\mathcal{D}_j}^{t};\theta\}$ ([77]) and the resulting *prediction error* $\varepsilon_j(t,\theta) = w_j(t) - \hat{w}_j(t|t-1;\theta)$, leading to:

$$\varepsilon_j(t,\theta) = H_j(q,\theta)^{-1}[(w_j(t) - \sum_{k\in\mathcal{D}_j} G_{jk}(q,\theta)w_k(t) - u_j(t)]. \tag{2.9}$$

The signal $u_j(t)$ in (2.9) is known and are not involved in the predictor model through parameterized components. The parameterized transfer functions $G_{jk}(q,\theta)$, $k \in \mathcal{D}_j$ and $H_j(q,\theta)$ are estimated by minimizing the sum of squared (prediction) errors (i.e. the *identification criterion*):

$$\hat{\theta}_N = \arg\min \frac{1}{N}\sum_{t=0}^{N-1} \varepsilon_j^2(t,\theta), \tag{2.10}$$

where N is the length of the *data set*. Under mild assumptions[3], it leads to consistent estimates, i.e.

$$\hat{\theta}_N \to \theta_0 \text{ with probability 1 as } N \to \infty,$$

provided additional conditions like data informativity are satisfied. Here, θ^0

[3] We will assume that the standard regularity conditions on the data are satisfied that are required for convergence results of the prediction error identification method. See [77] page 249; this includes the property that $e(t)$ has bounded moments of order higher than 4.

represents the true parameters that generated the data. We refer to this identification method as the *MISO direct method* that was introduced in [124]. In the MISO direct method considered in [124], the predictor inputs $\mathcal{D}_j$ is chosen to be $\mathcal{N}_j$. It is shown that the method leads to consistent estimates under the following conditions:

Proposition 2.1 (from [124]) *The estimates* (2.10) *are consistent under the following conditions:*

1. *The noise v_j is uncorrelated to all other noise signals that have a path to w_j i.e. all noise signals $v_k, k \in \mathcal{V}_j$;*
2. *The noise v_j is uncorrelated to all excitation signals in u;*
3. *Every loop through w_j in the network and in its parameterized model has a delay (*delay in loop condition*);*
4. *The system is in the model set ($\mathcal{S} \in \mathcal{M}$) i.e. there exists a θ_0 such that $G_{jk}(q, \theta_0) = G^0_{jk}(q)$ and $H_j(q, \theta_0) = H^0_j(q)$;*
5. *The spectral density of $\kappa(t) = \begin{bmatrix} w_{\mathcal{D}}(t)^\top & e_j(t) \end{bmatrix}^\top$, denoted as $\Phi_{\kappa(t)}(\omega)$ is positive definite for a sufficiently high number of frequencies ω (*data-informativity condition*).*

In [29], it is shown that the number of node signals that needs to be included in $w_{\mathcal{D}}$ can be further reduced by selecting $\mathcal{D}_j$ such that upon removal (*immersion* [29]) of the remaining *unmeasured nodes* (i.e. the nodes that are discarded in the identification/predictor model) from the network, while keeping the remaining node signals invariant, the target module remains the same. This is achieved if the *parallel path and loop condition* is satisfied ([29]):

> **Condition 2.1 (Parallel path/loop condition)** *In the graph of the dynamic network, every parallel path from the input of the target module w_i to the output of the target module w_j and every loop through w_j passes through a node that is included in $w_{\mathcal{D}_j}$, along with $i \in \mathcal{D}_j$.*

The above condition is a *target module invarinace* condition. Therefore, it is sufficient to select $\mathcal{D}_j \subset \mathcal{N}_j$ and (possibly) few other node signals that satisfies the condition as predictor inputs for the MISO direct method. For example, consider the network in figure 2.6 with the target module G_{21}. There are three parallel paths from input of the target module to the output of the target module i.e. ($w_1 \rightarrow w_6 \rightarrow w_2$), ($w_1 \rightarrow w_6 \rightarrow w_7 \rightarrow w_2$), ($w_1 \rightarrow w_6 \rightarrow w_7 \rightarrow w_3 \rightarrow w_2$) and two loops through output w_2 i.e. ($w_2 \rightarrow w_3 \rightarrow w_2$), ($w_2 \rightarrow w_1 \rightarrow w_2$). Therefore, selecting $w_{\mathcal{D}} = \{w_1, w_3, w_6\}$ is sufficient for the MISO direct method instead of selecting all the w-in neighbors of the output of the target module (i.e. w_7 can be excluded).

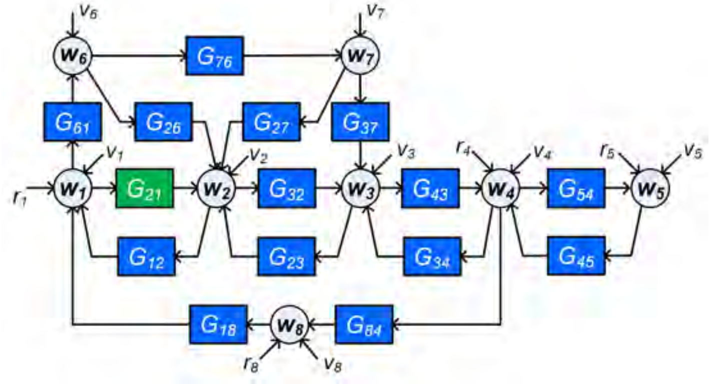


Figure 2.6: Example network in [125] with target module G_{21} (in green). There are three parallel paths i.e. $(w_1 \rightarrow w_6 \rightarrow w_2)$, $(w_1 \rightarrow w_6 \rightarrow w_7 \rightarrow w_2)$, $(w_1 \rightarrow w_6 \rightarrow w_7 \rightarrow w_3 \rightarrow w_2)$ and two loops through $w_j = w_2$ i.e. $(w_2 \rightarrow w_3 \rightarrow w_2)$, $(w_2 \rightarrow w_1 \rightarrow w_2)$.

However one of the possible consequences of removing an w-in-neighbour of w_j from $w_{\mathcal{D}}$ is that the disturbance signals that are acting on w_j and those acting on $w_{\mathcal{D}}$ can get correlated. The same situation occurs if the disturbance signals in v are correlated, i.e., the spectral density $\Phi_v(\omega)$ is non-diagonal. In those situations, we have to deal with the presence of confounding variables[4]. When not properly accounted for, confounding variables typically destroy the consistency properties of the direct method for estimating G^0_{ji}, as they introduce correlation between the measured node signals w_i and w_j that is not induced by the module dynamics G^0_{ji}. Phrased in identification-terms, confounding variables are correlated disturbances. In practical situations, confounding variables are highly unavoidable since not all nodes can be measured and noises can be correlated as well.

> **Remark 2.1** *In the MISO direct method, both [124] and [29], consistency results are shown by making a strong assumption that there are no confounding variables.*

2.5.2 The indirect method

The network model (2.3) can be rewritten as $w = T^0 r + \bar{v}$ where $T^0 = (I - G^0)^{-1} R^0$ and $\bar{v} = (I - G^0)^{-1} H^0 e$. This description of the dynamic network is called the *input-output model* of the dynamic network. Let us now explain the general approach of the indirect method on a high level. A consistent estimate $\hat{T}(q)$ of $T^0(q)$ can be obtained using an open loop MIMO identification method as in (2.11). In order to obtain a consistent estimate $\hat{G}$ of G^0 from the estimated $\hat{T}$, a

[4] A confounding variable is an unmeasured variable that affects both the input and output of an estimation problem. We will establish a more formal definition in the upcoming chapter.

post-processing step is necessary. On the basis of $\hat{T}(q)$, a consistent estimate $\hat{G}$ of G^0 can be obtained by solving $(I - \hat{G})\hat{T}(q) = R^0$. By identifiying only a submatrix of T and solving only a subset of the above equations, a target module embedded in the dynamic network can be identified, see [54].

The *indirect method* is based on selecting a particular set of predictor input signals r_k, $k \in \mathcal{D}$, and a set of predicted outputs w_ℓ, $\ell \in \mathcal{Y}$, that are used in a predictor model, leading to the following prediction error:

$$\varepsilon(t,\theta) = \bar{H}(q,\theta)^{-1}\left[w_{\mathcal{Y}}(t) - \bar{T}(q,\theta)r_{\mathcal{D}}\right] \tag{2.11}$$

The matrix $\bar{T}(q,\theta)$ refers to the parameterized transfer function matrix model of a submatrix of the network transfer matrix T^0, which maps external signals r into internal node signals w, and $\bar{H}(q,\theta)$ is the noise model. $\bar{T}(q,\theta)$ is estimated using an open loop MIMO identification method by minimizing a scalar cost function over θ, as e.g., the quadratic cost function

$$\hat{\theta}_N := \arg\min_{\theta} \frac{1}{N}\sum_{t=1}^{N} \varepsilon^T(t,\theta)P\varepsilon(t,\theta) \tag{2.12}$$

with P a positive definite weighting matrix. This will lead to consistent estimates in (2.12) under informative data conditions. However, in order to extract the dynamics of the target module G^0_{ji} from the estimates of $\bar{T}(q,\theta)$, we need to appropriately choose the predictor model. This requirement will lead to a predictor model setup as depicted in Figure 2.7. The output $w_{\mathcal{Y}}$ of the predictor model is selected to be composed of w_j and all node signals $w_{\mathcal{N}}$ that are in-neighbours of w_j, i.e. $\mathcal{N} = \mathcal{N}_j$. Therefore $\bar{T}$ in (2.11) is decomposed as

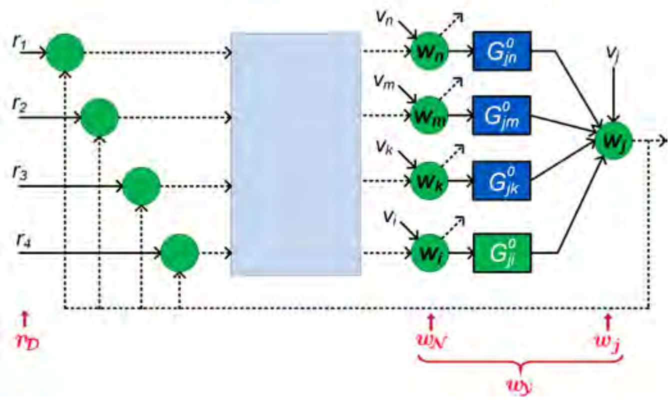


Figure 2.7: Predictor model for the indirect method.

$$\bar{T} = \begin{bmatrix} \bar{T}_{jr_{\mathcal{D}}} \\ \bar{T}_{\mathcal{N}r_{\mathcal{D}}} \end{bmatrix}. \tag{2.13}$$

If a consistent estimate $\hat{T}$ of $\bar{T}$ is made, then a consistent estimate of $G^0_{j\mathcal{N}}$ is obtained according to

$$\hat{G}_{j\mathcal{N}} = \hat{T}_{jr_{\mathcal{D}}}\hat{T}^{\dagger}_{\mathcal{N}r_{\mathcal{D}}} \tag{2.14}$$

where $\hat{T}^{\dagger}_{\mathcal{N}r_{\mathcal{D}}}$ is the right pseudo inverse of $\hat{T}_{\mathcal{N}r_{\mathcal{D}}}$.

The fact that the right inverse of $\hat{T}_{\mathcal{N}r_{\mathcal{D}}}$ needs to exist, requires the presence of a sufficient number of external excitation signals r in the network, i.e. $dim(r_{\mathcal{D}}) \geqslant dim(w_{\mathcal{N}})$. So we need at least as many external excitation signals r to be present as there are in-neighbours of w_j. The r-signals can be added to the nodes in $w_{\mathcal{N}}$ directly (as suggested in [54]), or can also be added elsewhere in the network, as in Figure 2.7. For the indirect method the requirements for *data-informativity* are rather straighforward: a sufficient condition for data-informativity is that $\Phi_{r_{\mathcal{D}}}(\omega) > 0$ for almost all ω, with $\Phi_{r_{\mathcal{D}}}(\omega)$ the spectral density of $r_{\mathcal{D}}$.

The indirect method has been studied in different settings. In [54], a setting with a predictor model having w_j and w-out neighbors of w_j in $w_{\mathcal{Y}}$ is also provided. Related indirect methods, such as the *two-stage method* and the *Instrumental Variable* (IV) method have been presented in [124] and [27] respectively. All these methods depend on the external r signals and hence might prove expensive due to the requirement of more actuators and costly experiments. Nevertheless, all the indirect methods can handle an Errors-in-variables (EIV) setting as well as networks with correlated process noise.

2.5.3 Reflections on the MISO direct and indirect methods

In order to arrive at accurate, consistent estimates of our target module, there are two prime requirements for the estimation setup:

1. An appropriate *predictor model* needs to be chosen. This choice of predictor model includes a selection of node signals to be included as measured signals, and to select inputs and outputs in the predictor model. The predictor model determines where sensors should be available in the network. The predictor model needs to satisfy particular properties in order to guarantee that the target module indeed can be estimated and no uncontrolled bias occurs in the estimated model.

2. The measured data that is taken from the network needs to satisfy condition of *data-informativity*, in other words it needs to be sufficiently rich in order to provide accurate estimates.

The two identification approaches (direct and indirect) are distinguished by the choice of predictor model. Whereas indirect methods use predictor models having only external excitation signals r as predictor input, the direct method also uses node signals w as predictor input. Different choices of predictor models will lead to different conditions for data-informativity. The direct method (2.9) has node signals $w_{\mathcal{D}}$ as predictor inputs, and therefore utilizes both external signals r and e for creating data-informativity. On the other hand, indirect methods rely on external excitation signals r for data informativity, and therefore will typically require more expensive experiments. The direct method provides asymptotically efficient estimates (i.e. consistency and minimum variance for the identification setup) at the cost of the need to include noise models $\bar{H}(q,\theta)$. The indirect method and its variations provides consistent estimates but not with minimum variance, however they do not necessarily require noise models $\bar{H}(q,\theta)$ for consistent estimates. When the node signals are measured with sensor noise (errors-in-variables (EIV) situation) or when the noises are correlated, the direct method becomes biased but the indirect method provides consistent estimates of the target module. Another important difference is that, in the direct method the predictor model is chosen such that the target module is directly parameterized an estimated as a part of the MISO model. This reflects the need for *parallel path/loop condition*, which falls under the condition for *target module invariance*. However, in indirect method, the predictor model is chosen such that the target module can be obtained from the estimated models using *post-processing*.

2.6 Bayesian learning

In this thesis, apart from PEM, we also take a Bayesian approach to simplify many complexities in identification of modules in a dynamic network.

2.6.1 PEM and Maximum Likelihood estimators

To provide the necessary background for the Bayesian approach used in this thesis, let us consider a data generating system of single-input-single-output (SISO) Finite Impulse Response (FIR) type given by the difference equation,

$$y(t) = \sum_{k=1}^{n} \theta_k u(t-k) + e(t), \tag{2.15}$$

with $y(t)$ and $u(t)$ are the output and input of the system; let us assume that $e(t)$ is a white Gaussian noise with variance σ^2. The above model can be given by,

$$y(t) = \phi^\top(t)\theta + e(t) \tag{2.16}$$

where $\theta = \begin{bmatrix}\theta_1 & \dots & \theta_n\end{bmatrix}^\top$ and $\phi(t) = \begin{bmatrix}u(t-1) & \dots & u(t-n)\end{bmatrix}^\top$. In the *frequentist approach*, the main tool to learn a model is using Maximum Likelihood (ML) estimation (see [10], Section 1.2.3). We find the ML estimate, i.e., the θ that best explains the data by maximizing the likelihood function:

$$\hat{\theta}_{ML} = \arg\max_{\theta} p(\mathbf{y};\theta) \tag{2.17}$$

where $\mathbf{y} = \begin{bmatrix}y(1) & \dots & y(N)\end{bmatrix}^\top$ and N is the length of the dataset. The above estimator is called *Maximum Likelihood estimator*. Since the logarithm is a monotonic function, we can maximize the log likelihood:

$$\hat{\theta}_{ML} = \arg\max_{\theta} \log p(\mathbf{y};\theta) = \arg\min_{\theta} \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{t=1}^{N} \left(y(t) - \phi^\top(t)\theta\right)^2 + \frac{N}{2}\log\sigma^2. \tag{2.18}$$

Neglecting the terms that are not dependent on θ, we can observe that, the above cost function is the same as the cost function for the prediction error methods (i.e. the sum of the square of the prediction errors). This implies that, assuming the noise is Gaussian, the estimate using PEM is a Maximum Likelihood estimate. The ML estimates have very attractive properties. One property is consistency, which we have already discussed before. The other attractive property is that the estimates are asymptotically normal. It means that as we increase the number of data N, the estimates become normally distributed around the true value of the parameters θ_0 with a covariance matrix that is the inverse of the *Fisher Information matrix*. The inverse of the Fisher information matrix determine the Cramér-Rao Lower Bound (CRLB). This means that, as the number of data grows, the Maximum Likelihood estimator has the smallest variance and also the minimum mean-squared error among all unbiased estimators. Therefore, this estimator is *asymptotically efficicient* since there are no unbiased estimators that can provide a smaller variance than the ML estimator.

Since the problem in (2.18) is the linear in parameters, the ML estimate $\hat{\theta}_{ML}$ (also the PEM estimate $\hat{\theta}_{PEM}$) is given by the Ordinary Least Squares (OLS) solution,

$$\hat{\theta}_{ML} = \hat{\theta}_{PEM} = [\Phi^\top\Phi]^{-1}\Phi^\top\mathbf{y}, \tag{2.19}$$

where $\Phi^\top = \begin{bmatrix}\phi(1) & \dots & \phi(N)\end{bmatrix}$.

We now evaluate the bias and variance of the above estimator. The bias is the difference between the expected value of the estimate and the true parameter. Therefore,

$$\mathbb{E}\left[\hat{\theta}_{PEM} - \theta_0\right] = \mathbb{E}\left[[\Phi^\top\Phi]^{-1}\Phi^\top\mathbf{y} - \theta_0\right] = \mathbb{E}\left[\theta_0 + [\Phi^\top\Phi]^{-1}\Phi^\top\mathbf{e} - \theta_0\right] = 0. \tag{2.20}$$

Here, $\mathbf{e} = \begin{bmatrix}e(1) & \dots & e(N)\end{bmatrix}^\top$ is the vectorized version of noise. The covariance is given by,

$$\mathbb{E}\left[(\hat{\theta}_{PEM} - \theta_0)(\hat{\theta}_{PEM} - \theta_0)^\top\right] = \sigma^2[\Phi^\top\Phi]^{-1}. \tag{2.21}$$

This covariance will be the CRLB and therefore the smallest variance that can be achieved among all unbiased estimators given the data. However, the estimator can have high variance and provide unsatisfactory results, for example, when the signal-to-noise ratio (SNR) is low or when the data length is small. Now we explore an alternative notion to quantify better models. Focusing on the quantity of Mean-squared error (MSE) of an estimator, we have

$$MSE\left[\hat{\theta}\right] = \mathbb{E}\left[\left\|\hat{\theta} - \theta_0\right\|^2\right] = \mathbb{E}\left[\left\|\hat{\theta} - \mathbb{E}[\hat{\theta}] + \mathbb{E}[\hat{\theta}] - \theta_0\right\|^2\right]$$

$$
\begin{aligned}
&= \mathbb{E}\left[\left\|\hat{\theta}-\mathbb{E}[\hat{\theta}]\right\|^2\right]+\left\|\mathbb{E}[\hat{\theta}-\theta_0]\right\|^2 \\
&= \operatorname{Tr}\left(\operatorname{cov}(\hat{\theta})\right)+\left\|\operatorname{bias}(\hat{\theta})\right\|^2 \\
&= \textbf{VARIANCE}+\textbf{BIAS}^2.
\end{aligned}
$$

The above decomposition of MSE into variance and bias is independent of any choice of estimator. For the above estimate using PEM or ML in (2.19), which is unbiased and efficient, the MSE will be minimum compared to any unbiased estimators since the estimator's variance is minimum (due to CRLB) and the bias is zero. So, this estimator is the best unbiased estimator in the MSE sense. However, relaxing the requirement of unbiasedness, designing an estimator with a smaller mean-squared error is possible. Therefore, we can reduce the variance by trading some bias. We call this as *bias-variance trade off*.

2.6.2 Bayesian estimation and regularized kernel-based methods

The key difference between PEMs or any *frequentist methods* and *Bayesian learning* is that in the latter situation the parameter vector is modeled as a random variable with a prior probability density through which prior belief can be incorporated to the estimation. Infact, how a border line differentiates two countries and sometimes creates war between them, the difference between *frequentist methods* and *Bayesian methods* is also a line between $\mathbf{y}$ and θ (i.e. $p(\mathbf{y}|\theta)$ for Bayesian and $p(\mathbf{y};\theta)$ for frequentist approach).

Let us take the model used in the previous section,

$$
\mathbf{y}=\Phi\theta+\mathbf{e}. \tag{2.22}
$$

Since we are following a Bayesian approach, θ is now a random vector with a prior distribution. The prior distribution is the interesting aspect of the Bayesian perspective. Through this prior distribution, we can encode our prior belief about the system/modules in the network, for example the stability of the modules. Here, θ are the coefficients of the finite impulse response of the system, and we take the prior of θ as a zero-mean Gaussian process i.e.

$$
\theta \sim \mathcal{N}(0,\mathbf{K}).
$$

The prior covariance matrix is called the *kernel* (due to the relation between Gaussian process regression and the theory of reproducing kernel Hilbert space (RKHS), see [107] for details). Assuming that θ and $\mathbf{e}$ are independent, we can write the joint Gaussian model for $\mathbf{y}$ and θ as,

$$
p\left(\begin{bmatrix}\theta\\ \mathbf{y}\end{bmatrix}\right)\sim\mathcal{N}\left(\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix},\begin{bmatrix}\mathbf{K} & \mathbf{K}\Phi^\top\\ \Phi\mathbf{K} & \Phi\mathbf{K}\Phi^\top+\sigma^2 I\end{bmatrix}\right). \tag{2.23}
$$

Then the posterior distribution of θ given the data $\mathbf{y}$ is given by ([2]),

$$p(\theta|\mathbf{y}) \sim \mathcal{N}(\mathbf{C}\mathbf{y}, \mathbf{P}_\theta) \tag{2.24}$$

where

$$\mathbf{P}_\theta = \left(\frac{\Phi^\top\Phi}{\sigma^2} + \mathbf{K}^{-1}\right)^{-1} \quad ; \quad \mathbf{C} = \frac{\mathbf{P}_\theta\Phi^\top}{\sigma^2}.$$

Therefore, the minimum mean-square error (MMSE) estimate is ([2]),

$$\hat{\theta}_{MMSE} = \mathbb{E}\left[\theta|\mathbf{y}\right] = \mathbf{C}\mathbf{y} = \left(\frac{\Phi^\top\Phi}{\sigma^2} + \mathbf{K}^{-1}\right)^{-1} \frac{\Phi^\top}{\sigma^2}\mathbf{y}, \tag{2.25}$$

i.e. the mean of the posterior distribution of θ given the data ($p(\theta|\mathbf{y})$). This estimate is nothing but the regularized least squares estimate with regularization parameter σ^2 and regularization term $\theta^\top\mathbf{K}^{-1}\theta$. This will be immediate when we make the link between $\hat{\theta}_{MMSE}$ and the maximum-a-posteriori (MAP) estimate of θ. The MAP estimate is given by,

$$\begin{aligned}\hat{\theta}_{MAP} &= \arg\max_\theta \log p(\theta|\mathbf{y}) = \arg\max_\theta \left(\log p(\mathbf{y}|\theta) + \log p(\theta) - \log p(\mathbf{y})\right) \\ &= \arg\min_\theta \left[-\log p(\mathbf{y}|\theta) - \log p(\theta)\right]. \end{aligned} \tag{2.26}$$

The above decomposition is done using *Bayes rule* and neglecting the term related to *marginal likelihood* $p(\mathbf{y})$ that does not depend on θ. The decomposition contains the *likelihood* $p(\mathbf{y}|\theta)$ and the prior $p(\theta)$. In the case of a Gaussian prior and Gaussian noise, we get the *regularized least-squares criterion* as following from (2.26),

$$\hat{\theta}_{MAP} = \hat{\theta}_{RLS} = \arg\min_\theta \left[\|\mathbf{y} - \Phi\theta\|^2 + \sigma^2\theta^\top\mathbf{K}^{-1}\theta\right] = \hat{\theta}_{MMSE}, \tag{2.27}$$

where $\|\mathbf{y} - \Phi\theta\|_2^2$ is the least squares term and $\|\theta\|_{\mathbf{K}^{-1}}^2 = \theta^\top\mathbf{K}^{-1}\theta$ is the regularization term.

2.6.3 Priors, Kernels and Empirical Bayes

As already mentioned, we want to trade some bias in order to reduce the variance and in turn the MSE. In order to get a good bias-variance trade off, we need to incorporate our prior knowledge about the system or the quantity to be estimated. As can be seen from the regularization term in (2.27), this can be done by suitably choosing our *kernel* $\mathbf{K}$.

There are many kernels applicable for regularized system identification, like the *stable spline* kernels [92], *DC* kernel [21], *tuned-correlated* (TC) kernel [21]. There are also works on designing kernels for system identification like [19, 20]. Let us consider a kernel (i.e. prior covariance matrix) that will also be used in this thesis, namely the *first-order stable spline* kernel [92] given by,

$$[\mathbf{K}]_{i,j} = \lambda \left[\mathbf{K}_\beta\right]_{i,j} = \lambda \beta^{\max(i,j)},$$

where $\beta \in [0, 1)$ is a *hyperparameter* that regulates the decay rate of the realizations of the corresponding Gaussian vector, while $\lambda \geqslant 0$ tunes their amplitude. The choice of this kernel is motivated by the fact that it enforces favorable properties such as stability and smoothness of the estimated impulse responses [91], [92]. If we follow a full Bayesian approach, these hyperparameters admit a hyperprior density. These hyperprior densities can have hyperparameters. This hierarchy of hyperparameters goes on and results in a *hierarchical model*. In most cases, the hierarchy is ended by using *non-informative priors* or *diffuse priors* (see [10], Section 2.4.3). For example, introducing a prior distribution $p(\beta)$ = constant. However, another possible approach is the *empirical Bayes* approach [80], or generalized maximum likelihood [136]. In the machine learning literature it is also called the evidence approximation (see [10], Section 3.5). We will be using this approach in this thesis. In this approach, we stop the hierarchy at any point we want, by considering the remaining parameters as deterministic *hyperparameters* i.e. they have a fixed value.

How to find out these deterministic hyperparameters' values? There are many possible ways to find it. For example, there are approaches based on cross-validation (CV) or generalized cross-validation [137]. In this thesis, we use another popular approach in the machine learning community, where we find the hyperparameters by maximizing the marginal likelihood function of the data. Considering the *first-order stable spline* kernel which has two hyperparameters λ and β, for the problem in Section 2.6.2, we obtain the hyperparameters by,

$$\hat{\lambda}, \hat{\beta} = \arg\max_{\lambda,\beta} p(\mathbf{y}; \lambda, \beta). \tag{2.28}$$

We maximize the marginal likelihood function i.e. the probability density function of $\mathbf{y}$ after integrating out θ. Therefore, in the empirical Bayes approach, we use the data to fine-tune the prior distribution. This approach of maximizing the marginal likelihood function is also called *type-2 maximum likelihood* [9] due to its resemblance with the frequentist approach of maximum likelihood estimation. Due to this, we can use attractive methods for computing maximum likelihood estimates to compute the hyperparameters. In this thesis, we will be using the Expectation-Maximization (EM) method [35] and the Monte-Carlo Expectation-Maximization (MCEM) method [147] to solve the marginal likelihood problems.

2.6.4 Approximate inference and Gibbs sampling

For many problems related to probabilistic models, it might be required to evaluate the posterior distribution or expectation of a function with respect to a posterior distribution. For example, in the EM algorithm, we need to evaluate the expectation of the complete-data log-likelihood with respect to the posterior distribution of the latent variables. For very few models, the posterior distribution is available in closed form (for example, the posterior distribution of $p(\theta|\mathbf{y})$ in (8.13)). For most probabilistic models of practical interest, exact inference is analytically intractable, and so we have to resort to some form of approximation.

There are two classes of approximation schemes based on whether it relies on stochastic or deterministic approximations. Deterministic approximation methods include *Variational inference* or *Variational Bayes* and *expectation propagation*, which rely of approximating an unknown distribution with tractable density by minimizing a distance measure like Kullback-Leibler (KL) divergence (see [10], Chapter 10). Stochastic approximation methods or Sampling methods are based on numerical sampling. They include methods like *Markov Chain Monte Carlo* (MCMC). Even though there are applications that directly require posterior distributions, posterior distributions are required to evaluate expectations in most applications. If we take the EM algorithm, the posterior distribution is required to evaluate the expectation of the complete-data log-likelihood with respect to the posterior distribution of the latent variables. Therefore the problem is to find the expectation of a function $f(z)$ with respect to a probability distribution $p(z)$.

Sampling methods are used to obatain a set of samples $z^{(l)}$ (where $l = 1, \ldots, M$) drawn independently from the distribution $p(z)$. These samples are used to find an unbiased estimate of the expectation of any function $f(z)$ by a finite sum ,

$$\mathbb{E}[f(z)] \approx \frac{1}{M} \sum_{l=1}^{M} f(z^{(l)}). \tag{2.29}$$

Simple distributions can be sampled using standard methods and inverse transform sampling. However, the posterior distributions that need to be sampled can be complex and high-dimensional. In this thesis, we would like to sample in spaces of high dimensionality. Therefore, we require advanced sampling techniques like Markov Chain Monte Carlo (MCMC). There are various MCMC sampling methods available in the literature. For example the *Metropolis* algorithm [86], the *Metropolis-Hastings algorithm* [62], *Gibbs sampling* [52], the *slice sampling* [87]. In this thesis, we will be using the Gibbs sampling.

Gibbs sampling is a simple and widely applicable MCMC algorithm and it can be seen as a special case of the Metropolis-Hastings algorithm. Gibbs sampling allows us to sample from a joint distribution that is unknown (i.e. $p(\mathbf{z}) = p(z_1, z_2, \ldots, z_n)$) or difficult to sample from directly. Suppose that we have chosen some initial state for the Markov chain, each step of the Gibbs sampling procedure involves replacing the value of one of the variables z_i by a value drawn from the distribution of that variable conditioned on the values of the remaining variables $\mathbf{z}_{\backslash i} = \{z_1, z_2, \ldots, z_n\} \backslash z_i$, i.e., $p(z_i|\mathbf{z}_{\backslash i})$. This procedure is repeated iteratively.

- sample $z_1^{(i+1)} \sim p(z_1|z_2^{(i)}, \ldots, z_n^{(i)})$,

- sample $z_2^{(i+1)} \sim p(z_2|z_1^{(i+1)}, z_3^{(i)}, \ldots, z_n^{(i)})$,

  $\vdots$

- sample $z_n^{(i+1)} \sim p(z_n|z_1^{(i+1)}, z_2^{(i+1)}, \ldots, z_{n-1}^{(i+1)})$.

It can be shown that the sequence of samples constitutes a Markov chain, and the stationary distribution of that Markov chain is the sought-after joint distribution. As the size of i grows, the samples are drawn from the required probability distribution. Normally, we discard the first few samples since the Markov chain will be poorly mixed and the obtained samples will be far away from the stationary distribution, which is the target distribution for the *Gibbs sampler*. Therefore, we discard the first B samples, which is known as the *burn-in* period. If the burn-in period is large enough, then we produce samples from the stationary distribution. It can be seen that in order to sample from a joint distribution, the conditional distribution of each variable should be known. This is a requirement when using Gibbs sampling. Gibbs sampling is a simple and effective sampling method, provided we know the conditional distributions. It does not require any tuning of the proposal distribution like the Metropolis-Hastings algorithm. Similar to the other MCMC techniques, the generated samples can be correlated, but we need independent samples. Instead of sampling individual variables, a group of variables can be sampled to tackle this. This is called *blocking Gibbs sampling* algorithm or *blocked Gibbs sampler* [68]. This is done by choosing blocks of variables, not necessarily disjoint, and then sampling jointly from the variables in each block in turn, conditioned on the remaining variables. We adopt this approach of Gibbs sampling in our thesis.

2.7 Summary

This chapter provides the necessary background and preliminaries. The dynamic network setup is introduced and the state-of-the-art approach for identifying a module in a dynamic network has been provided. A reflection has been made on the available indirect and direct PEMs. Apart from PEM, this thesis follows the approach of Bayesian learning as well. Necessary background on Bayesian estimation, its relation with regularization, imposing prior belief through kernels and estimating the shape of the prior using data has been provided. Solving a marginal likelihood problem involves evaluating posterior densities and expectations with respect to posterior densities. Therefore, the background about the Bayesian inference techniques has also been provided. In the next chapter, we will answer how to handle the confounding variables and correlated noises in the network and obtain asymptotically efficient estimates.

2.8 Related videos

Single module identification in dynamic networks

Karthik R. Ramaswamy

European Control Conference, June 29-July 2 2021, Rotterdam, Netherlands (ECC21)

Tutorial Session: "Data Driven Modeling in Dynamic Networks"

Tutorial Session on Data-Driven Modeling in Dynamic Networks

Single module identification in dynamic networks – the current status

Paul Van den Hof, Karthik Ramaswamy

21st IFAC World Congress, 12-17 July 2020, Berlin, Germany

Open invited track: "Data-driven modeling and learning in dynamic networks"

Single module identification - current status

3 CHAPTER

The Local Direct Method

The identification of local modules in dynamic networks with known topology has recently been addressed by formulating conditions for arriving at consistent estimates of the module dynamics. These results are under the assumption of having disturbances that are uncorrelated over the different nodes and the absence of confounding variables that can occur due to unmeasured nodes. The conditions typically reflect selecting a set of node signals that are taken as predictor inputs in a MISO identification setup. In this chapter an extension is made to arrive at an identification setup for the situation that process noises on the different node signals can be correlated with each other and confounding variables can be present. In this situation the local module may need to be embedded in a MIMO identification setup for arriving at a consistent estimate with maximum likelihood (ML) properties. This requires the proper treatment of confounding variables. The result is a general theory to handle this situation and a set of algorithms that, based on the given network topology and disturbance correlation structure, selects an appropriate set of node signals as predictor inputs and outputs in a MISO or MIMO identification setup. Three algorithms are presented that differ in their approach of selecting measured node signals. Either a maximum or a minimum number of measured node signals can be considered, as well as a preselected set of measured nodes.

This chapter is based on the publication: K.R. Ramaswamy and P.M.J. Van den Hof, "A local direct method for module identification in dynamic networks with correlated noise", *IEEE Trans. Automatic Control*, Vol. 66, no. 11, pp. 5237-5252, November 2021.

3.1 Introduction

A standing assumption in the works [29, 83, 84, 101, 102, 124] is that the process noises entering the nodes of the dynamic network are uncorrelated with each other. This assumption facilitates the analysis and the development of methods for local module identification, reaching *consistent* module estimates using the MISO direct method. In one of the following situations,

a. process noises are correlated over the nodes i.e. $\Phi_v(\omega)$ is non-diagonal, or

b. when some w-in-neighbours of w_j are not included in $w_{\mathcal{D}}$, after immersion,

the disturbance signals that are acting on the inputs and outputs in the predictor model can get correlated, and these disturbance signals are referred to as *confounding variables*[1]. In practical scenarios, these situations are common since not all nodes can be measured, and disturbances in dynamic networks can be correlated. These confounding variables destroy the consistency results for the considered MISO direct method. In these situations, it is necessary to consider also the *noise topology or disturbance correlation structure* when selecting an appropriate identification setup. Even though the indirect and two-stage methods in [27, 54] can handle the situation of correlated noise and deliver consistent estimates, the obtained estimates will not have minimum variance, considering the selected set of signals in the predictor model.

This chapter considers the situation of having dynamic networks with disturbance signals on different nodes that are possibly correlated. At the same time, our target moves from consistency only, to also minimum variance (or Maximum Likelihood (ML)) properties of the obtained local module estimates. We will assume that the topology of the network is known and the (Boolean) correlation structure of the noise disturbances, i.e., the zero-elements in the spectral density matrix of the noise. While one could use techniques for full network identification (e.g., [144]), we aim to develop a method that uses only local information. In this way, we avoid (i) the need to collect node measurements that are "far away" from the target module, and (ii) the need to identify unnecessary modules that would come with the price of the requirement of excess external excitation.

Using the reasoning first introduced in [126], we build a constructive procedure that, choosing a limited number of predictor inputs and predicted outputs, builds an identification setup that guarantees ML properties (and thus asymptotic minimum variance) when applying a direct prediction error identification method. In this situation, we have to deal with so-called *confounding variables* (see e.g., [126], [30]), that is, unmeasured variables that directly or indirectly influence both the predicted output and the predictor inputs, and lead to lack of consistency. The effect of confounding variables will be mitigated by extending the number of predictor inputs and/or predicted outputs in the identification setup, thus including more measured node signals

[1] A confounding variable is an unmeasured variable that has paths to both the input and output of an estimation problem [90].

in the identification. Finally, we provide a generally applicable theory for signal selection and also different node selection schemes.

This chapter is organized as follows. We start with providing a summary of available results and the general idea of the solution to the above problem before getting into technical details. Important concepts and notations used in this chapter are defined in Section 3.4 while the MIMO identification setup and main results are presented in subsequent sections. Sections 3.7-3.9 provide algorithms and illustrative examples for three different ways of selecting input and output node signals: the full input case, the minimum input case, and the user selection case. This is followed by Conclusions. The technical proofs of all results are collected in the Appendix.

3.2 Available results and problem specification

The following results are available from previous work:

- When $\mathcal{D}_j$ is chosen equal to $\mathcal{N}_j$ and noise v_j is uncorrelated to all v_k, $k \in \mathcal{V}_j$, then G_{ji} can be consistently estimated in a MISO setup, provided that there is enough excitation in the predictor input signals, see [124].

- When $\mathcal{D}_j$ is a subset of $\mathcal{N}_j$, and disturbance are uncorrelated, confounding variables can occur in the estimation problem, and these have to be taken into account in the choice of $\mathcal{D}_j$ in order to arrive at consistent estimates of G_{ji}, see [29].

- In [30] relaxed conditions for the selection of $\mathcal{D}_j$ have been formulated, while still staying in the context of MISO identification with noise spectrum of v (Φ_v) being diagonal. This is particularly done by choosing additional predictor input signals that are not in $\mathcal{N}_j$,.i.e. that are no in-neighbors of the output w_j of the target module.

- For non-diagonal Φ_v, an indirect/two-stage identification method can be used to arrive at consistent estimates of G_{ji} [29, 54, 124]. However the drawback of these methods is that they do not allow for a maximum likelihood analysis, i.e. they will not lead to minimum variance results.

- This latter argument also holds for the method in [83, 84], where Wiener-filter estimates are combined to provide local module estimates, and diagonal Φ_v is considered.

In this chapter, we go beyond consistency properties, and address the following problem:

How to handle confounding variables and identify a single module in a dynamic network, such that the estimate is consistent and asymptotically has Maximum Likelihood properties, and thus also minimum variance properties?

3.3 General philosophy of the solution

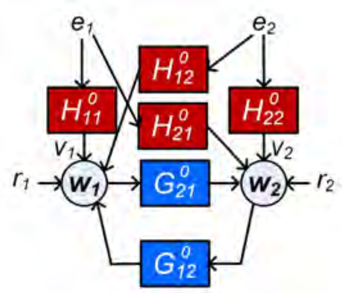


Figure 3.1: Two-node example network from [126] with v_1 and v_2 dynamically correlated and e_1, e_2 white noise processes.

Addressing the above problem requires careful treatment and modeling of the noise acting on the different node signals. This can be illustrated through a simple example that is presented in [126], where a two-node network is considered as given in Figure 3.1, with v_1 and v_2 being dynamically correlated and the objective to identify G_{21}^0. In this case, a SISO identification using the direct method with input w_1 and output w_2 will lead to a biased estimate of G_{21}^0 because of the unmodeled correlation of the disturbance signals on w_1 and w_2[2]. For an analysis of this, see [126]. Suppose both node signals w_1 and w_2 are predicted as outputs. In that case, the correlation between the disturbance signals can be incorporated in a 2 × 2 non-diagonal noise model, thus leading to an unbiased estimate of G_{21}^0. In this way, bias due to correlation in the noise signals can be avoided by predicting additional outputs other than the output of the target module. This leads to the following two suggestions:

- confounding variables can be dealt with by modeling correlated disturbances on the node signals, and
- this can be done by moving from a MISO identification setup to a MIMO setup.

These suggestions are being explored in the current chapter. Next, we will present an example to illustrate the problem further.

Example 3.1 *Consider the network sketched in Figure 3.2, and let module G_{21}^0 be the target module for identification. If the node signals w_1, w_2 and w_4 can be measured, then a two-input one-output model with inputs w_1, w_4 and output w_2 can be considered. This can lead to a consistent estimate of G_{21}^0 and G_{24}^0, provided that the disturbance signal v_2 is uncorrelated to all other disturbance signals. However if e.g. v_4 and v_2 are dynamically correlated,*

[2] In this particular example the bias is caused by the presence of H_{21}.

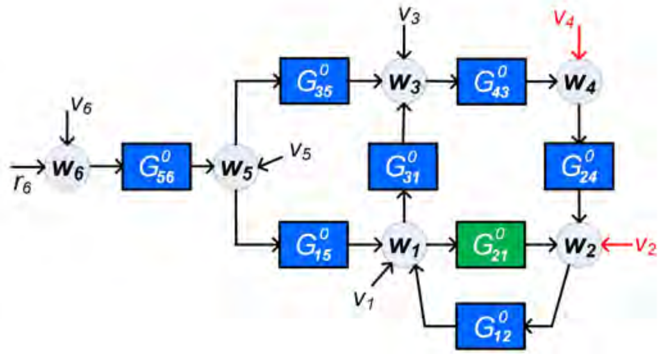


Figure 3.2: Example network with target module G^0_{21} (in green) and with v_2 dynamically correlated with v_4 (red colored).

> *implying that a noise model H of the two-dimensional noise process is non-diagonal, then a biased estimate will result for this approach. A solution is then to include w_4 in the set of predicted outputs, and to add node signal w_3 as predictor input for w_4. We then combine predicting w_2 on the basis of (w_1, w_4) with predicting w_4 on the basis of w_3. The correlation between v_2 and v_4 is then covered by modelling a 2×2 non-diagonal noise model of the joint process (v_2, v_4).*

We will now look into another way of handling confounding variables. Consider the simple cascade three node network example in figure 3.3, with v_1 and v_3 being dynamically correlated. In this case, a SISO identification using w_2 as input and w_1 as output will lead to biased estimate of G^0_{12} due to unmodelled correlation of the disturbance signals on w_1 and w_2. Even though the direct disturbance on w_2 (i.e. v_2) is uncorrelated to v_1, there exists correlation of the disturbance signals on w_1 and w_2 due to the fact that v_3 and v_1 are correlated and v_3 has a path to w_2 through w_3 which is an *unmeasured node* (i.e. a node signal that is not in the SISO predictor model/identification). One possible way to handle this situation, which

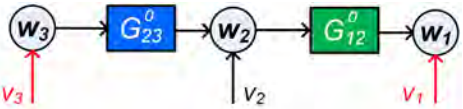


Figure 3.3: Simple example network with target module G^0_{12} (in green) and with v_1 dynamically correlated with v_3 (red colored).

we have already discussed, is to predict w_2 as well (i.e., add w_2 to the predicted output) and model the correlated disturbances using a 2×2 non-diagonal noise model. Another way to handle this is to include w_3 as predictor input along with w_2 as input and w_1 as output, and block the effect of confounding variable that affects the estimation of target module G^0_{12}. What we are doing here is *block and divert*. Here, we are blocking the effect of the confounding variable on the target module and diverting the effect of the confounding variable to another estimated module. In essence, the unmodelled correlation between v_1 and v_3 will create bias in the estimated model from w_3 to w_1 (i.e. the model will not be zero), however it will not create bias in the estimated model from w_2 to w_1 (which is the target module).

In the next sections we will formalize the procedure that has been discussed above for general networks.

3.4 Concepts and notation

In line with [90] we define the notion of a confounding variable.

Definition 3.1 (confounding variable) *Consider a dynamic network defined by*

$$w = G^0 w + H^0 e + R^0 r \tag{3.1}$$

with e a white noise process, and consider the graph related to this network, with node signals w and e. Let $w_{\mathcal{X}}$ and $w_{\mathcal{Y}}$ be two subsets of measured node signals in w, and let $w_{\mathcal{Z}}$ be the set of unmeasured node signals in w. Then a noise component e_ℓ in e is a confounding variable for the estimation problem $w_{\mathcal{X}} \rightarrow w_{\mathcal{Y}}$, if in the graph there exist simultaneous paths from e_ℓ to node signals $w_k, k \in \mathcal{X}$ and $w_n, n \in \mathcal{Y}$, while these paths are either direct or only pass through nodes that are in $w_{\mathcal{Z}}$. □

We will denote $w_{\mathcal{Y}}$ as the node signals in w that serve as predicted outputs, and $w_{\mathcal{D}}$ as the node signals in w that serve as predictor inputs. It can happen that a node signal can be in both input and output. Next we decompose $w_{\mathcal{Y}}$ and $w_{\mathcal{D}}$ into disjoint sets according to: $\mathcal{Y} = \mathcal{Q} \cup \{o\}$; $\mathcal{D} = \mathcal{Q} \cup \mathcal{U}$ where $w_{\mathcal{Q}}$ are the node signals that are common in $w_{\mathcal{Y}}$ and $w_{\mathcal{D}}$; w_o is the output w_j of the target module; if $j \in \mathcal{Q}$ then $\{o\}$ is void; $w_{\mathcal{U}}$ are the node signals that are only in $w_{\mathcal{D}}$. In this situation the measured nodes will be $w_{\mathcal{D} \cup \mathcal{Y}}$ and the unmeasured nodes $w_{\mathcal{Z}}$ will be determined by the set $\mathcal{Z} = \mathcal{L} \backslash \{\mathcal{D} \cup \mathcal{Y}\}$, where $\mathcal{L} = \{1, 2, \cdots L\}$. There can exist two types of confounding variable namely *direct and indirect confounding variables*. For *direct confounding variables* the simultaneous paths mentioned in the definition are both *direct paths*, while in all other cases we refer to the confounding variables as *indirect confounding variables*. For example, in the network as shown in Figure 3.4 with $\mathcal{D} = \{2\}$, $\mathcal{Y} = \{1\}$ and $\mathcal{Z} = \{3\}$, for the estimation problem $w_2 \rightarrow w_1$, e_2 is a *direct confounding variable* since it has a simultaneous path to w_1 and w_2 where both

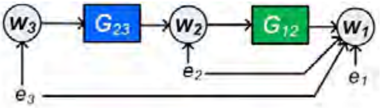


Figure 3.4: A simple network with 3 nodes w_1, w_2, w_3 and unmeasured noise sources e_1, e_2 and e_3. G_{12} is the target module to be identified.

the paths are *direct paths*. Meanwhile e_3 is an *indirect confounding variable* since it has a simultaneous path to w_1 and w_2 where one of the path is an unmeasured path[3].

> **Remark 3.1** *Confounding variables are defined in accordance with their use in [30], on the basis of a network description as in (3.1). In this definition absence of confounding variables still allows that there are unmeasured signals that create correlation between the inputs and outputs of an estimation problem, in particular if the white noise signals in e are statically correlated, i.e $cov(e)$ being non-diagonal. It will appear that this type of correlations will not hinder our identification results, as analysed in Section 9.5.*

3.5 Main results - Line of reasoning

On the basis of the decomposition of node signals as defined in the previous section we are going to represent the system's equations (3.1) in the following structured form[4]:

$$\begin{aligned}\begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \\ w_{\mathcal{Z}} \end{bmatrix} &= \begin{bmatrix} G_{\mathcal{Q}\mathcal{Q}} & G_{\mathcal{Q}o} & G_{\mathcal{Q}\mathcal{U}} & G_{\mathcal{Q}\mathcal{Z}} \\ G_{o\mathcal{Q}} & G_{oo} & G_{o\mathcal{U}} & G_{o\mathcal{Z}} \\ G_{\mathcal{U}\mathcal{Q}} & G_{\mathcal{U}o} & G_{\mathcal{U}\mathcal{U}} & G_{\mathcal{U}\mathcal{Z}} \\ G_{\mathcal{Z}\mathcal{Q}} & G_{\mathcal{Z}o} & G_{\mathcal{Z}\mathcal{U}} & G_{\mathcal{Z}\mathcal{Z}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \\ w_{\mathcal{Z}} \end{bmatrix} + R(q)r \\ &+ \begin{bmatrix} H_{\mathcal{Q}\mathcal{Q}} & H_{\mathcal{Q}o} & H_{\mathcal{Q}\mathcal{U}} & H_{\mathcal{Q}\mathcal{Z}} \\ H_{o\mathcal{Q}} & H_{oo} & H_{o\mathcal{U}} & H_{o\mathcal{Z}} \\ H_{\mathcal{U}\mathcal{Q}} & H_{\mathcal{U}o} & H_{\mathcal{U}\mathcal{U}} & H_{\mathcal{U}\mathcal{Z}} \\ H_{\mathcal{Z}\mathcal{Q}} & H_{\mathcal{Z}o} & H_{\mathcal{Z}\mathcal{U}} & H_{\mathcal{Z}\mathcal{Z}} \end{bmatrix} \begin{bmatrix} e_{\mathcal{Q}} \\ e_o \\ e_{\mathcal{U}} \\ e_{\mathcal{Z}} \end{bmatrix}\end{aligned} \tag{3.2}$$

where we make the notation agreement that the matrix H is not necessarily monic, and the scaling of the white noise process e is such that $cov(e) = I$. Without loss of generality, we can assume $r = 0$ for the sake of brevity.

Our objective is to end up with an identification problem in which we identify the dynamics from inputs $(w_{\mathcal{Q}}, w_{\mathcal{U}})$ to outputs $(w_{\mathcal{Q}}, w_o)$, while our target module $G_{ji}(q)$ is present as one of the scalar transfers (modules) in this identified (MIMO) model. This can be realized by the following steps:

1. Firstly, we write the system's equations for the measured variables as

$$\underbrace{\left[\begin{array}{c} w_{\mathcal{Q}} \\ w_o \\ \hline w_{\mathcal{U}} \end{array}\right]}_{w_m} = \underbrace{\left[\begin{array}{c|c} \bar{G} & 0 \\ \hline \bar{G}_{\mathcal{U}\mathcal{D}} & \bar{G}_{\mathcal{U}o} \end{array}\right]}_{\bar{G}_m} \left[\begin{array}{c} w_{\mathcal{Q}} \\ w_{\mathcal{U}} \\ \hline w_o \end{array}\right] + \underbrace{\left[\begin{array}{c|c} \bar{H} & 0 \\ \hline 0 & \bar{H}_{\mathcal{U}\mathcal{U}} \end{array}\right]}_{\bar{H}_m} \underbrace{\left[\begin{array}{c} \xi_{\mathcal{Q}} \\ \xi_o \\ \hline \xi_{\mathcal{U}} \end{array}\right]}_{\xi_m} \tag{3.3}$$

[3] An unmeasured path is a path that passes through nodes in $w_{\mathcal{Z}}$ only. Analogously, we can define unmeasured loops through a node w_k.

[4] From now on, ^{0} is dropped for convenience.

with ξ_m a white noise process, while $\bar{H}$ is monic, stable and stably invertible and the components in $\bar{G}$ are zero if it concerns a mapping between identical signals. This step is made by removing the non-measured signals $w_{\mathcal{Z}}$ from the network, while maintaining the second order properties of the remaining signals. This step is referred to as immersion of the nodes in $w_{\mathcal{Z}}$ [29].

2. As an immediate result of the previous step we can write an expression for the output variables $w_{\mathcal{Y}}$, by considering the upper part of the equation (3.3), as

$$\underbrace{\begin{bmatrix} w_{\mathcal{Q}} \\ w_o \end{bmatrix}}_{w_{\mathcal{Y}}} = \underbrace{\begin{bmatrix} \bar{G}_{\mathcal{Q}\mathcal{Q}} & \bar{G}_{\mathcal{Q}\mathcal{U}} \\ \bar{G}_{o\mathcal{Q}} & \bar{G}_{o\mathcal{U}} \end{bmatrix}}_{\bar{G}} \underbrace{\begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{U}} \end{bmatrix}}_{w_{\mathcal{D}}} + \underbrace{\begin{bmatrix} \bar{H}_{\mathcal{Q}\mathcal{Q}} & \bar{H}_{\mathcal{Q}o} \\ \bar{H}_{o\mathcal{Q}} & \bar{H}_{oo} \end{bmatrix}}_{\bar{H}} \underbrace{\begin{bmatrix} \xi_{\mathcal{Q}} \\ \xi_o \end{bmatrix}}_{\xi_{\mathcal{Y}}} \tag{3.4}$$

with $cov(\xi_{\mathcal{Y}}) := \bar{\Lambda}$.

3. Thirdly, we will provide conditions to guarantee that $\bar{G}_{ji}(q) = G_{ji}(q)$, i.e the target module appearing in equation (3.4) is the target module of the original network *(invariance of target module)*. This will require conditions on the selection of node signals in $w_{\mathcal{Q}}, w_o, w_{\mathcal{U}}$.

4. Finally, it will be shown that, on the basis of (3.4), under fairly general conditions, the transfer functions $\bar{G}(q)$ and $\bar{H}(q)$ can be estimated consistently, and with maximum likelihood properties. A pictorial representation of the identification setup with the classification of different sets of signals in (3.4) is provided in Figure 3.5. The figure also contains set $\mathcal{A}, \mathcal{B}, \mathcal{F}_n$ which will be introduced in the sequel.

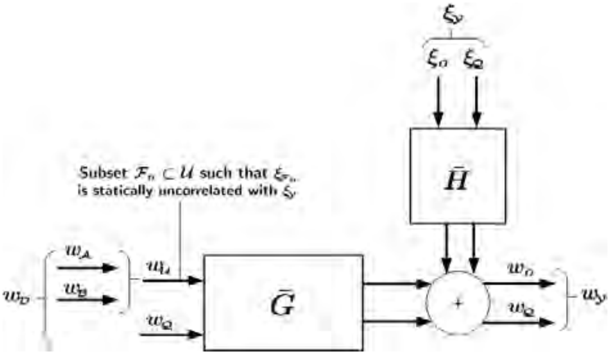


Figure 3.5: Figure to depict the identification setup and classification of different sets of signals in the input and output of the identification problem.

The combination of steps 3 and 4 will lead to a consistent and maximum likelihood estimation of the target module $G_{ji}(q)$. It has to be noted that an identification setup results, in which signals can simultaneously act as input and as output (the set $w_{\mathcal{Q}}$). Because $\bar{G}_{\mathcal{Q}\mathcal{Q}}$ is restricted to be hollow, this does not lead to trivial transfers between signals that are the same. A related situation appears when identifying a full network, while using all node signals as both inputs and outputs, as in [144].

The steps 1)-4) above will require conditions on the selection of node signals, based on the known topology of the network and an allowed correlation structure of the disturbances in the network. Specifying these conditions on the selection of sets $w_{\mathcal{Q}}, w_o, w_{\mathcal{U}}$, will be an important objective of the next section.

3.6 Main Results - Derivations

3.6.1 System representation after immersion (Steps 1-2)

First we will show that a network in which signals in $w_{\mathcal{Z}}$ are removed (immersed) can indeed be represented by (3.3).

Proposition 3.1 *Consider a dynamic network given by (3.2), where the set of all nodes $w_{\mathcal{L}}$ is decomposed in disjunct sets $w_{\mathcal{Q}}$, w_o, $w_{\mathcal{U}}$ and $w_{\mathcal{Z}}$ as defined in Section 3.4. Then, for the situation $r = 0$,*

1. *there exists a representation (3.3) of the measured node signals w_m, with $\bar{H}_m$ monic, stable and stably invertible, and ξ_m a white noise process, and*
2. *for this representation there are no confounding variables for the estimation problem $w_{\mathcal{U}} \rightarrow w_{\mathcal{Y}}$.*

Proof: See appendix. ■

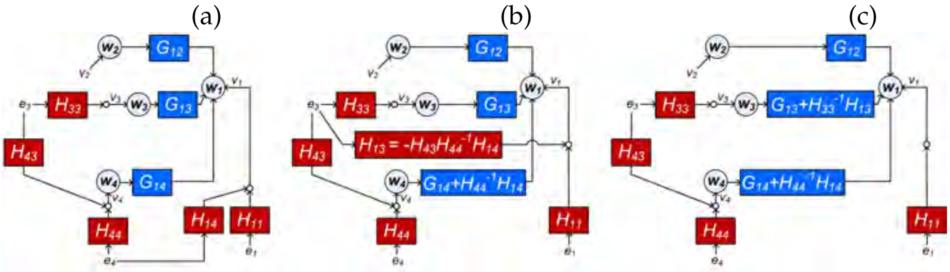


Figure 3.6: (a): Original network with 4 nodes $\{w_i\}_{i=1,\cdots 4}$, and unmeasured white noise sources $\{e_i\}_{i=1,\cdots 4}$; (b): Transformed network with confounding variable for $w_4 \rightarrow w_1$ removed; (c): Transformed network with also the confounding variable for $w_3 \rightarrow w_1$ removed.

The consequence of Proposition 3.1 is that the output node signals in $w_{\mathcal{Y}}$ can be explicitly written in the form of (3.4), in terms of input node signals $w_{\mathcal{D}}$ and disturbances, without relying on (unmeasured) node signals in $w_{\mathcal{Z}}$. The particular structure of network representation (3.3) implies that there are no confounding variables for the estimation problem $w_{\mathcal{U}} \rightarrow w_{\mathcal{Y}}$. This will be an important phenomenon for our identification setup. Based on (3.4), a typical prediction error identification method can provide estimates of $\bar{G}$ and $\bar{H}$ from measured signals $w_{\mathcal{Y}}$ and $w_{\mathcal{D}}$ with $\mathcal{D} = \mathcal{Q} \cup \mathcal{U}$. In this estimation problem, confounding variables for the estimation problem $w_{\mathcal{Q}} \rightarrow w_{\mathcal{Y}}$ are treated by correlated noise modelling in $\bar{H}$, while confounding variables for the estimation problem $w_{\mathcal{U}} \rightarrow w_{\mathcal{Y}}$ are not present, due to the structure of (3.3).

In the following example, the step towards (3.3) will be illustrated, as well as its effect on the dynamics in $\bar{G}$.

Example 3.2 *Consider the 4-node network depicted in Figure 3.6(a), where all nodes are considered to be measured, and where we select $w_o = w_1$, $\mathcal{U} = \{2,3,4\}$, and $\mathcal{Q} = \emptyset$. In this network, there is a confounding variable e_4 for the problem $w_4 \rightarrow w_1$ (i.e $w_{\mathcal{U}} \rightarrow w_{\mathcal{Y}}$), meaning that for the situation $\xi = e$ the noise model $\bar{H}_m$ in (3.3) will not be block diagonal. Therefore the network does not comply with the representation in* (3.3) *and* (3.4). *We can remove the confounding variable, by shifting the effect of H_{14} into a transformed version of G_{14}, which now becomes $G_{14} + H_{44}^{-1}H_{14}$, as depicted in Figure 3.6(b). However, since this shift also affects the transfer from e_3 to w_1, the change of G_{14} needs to be mitigated by a new term H_{13}, in order to keep the network signals invariant. In the resulting network the confounding variable for $w_4 \rightarrow w_1$ is removed, but a new confounding variable (e_3) for $w_3 \rightarrow w_1$ has been created. In the second step, shown in Figure 3.6(c), the term H_{13} is removed by incorporating its effect in the module G_{13} which now becomes $G_{13} + H_{33}^{-1}H_{13}$. In the resulting network there are no confounding variables for $w_{\mathcal{U}} \rightarrow w_1$. This representation complies with the structure in (3.3). Note that in the transformed network, the dynamics of G_{12} is left invariant, while the dynamics of G_{14} and G_{13} have been changed. The intermediately occurring confounding variables relate to a sequence of linked confounders, as discussed in [30].* □

In the next subsection it will be investigated under which conditions our target module will remain invariant under the above transformation to a representation (3.3) without confounding variables.

3.6.2 Module invariance result (Step 3)

The transformation of a network into the form (3.3), leading to the resulting identification setup of (3.4), involves two basic steps, each of which can lead to a change of dynamic modules in $\bar{G}$. These two steps are

(a) Removing of non-measured signals in $w_{\mathcal{Z}}$ (immersion), and

(b) Transforming the system's equations to a form where there are no confounding variables for $w_{\mathcal{U}} \rightarrow w_{\mathcal{Y}}$.

Module invariance in step (a) is covered by the following Condition:

Condition 3.1 (parallel path and loop condition[29]) *Let G_{ji} be the target network module to be identified. In the original network (3.2):*

- *Every path from w_i to w_j, excluding the path through G_{ji}, passes through a node $w_k, k \in \mathcal{D}$, and*
- *Every loop through w_j passes through a node in $w_k, k \in \mathcal{D}$.* □

This condition has been introduced in [29] for a MISO identification setup, to guarantee that when immersing (removing) nonmeasured node signals from the network, the target module will remain invariant. As an alternative, more generalized notions of network abstractions have been developed for this purpose in [141]. Condition 3.1 will be used to guarantee module invariance under step (a).

Step (b) above is a new step, and requires studying module invariance in the step transforming a network from an original format where all nodes are measured, into a structure that complies with (3.3), i.e. with absence of confounding variables for $w_{\mathcal{U}} \rightarrow w_{\mathcal{Y}}$.

We are going to tackle this problem, by decomposing the set $\mathcal{U}$ into two disjunct sets $\mathcal{U} = \mathcal{A} \cup \mathcal{B}$ aiming at the situation that in the transformed network, the modules $G_{\mathcal{Y}\mathcal{A}}$ stay invariant, while for the modules $G_{\mathcal{Y}\mathcal{B}}$ we accept that the transformation can lead to module changes. We construct $\mathcal{A}$ by choosing signals $w_k \in w_{\mathcal{U}}$ such that in the original network there are no confounding variables for the estimation problem $w_{\mathcal{A}} \rightarrow w_{\mathcal{Y}}$. For the selection of $\mathcal{B}$, we do allow confounding variables for the estimation problem $w_{\mathcal{B}} \rightarrow w_{\mathcal{Y}}$. By requiring a particular "disconnection" between the sets $\mathcal{A}$ and $\mathcal{B}$, we can then still guarantee that the modules $G_{\mathcal{Y}\mathcal{A}}$ stay invariant.

The following condition will address the major requirement for addressing our step (b).

Condition 3.2 *$\mathcal{U}$ is decomposed into two disjunct sets, $\mathcal{U} = \mathcal{A} \cup \mathcal{B}$ (see Figure 3.5), such that* in the original network *(3.2) there are no confounding variables for the estimation problems $w_{\mathcal{A}} \rightarrow w_{\mathcal{Y}}$ and $w_{\mathcal{A}} \rightarrow w_{\mathcal{B}}$.* □

Condition 3.2 is not a restriction on $\mathcal{U}$, as such a decomposition can always be made, e.g. by taking $\mathcal{A} = \emptyset$ and $\mathcal{B} = \mathcal{U}$. The flexibility in choosing this decomposition will be instrumental in the sequel of this chapter.

Example 3.3 (Example 3.2 continued) *In the example network depicted in Figure 3.6, we observe that in the original network there is a confounding variable for $w_4 \rightarrow w_1$. However in the step towards creating a network without confounding variables for $w_{\mathcal{U}} \rightarrow w_{\mathcal{Y}}$ an intermediate step occurs, where there is also a confounding variable for $w_3 \rightarrow w_1$, as depicted in Figure 3.6(b). For $\mathcal{U} = \{2, 3, 4\}$ the choice $\mathcal{A} = \{2, 3\}$, $\mathcal{B} = \{4\}$, is not valid* *since there exists a confounding variable (e_3) for $w_3 \rightarrow w_4$ which violates the second condition that there should be no confounding variables for $w_\mathcal{A} \rightarrow w_\mathcal{B}$. Therefore the appropriate choice satisfying Condition 3.2 is $\mathcal{A} = \{2\}$ and $\mathcal{B} = \{3, 4\}$. Note that this matches with the situation that in the transformed network (Figure 3.6(c)), the module $G_{\mathcal{YA}}$ remains invariant, and the modules $G_{\mathcal{YB}}$ get changed.* □

We can now formulate the module invariance result.

Theorem 3.1 (Module invariance result) *Let G_{ji} be the target network module. In the transformed system's equation (3.4), it holds that $\bar{G}_{ji} = G^0_{ji}$ under the following conditions:*

1. *The parallel path and loop Condition 3.1 is satisfied, and*
2. *The following three conditions are satisfied:*
   a. *$\mathcal{U}$ is decomposed in $\mathcal{A}$ and $\mathcal{B}$, satisfying Condition 3.2, and*
   b. *$i \in \{\mathcal{A} \cup \mathcal{Q}\}$, and*
   c. *Every path from $\{w_i, w_j\}$ to $w_\mathcal{B}$ passes through a measured node in $w_{\mathcal{L}\setminus\mathcal{Z}}$.*

Proof: See appendix.

A more detailed illustration of the conditions in the theorem will be deferred to three different algorithms for selecting the node signals, to be presented in Sections 3.7-3.9. We will first develop the identification results for the general case.

3.6.3 Identification results (Step 4)

If the conditions of Theorem 3.1 are satisfied, then the target module $\bar{G}_{ji} = G^0_{ji}$ can be identified on the basis of the system's equation (3.4). For this system's equation we can set up a predictor model with input $w_\mathcal{D}$ and outputs $w_\mathcal{Y}$, for the estimation of $\bar{G}$ and $\bar{H}$. This will be based on a parameterized model set determined by

$$\mathcal{M} := \left\{(\bar{G}(\theta), \bar{H}(\theta), \bar{\Lambda}(\theta)), \theta \in \Theta\right\},$$

while the actual data generating system is represented by $\mathcal{S} = (\bar{G}(\theta_o), \bar{H}(\theta_o), \bar{\Lambda}(\theta_0))$. The corresponding identification problem is defined by considering the one-step-ahead prediction of $w_\mathcal{Y}$ in the parametrized model, according to $\hat{w}_\mathcal{Y}(t|t-1;\theta) := \mathbb{E}\{w_\mathcal{Y}(t) \mid w_\mathcal{Y}^{t-1}, w_\mathcal{D}^t; \theta\}$ where $w_\mathcal{D}^t$ denotes the past of $w_\mathcal{D}$, i.e. $\{w_\mathcal{D}(k), k \leqslant t\}$. The resulting prediction error becomes: $\varepsilon(t,\theta) := w_\mathcal{Y}(t) - \hat{w}_\mathcal{Y}(t|t-1;\theta)$, leading to

$$\varepsilon(t,\theta) = \bar{H}(q,\theta)^{-1}\left[w_\mathcal{Y}(t) - \bar{G}(q,\theta)w_\mathcal{D}(t)\right], \tag{3.5}$$

and the weighted least squares identification criterion

$$\hat{\theta}_N = \arg\min_{\theta} \frac{1}{N} \sum_{t=0}^{N-1} \varepsilon^T(t,\theta) W \varepsilon(t,\theta), \tag{3.6}$$

with W any positive definite weighting matrix. This parameter estimate then leads to an estimated subnetwork $\bar{G}_{\mathcal{YD}}(q,\hat{\theta}_N)$ and noise model $\bar{H}(q,\hat{\theta}_N)$, for which consistency and minimum variance results will be formulated next.

Theorem 3.2 (Consistency) *Consider a dynamic network represented by (3.3), and a related (MIMO) network identification setup with predictor inputs $w_{\mathcal{D}}$ and predicted outputs $w_{\mathcal{Y}}$, according to (3.4). Let $\mathcal{F}_n \subseteq \mathcal{U}$ be the set of node signals k for which ξ_k is statically uncorrelated with $\xi_{\mathcal{Y}}$*[a] *and let $\mathcal{F} := \mathcal{U}\backslash\mathcal{F}_n$. Then a direct prediction error identification method according to (4.5)-(4.6), applied to a parametrized model set $\mathcal{M}$ will provide consistent estimates of $\bar{G}$ and $\bar{H}$ if:*

a. $\mathcal{M}$ is chosen to satisfy $\mathcal{S} \in \mathcal{M}$;

b. $\Phi_\kappa(\omega) > 0$ for a sufficiently high number of frequencies, where $\kappa(t) := \begin{bmatrix} w_{\mathcal{D}}^\top(t) & \xi_{\mathcal{Q}}^\top(t) & w_o(t) \end{bmatrix}^\top$;
(data-informativity condition).

c. The following paths/loops should have at least a delay:

- *All paths/loops from $w_{\mathcal{Y}\cup\mathcal{F}}$ to $w_{\mathcal{Y}}$ in the network (3.4) and in its parametrized model; and*
- *For every $w_k \in \mathcal{F}_n$, all paths from $w_{\mathcal{Y}\cup\mathcal{F}}$ to w_k in the network (3.4), or all paths from w_k to $w_{\mathcal{Y}}$ in the parametrized model.*

(delay in path/loop condition.)

[a] This implies that $\mathbb{E}[\xi_k(t)\xi_{\mathcal{Y}}(t)] = 0$.

Proof: See appendix.

The consistency theorem has a structure that corresponds to the classical result of the direct prediction error identification method applied to a closed-loop experimental setup, [77]. A system in the model set condition (a), an informativity condition on the measured data (b), and a loop delay condition (c). Note however that conditions (b) and (c) are generalized versions of the typical closed-loop case [77, 124], and are dedicated for the considered network setup.

It is important to note that Theorem 3.2 is formulated in terms of conditions on the network in (3.3), which we refer to as the *transformed network*. However, it is quintessential to formulate the conditions in terms of properties of signals in the *original network*, represented by (3.2).

Proposition 3.2 *If in the original network, $\mathcal{U}$ is decomposed in two disjunct sets $\mathcal{A}$ and $\mathcal{B}$ satisfying Condition 3.2, then Condition c of Theorem 3.2 can be reformulated as:*

c. The following paths/loops should have at least a delay:

- *All paths/loops from $w_{\mathcal{Y}\cup\mathcal{B}}$ to $w_{\mathcal{Y}}$ in the original network (3.2) and in the parametrized model; and*
- *For every $w_k \in \mathcal{A}$, all paths from $w_{\mathcal{Y}\cup\mathcal{B}}$ to w_k in the network (3.2), or all paths from w_k to $w_{\mathcal{Y}}$ in the parametrized model.*

Proof: See appendix.

Condition (b) of Theorem 3.2 requires that there should be enough excitation present in the node signals, which actually reflects a type of identifiability property [143]. Note that this excitation condition may require that there are external excitation signals present at some locations, see also [22, 53, 65, 124, 133, 145], and [127], where it is shown that $dim(r) \geqslant |\mathcal{Q}|$, with $|\mathcal{Q}|$ the cardinality of $\mathcal{Q}$. Since we are using a direct method for identification, excitation signals r are not directly used in the predictor model, although they serve the purpose of providing excitation in the network. A first result of a generalized method where, besides node signals w, also signals r are included in the predictor inputs, is presented in [105].

Since in the result of Theorem 3.2 we arrive at white innovation signals, the result can be extended to formulate Maximum Likelihood properties of the estimate.

Theorem 3.3 *Consider the situation of Theorem 3.2, and let the conditions for consistency be satisfied. Let $\xi_{\mathcal{Y}}$ be normally distributed, and let $\bar{\Lambda}(\theta)$ be parametrized independently from $\bar{G}(\theta)$ and $\bar{H}(\theta)$. Then, under zero initial conditions, the Maximum Likelihood estimate of θ^0 is*

$$\hat{\theta}_N^{ML} = \arg\min_{\theta} \det\left(\frac{1}{N}\sum_{t=1}^{N} \varepsilon(t,\theta)\varepsilon^T(t,\theta)\right) \tag{3.7}$$

$$\Lambda(\hat{\theta}_N^{ML}) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon(t,\hat{\theta}_N^{ML})\varepsilon^T(t,\hat{\theta}_N^{ML}). \tag{3.8}$$

Proof: Can be shown by following a similar reasoning as in Theorem 1 of [144]. □

So far, we have analysed the situation for given sets of node signals $w_{\mathcal{Q}}$, w_o, $w_{\mathcal{A}}$, $w_{\mathcal{B}}$ and $w_{\mathcal{Z}}$. The presented results are very general and allow for different algorithms to select the appropriate signals and specify the particular signal sets, that will guarantee target module invariance and consistent and minimum variance module estimates with the presented local direct method. In the next sections we will focus on formulating guidelines for the selection of these sets, such that the target module invariance property holds, as formulated in Theorem

3.1. For formulating these conditions, we will consider three different situations with respect to the availability of measured node signals.

(a) In the *Full input case*, we will assume that all in-neighbors of the predicted output signals are measured and used as predictor input;

(b) In the *Minimum input case*, we will include the smallest possible number of node signals to be measured for arriving at our objective;

(c) In the *User selection case*, we will formulate our results for a prior given set of measured node signals;

3.7 Algorithm for signal selection: full input case

The first algorithm to be presented is based on the strategy that for any node signal that is selected as output, we have access to all of its w-in-neighbors, that are to be included as predictor inputs. This strategy will lead to an identification setup with a maximum use of measured node signals that contain information that is relevant for modeling our target module G_{ji}. The following strategy will be followed:

- We start by selecting $i \in \mathcal{D}$ and $j \in \mathcal{Y}$;
- Then we extend $\mathcal{D}$ in such a way that all w-in-neighbors of $w_{\mathcal{Y}}$ are included in $w_{\mathcal{D}}$.
- All node signals in $w_{\mathcal{D}}$ that have noise terms v_k, $k \in \mathcal{D}$ that are correlated with any v_ℓ, $\ell \in \mathcal{Y}$ (*direct* confounding variables for $w_{\mathcal{D}} \rightarrow w_{\mathcal{Y}}$), are included in $\mathcal{Y}$ too. They become elements of $\mathcal{Q}$.
- With $\mathcal{A} := \mathcal{D} \backslash \mathcal{Q}$ it follows that by construction there are no *direct* confounding variables for the estimation problem $w_{\mathcal{A}} \rightarrow w_{\mathcal{Y}}$.
- Then we choose $w_{\mathcal{B}}$ as a subset of nodes that are not in $w_{\mathcal{Y}}$ nor in $w_{\mathcal{A}}$. This set needs to be introduced to block the *indirect* confounding variables for the estimation problem $w_{\mathcal{A}} \rightarrow w_{\mathcal{Y}}$, and will be chosen to satisfy Condition 2a and 2c of Theorem 3.1.
- Every node signal w_k, $k \in \mathcal{A}$ for which there are only indirect confounding variables and cannot be blocked by a node in $w_{\mathcal{B}}$, is
  - moved to $\mathcal{B}$ if Conditions 2a and 2c of Theorem 3.1 are satisfied and $k \neq i$; (else)
  - included in $\mathcal{Y}$ and moved to $\mathcal{Q}$;
- Finally, we define the identification setup as the estimation problem $w_{\mathcal{D}} \rightarrow w_{\mathcal{Y}}$, with $\mathcal{D} = \mathcal{Q} \cup \mathcal{A} \cup \mathcal{B}$ and $\mathcal{Y} = \mathcal{Q} \cup \{o\}$.

Note that because all w-in-neighbors of $w_{\mathcal{Y}}$ are included in $w_{\mathcal{D}}$, we automatically satisfy the parallel path and loop condition 3.1. In order for the selection of node signals $w_{\mathcal{B}}$ to satisfy the conditions of Theorem 3.1, we will specify the following Property 3.1.

Property 3.1 *Let the node signals $w_{\mathcal{B}}$ be chosen to satisfy the following properties:*

1. *If, in the original network, there are no confounding variables for the estimation problem $w_{\mathcal{A}} \rightarrow w_{\mathcal{Y}}$, then $\mathcal{B}$ is void implying that $w_{\mathcal{B}}$ is not present;*
2. *If, in the original network, there are confounding variables for the estimation problem $w_{\mathcal{A}} \rightarrow w_{\mathcal{Y}}$, then all of the following conditions need to be satisfied:*
   a. *For any confounding variable for the estimation problem $w_{\mathcal{A}} \rightarrow w_{\mathcal{Y}}$, the unmeasured paths from the confounding variable to node signals $w_{\mathcal{A}}$ pass through a node in $w_{\mathcal{B}}$.*
   b. *There are no confounding variables for the estimation problem $w_{\mathcal{A}} \rightarrow w_{\mathcal{B}}$.*
   c. *Every path from $\{w_i, w_j\}$ to $w_{\mathcal{B}}$ passes through a measured node in $w_{\mathcal{L}\setminus\mathcal{Z}}$.* □

Property 2a) ensures that, after including $w_{\mathcal{B}}$ in the set of measured signals, there are no *indirect* confounding variables for the estimation problem $w_{\mathcal{A}} \rightarrow w_{\mathcal{Y}}$, and Property 2b) guarantees that there are no confounding variables for the estimation problem $w_{\mathcal{A}} \rightarrow w_{\mathcal{B}}$. Together we satisfy Condition 2a) of Theorem 3.1. Also, Property 2c) guarantees condition 2c) of Theorem 3.1 to be satisfied. Finally, as per the algorithm, w_i can be either in $w_{\mathcal{A}}$ or $w_{\mathcal{Q}}$. Therefore at the end of the algorithm, we will obtain sets of signals that satisfy the conditions in Theorem 3.1 for target module invariance.

Example 3.4 *Consider the network in Figure 3.7. G_{12} is the target module that we want to identify. We now select the signals according to the algorithm presented in this section. First we include the input of the target module w_2 in $w_{\mathcal{D}}$ and the output of the target module w_1 in $w_{\mathcal{Y}}$. Next we include all w-in-neighbors of $w_{\mathcal{Y}}$ (i.e. w_3 and w_4) in $w_{\mathcal{D}}$. All node signals in $w_{\mathcal{D}}$ that have noise terms v_k, $k \in \mathcal{D}$ that are correlated with any $v_\ell, \ell \in \mathcal{Y}$ need to be included in $\mathcal{Y}$ too. This concerns w_2, since v_1 is correlated with v_2. Now $w_{\mathcal{Y}} = \{w_1, w_2\}$ has changed and we need to include the w-in-neighbors of w_2, which is w_5, in $w_{\mathcal{D}}$, leading to $w_{\mathcal{D}} = \{w_2, w_3, w_4, w_5\}$. After a check we can conclude that all node signals in $w_{\mathcal{D}}$ that have noise terms v_k, $k \in \mathcal{D}$ that are correlated with any $v_\ell, \ell \in \mathcal{Y}$ are included in $\mathcal{Y}$ too.*

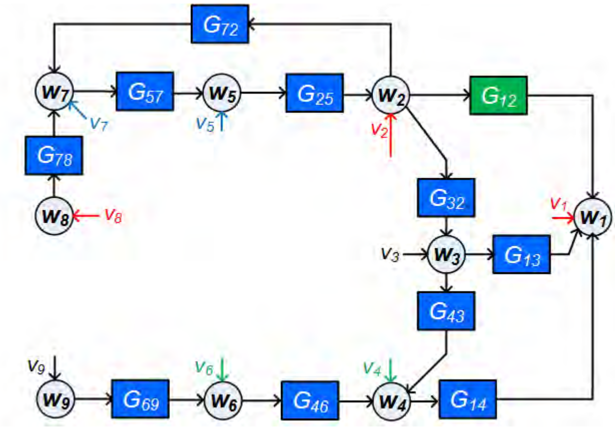


Figure 3.7: Example network with v_1 dynamically correlated with v_2 and v_8 (red colored). v_4 is dynamically correlated with v_6 (green colored) and v_5 is dynamically correlated with v_7 (blue colored).

The result now becomes

$$\mathcal{Y} = \{1,2\} \quad ; \quad \mathcal{D} = \{2,3,4,5\} \tag{3.9}$$

$$\mathcal{Q} = \mathcal{Y} \cap \mathcal{D} = \{2\} \quad ; \quad \mathcal{A} = \mathcal{D} \backslash \mathcal{Q} = \{3,4,5\}. \tag{3.10}$$

Since v_8 is dynamically correlated with v_1, in the resulting situation we will have a confounding variable for the estimation problem $w_5 \rightarrow w_1$ (i.e. $w_{\mathcal{A}} \rightarrow w_{\mathcal{Y}}$). As per condition 2a of Property 3.1, the path of the confounding variable e_8 to w_5 should be blocked by a node signal in $w_{\mathcal{B}}$, which can be either w_7 or w_8. w_7 cannot be chosen in $w_{\mathcal{B}}$ since this would create a confounding variable for $w_{\mathcal{A}} \rightarrow w_{\mathcal{B}}$ (i.e. $w_5 \rightarrow w_7$). Moreover, $w_7 \in w_{\mathcal{B}}$ would also create an unmeasured path $w_i \rightarrow w_7$ with $w_i = w_2$, thereby violating Condition 2c of Property 3.1. When w_8 is chosen in $w_{\mathcal{B}}$, the conditions in Property 3.1 are satisfied and hence we choose $\mathcal{B} = \{8\}$. The resulting estimation problem is $(w_2, w_3, w_4, w_5, w_8) \rightarrow (w_1, w_2)$, and will according to Theorem 3.2 provide a consistent and maximum likehood estimate of G_{12}.

3.8 Algorithm for signal selection: minimum input case

Rather than measuring all node signals that are w-in-neighbors of the output w_j of our target module G_{ji}, we now focus on an identification setup that uses a minimum number of measured node signals, according to the following strategy:

- We start by selecting $i \in \mathcal{D}$ and $j \in \mathcal{Y}$;
- Then we extend $\mathcal{D}$ with a minimum number of node signals that satisfies the parallel path and loop Condition 3.1.
- Every node signal w_k in $w_\mathcal{D}$ for which there is a *direct* or *indirect* confounding variable for the estimation problem $w_k \rightarrow w_\mathcal{Y}$ is included in $\mathcal{Y}$ and $\mathcal{Q}$.
- With $\mathcal{A} := \mathcal{D} \backslash \mathcal{Q}$ and $\mathcal{B} = \emptyset$ it follows that by construction there are no confounding variables for the estimation problem $w_\mathcal{A} \rightarrow w_\mathcal{Y}$.
- Finally, we define the identification setup as the estimation problem $w_\mathcal{D} \rightarrow w_\mathcal{Y}$, with $\mathcal{D} = \mathcal{Q} \cup \mathcal{A}$.

As we can observe, the algorithm does not require selection of set $\mathcal{B}$. This is attributed to the way we handle the indirect confounding variables for the estimation problem $w_\mathcal{A} \rightarrow w_\mathcal{Y}$. Instead of tackling these confounding variables by adding blocking node signals $w_\mathcal{B}$ (as in full input case) to be added as predictor inputs, we deal with them by moving the concerned $w_k, k \in \mathcal{A}$ to $w_\mathcal{Q}$ and thus to the set of predicted outputs. We choose this approach in order to minimize the required number of measured node signals. In this way, by construction, there will be no *direct* or *indirect* confounding variables for the estimation problem $w_\mathcal{A} \rightarrow w_\mathcal{Y}$. From this result, we can guarantee that the conditions in Theorem 3.1 will be satisfied since $\mathcal{B} = \emptyset$. Thus at the end of the algorithm we obtain a set of signals that provides target module invariance.

Example 3.5 *Consider the same network as in Example 3.4 represented by Figure 3.7. Applying the algorithm of this section, we first include the input of the target module* w_2 *in* $w_\mathcal{D}$ *and the output of the target module* w_1 *in* $w_\mathcal{Y}$*. There exist two parallel paths from* w_2 *to* w_1*, namely* $w_2 \rightarrow w_3 \rightarrow w_1$ *and* $w_2 \rightarrow w_3 \rightarrow w_4 \rightarrow w_1$ *and no loops through* w_1*. In order to satisfy Condition 3.1 we can include either* w_3 *in* $\mathcal{D}$ *such that* $\mathcal{D} = \{2, 3\}$ *or both* w_3, w_4 *in* $\mathcal{D}$ *such that* $\mathcal{D} = \{2, 3, 4\}$*. We choose the former to have minimum number of node signals. Because of the correlation between* v_2 *and* v_1 *there is a confounding variable for the estimation problem* $w_2 \rightarrow w_1$*. According to step 3 of the algorithm,* w_2 *is then moved to* $\mathcal{Y}$ *and* $\mathcal{Q}$*, leading to* $w_\mathcal{Y} = \{w_1, w_2\}$*. Because of this change of* $\mathcal{Y}$ *we have to recheck for presence of confounding variables. However this change does not introduce any additional confounding variables. The resulting estimation problem is* $(w_2, w_3) \rightarrow (w_1, w_2)$ *with* $w_\mathcal{A} = w_3$*,* $w_\mathcal{B} = \emptyset$*,* $w_\mathcal{Q} = w_2$ *and* $w_\mathcal{Y} = (w_1, w_2)$*.* □

In comparison with the full input case, the algorithm in this section will typically have a higher number of predicted output nodes and a smaller number of predictor inputs. This implies that there is a stronger emphasis on estimating a (multivariate) noise model $\bar{H}$. Given the choice of the direct identification method, and the choice of signals to satisfy the parallel path and loop condition, this algorithm indeed adds the smallest number of additional signals to be measured, as the removal of any of the additional signals will lead to conflicts with the required conditions.

3.9 Algorithm for signal selection: User selection case

Next we focus on the situation that we have a prior given set of nodes that we have access to i.e. a set of nodes that can (possibly) be measured. We refer to these nodes as *accessible nodes* while the remaining nodes are called *inaccessible*. This strategy is different from the *full input case* since we do not assume that we have access to all in-neighbours of $w_{\mathcal{Y}}$. This will lead to an identification setup with use of accessible node signals that contain information which is relevant for modeling our target module G_{ji}. We consider the situation that nodes w_i and w_j are accessible nodes and there are accessible nodes that satisfy the parallel path and loop Condition 3.1.

The following strategy will be followed:

1. We start by selecting $i \in \mathcal{D}$ and $j \in \mathcal{Y}$;
2. Then we extend $\mathcal{D}$ to satisfy the parallel path and loop Condition 3.1;
3. We include in $\mathcal{D}$ all accessible w-in-neighbors of $\mathcal{Y}$;
4. We extend $\mathcal{D}$ in such a way that for every non-accessible w-in-neighbor w_k of $w_{\mathcal{Y}}$ we include all accessible nodes that have path to w_k that runs through non-accessible nodes only.
5. If there is a direct confounding variable for $w_i \rightarrow w_{\mathcal{Y}}$, or an indirect one that has a path to w_i that does not pass through any accessible nodes, then i is included in $\mathcal{Y}$ and $\mathcal{Q}$;
6. A node signal w_k, $k \in \mathcal{D}$ is included in $\mathcal{A}$ if there are either no confounding variables for $w_k \rightarrow w_{\mathcal{Y}}$ or only indirect confounding variables that have paths to w_k that pass through accessible nodes.
7. Every node signal $w_k, k \in \mathcal{D} \backslash \{i\}$ that has a *direct* confounding variable for $w_k \rightarrow w_{\mathcal{Y}}$, or an *indirect* confounding variable with a path to w_k that does not pass through any accessible nodes is:
   - included in $\mathcal{B}$ if condition 2a and 2c of Theorem 3.1 are satisfied on including it in $w_{\mathcal{B}}$ (else)
   - included in $\mathcal{Y}$ and $\mathcal{Q}$; return to step 3.
8. Every node signal w_k, $k \in \mathcal{A}$ for which there are only indirect confounding variables as meant in Step 6, is
   - moved to $\mathcal{B}$ if Conditions 2a and 2c of Theorem 3.1 are satisfied and $k \neq i$; (else)

- kept in $\mathcal{A}$ while a set of accessible nodes that blocks the path of the confounding variable is added to $\mathcal{B} \cup \mathcal{A}$, while satisfying Conditions 2a and 2c of Theorem 3.1; (else)
- included in $\mathcal{Y}$ and $\mathcal{Q}$;

In the algorithm above, the prime reasoning is to deal with confounding variables for $w_{\mathcal{A}} \rightarrow w_{\mathcal{Y}}$. Direct confounding variables lead to including the respective node in the outputs $\mathcal{Y}$ or shifting the respective input node to $\mathcal{B}$, while indirect confounding variables are treated by either shifting the input node to $\mathcal{B}$ or, if its effect can be blocked, by adding an accessible node to the inputs in $\mathcal{B}$, or, if the blocking conditions can not be satisfied, by including the node in the output $\mathcal{Y}$. Note that the algorithm always provides a solution if Condition 1 of Theorem 3.1 (parallel path and loop condition) can be satisfied.

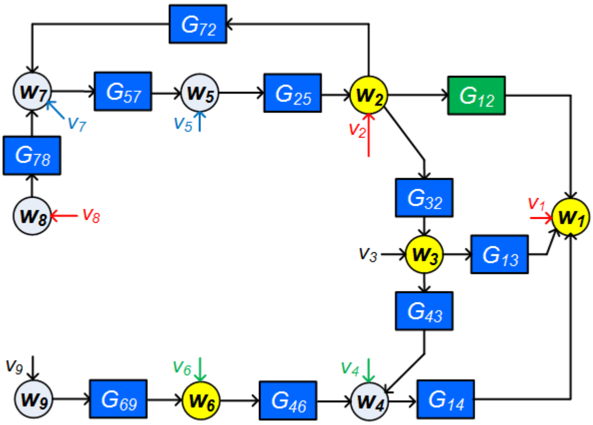


Figure 3.8: Example network of Figure 3.7 with accessible nodes w_1, w_2, w_3, w_6 indicated in yellow.

Example 3.6 *Consider the same network as in example 3.4 represented by Figure 3.8. However, we are given that only the nodes w_1, w_2, w_3 and w_6 are accessible. We now select the signals according to the algorithm presented in this section. First we include $w_i = w_2$ in $w_{\mathcal{D}}$ and $w_j = w_1$ in $w_{\mathcal{Y}}$. Then we extend $\mathcal{D}$ such that the parallel path and loop Condition 3.1 is satisfied. This is done by selecting $\mathcal{D} = \{2, 3\}$. According to step 4, we extend $\mathcal{D}$ by node w_6 as it serves as nearest accessible in-neighbor of w_4, being an inaccessible in-neighbor of w_1. As per Step 5, since v_1 and v_2 are correlated, w_2 is moved to $\mathcal{Y}$ and $\mathcal{Q}$. As per Step 6, there are no confounding variables for the estimation problem $w_3 \rightarrow w_1$ and hence w_3 is included in $w_{\mathcal{A}}$. Since v_4 and v_6 are correlated, it implies that there is an indirect confounding variable for the estimation problem $w_6 \rightarrow w_1$, which however does not pass through* *an accessible node. Step 7 does not apply since $w_3 \in w_{\mathcal{A}}$ has no confounding variables. Step 8 requires to deal with the indirect confounding variable v_4 for $w_6 \to w_1$. Checking Conditions 2a and 2c of Theorem 3.1 for $\mathcal{A}$ and $\mathcal{B}$, it appears that every path from $w_i = w_2$ or from $w_j = w_1$ to w_6 passes through a measured node and there are no confounding variable for the estimation problem $w_{\mathcal{A}} \to w_6$. Hence we include w_6 in $w_{\mathcal{B}}$. As a result, the estimation problem is $(w_2, w_3, w_6) \to (w_1, w_2)$.*

Remark 3.2 *Rather than starting the signal selection problem from a fixed set of accessible notes, the provided theory allows for an iterative and interactive algorithm for selecting accessible nodes in sensor allocation problems in a flexible way.*

3.10 Discussion

All three presented algorithms lead to a set of selected node signals that satisfy the conditions for target module invariance, and thus provide a predictor model in which no confounding variables can deteriorate the estimation of the target module. Only in the "User selection case" this is conditioned on the fact that appropriate node signals should be available to satisfy the parallel path and loop condition. Under these circumstances the presented algorithms are sound and complete [74]. This attractive feasibility result is mainly attributed to the addition of predicted outputs, that adds flexibility to solve the problem of confounding variables.

Note that the presented algorithms do not guarantee the consistency of the estimated target module. For this to hold the additional conditions for consistency, among which data-informativity and the delay in path/loop condition, need to be satisfied too, as illustrated in Figure 3.9. A specification of path-based conditions for data-informativity is the next important step, and results on this problem are presented in Chapter 5 of this thesis. Including these path-based conditions in the signal selection algorithms would be a next natural step to take. This also holds for the development of data-driven techniques to estimate the correlation structure of the disturbances, which will be addressed in Chapter 6 of this thesis.

It can be observed that the three algorithms presented in the previous sections rely only on the graphical conditions of the network. This paves way to automate the signal selection procedure using graph based algorithms that are scalable to large dimensions, with input being topology of the network and disturbance correlation structure represented as adjacency matrices. Also, it can be observed that the three considered cases in the previous sections, most likely will lead to three different experimental setups for estimating the single target module. For all three cases we can arrive at consistent and maximum likehood estimates of the target module. However, because of the fact that the experimental setups are

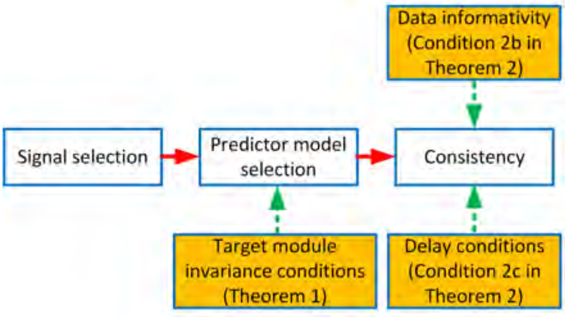

Data informativity (Condition 2b in Theorem 2)
Signal selection
Predictor model selection
Consistency
Target module invariance conditions (Theorem 1)
Delay conditions (Condition 2c in Theorem 2)

Figure 3.9: Figure to depict that consistency result requires satisfaction of conditions in Theorem 3.2 along with the appropriate predictor model.

different in the three cases, the data-informativity conditions and the statistical properties of the target module estimates will be different. The minimum variance expressions, in the form of the related Cramér-Rao lower bounds, will typically be different for the different experimental setups. Comparing these bounds for different experimental setups is beyond the scope of this chapter and considered as topic for future research.
We have formulated identification criteria in the realm of classical prediction error methods. This will typically lead to complex non-convex optimization problems that will scale poorly with the dimensions (number of parameters) of the problems. However alternative optimization approaches are becoming available that scale well and that rely on regularized kernel-based methods, thus exploiting new developments that originate from machine learning, see Chapter 8, and relaxations that rely on sequential convex optimization [51, 140], see Chapter 6.

3.11 Conclusions

A new local module identification approach has been presented to identify local modules in a dynamic network with given topology and process noise that is correlated over the different nodes. For this case, it is shown that the problem can be solved by moving from a MISO to a MIMO identification setup. In this setup the target module is embedded in a MIMO problem with appropriately chosen inputs and outputs, that warrant the consistent estimation of the target module with maximum likelihood properties. The key part of the procedure is the handling of direct and indirect confounding variables that are induced by correlated disturbances and/or non-measured node signals, and thus essentially dependent on the (Boolean) topology of the network and the (Boolean) correlation structure of the disturbances. A general theory has been developed that allows for specification of different types of algorithms, of which the "full input case", the "minimum input case" and the "user selection case" have been illustrated through examples. The presented theory is suitable for generalization to the estimation of sets of target modules. This work on handling confounding variables finds its importance not only in the engineering domain but also in fields like neuroscience, medicine, economics and marketing, where it serves as an important topic.

Appendices

3.A Proof of Proposition 3.1

Starting with the network representation (3.2), we can eliminate the non-measured node variables $w_{\mathcal{Z}}$ from the equations, by writing the last (block) row of (3.2) into an explicit expression for $w_{\mathcal{Z}}$:

$$w_{\mathcal{Z}} = (I - G_{\mathcal{Z}\mathcal{Z}})^{-1} \left[\sum_{k \in \mathcal{Q} \cup \{o\} \cup \mathcal{U}} G_{\mathcal{Z}k} w_k + \sum_{\ell \in \mathcal{Q} \cup \{o\} \cup \mathcal{U} \cup \mathcal{Z}} H_{\mathcal{Z}\ell} w_\ell \right],$$

and by substituting this $w_{\mathcal{Z}}$ into the expressions for the remaining w-variables. As a result

$$\begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \end{bmatrix} = \begin{bmatrix} \breve{G}_{\mathcal{Q}\mathcal{Q}} & \breve{G}_{\mathcal{Q}o} & \breve{G}_{\mathcal{Q}\mathcal{U}} \\ \breve{G}_{o\mathcal{Q}} & \breve{G}_{oo} & \breve{G}_{o\mathcal{U}} \\ \breve{G}_{\mathcal{U}\mathcal{Q}} & \breve{G}_{\mathcal{U}o} & \breve{G}_{\mathcal{U}\mathcal{U}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \end{bmatrix} + \breve{v},$$

$$\breve{v} = \breve{H} \begin{bmatrix} e_{\mathcal{Q}} \\ e_o \\ e_{\mathcal{U}} \\ e_{\mathcal{Z}} \end{bmatrix} = \begin{bmatrix} \breve{H}_{\mathcal{Q}\mathcal{Q}} & \breve{H}_{\mathcal{Q}o} & \breve{H}_{\mathcal{Q}\mathcal{U}} & \breve{H}_{\mathcal{Q}\mathcal{Z}} \\ \breve{H}_{o\mathcal{Q}} & \breve{H}_{oo} & \breve{H}_{o\mathcal{U}} & \breve{H}_{o\mathcal{Z}} \\ \breve{H}_{\mathcal{U}\mathcal{Q}} & \breve{H}_{\mathcal{U}o} & \breve{H}_{\mathcal{U}\mathcal{U}} & \breve{H}_{\mathcal{U}\mathcal{Z}} \end{bmatrix} \begin{bmatrix} e_{\mathcal{Q}} \\ e_o \\ e_{\mathcal{U}} \\ e_{\mathcal{Z}} \end{bmatrix} \tag{3.11}$$

with $cov(e) = I$, and where

$$\breve{G}_{kh} = G_{kh} + G_{k\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} G_{\mathcal{Z}h} \tag{3.12}$$

with $k, h \in \{\mathcal{Q} \cup \{o\} \cup \mathcal{U}\}$, and

$$\breve{H}_{k\ell} = H_{k\ell} + G_{k\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} H_{\mathcal{Z}\ell}, \tag{3.13}$$

with $\ell \in \{\mathcal{Q} \cup \{o\} \cup \mathcal{U} \cup \mathcal{Z}\}$.

On the basis of (3.11), the spectral density of $\breve{v}$ is given by $\Phi_{\breve{v}} = \breve{H}\breve{H}^*$. Applying a spectral factorization [150] to $\Phi_{\breve{v}}$ will deliver $\Phi_{\breve{v}} = \tilde{H}\tilde{\Lambda}\tilde{H}^*$ with $\tilde{H}$ a monic, stable and minimum phase rational matrix, and $\tilde{\Lambda}$ a positive definite (constant) matrix. Then there exists a white noise process $\tilde{\xi}$ defined by $\tilde{\xi} := \tilde{H}^{-1}\breve{H}e$ such that $\tilde{H}\tilde{\xi} = \breve{v}$, with $\mathrm{cov}(\tilde{\xi}) = \tilde{\Lambda}$, while $\tilde{H}$ is of the form

$$\tilde{H} = \begin{bmatrix} \tilde{H}_{11} & \tilde{H}_{12} & \tilde{H}_{13} \\ \tilde{H}_{21} & \tilde{H}_{22} & \tilde{H}_{23} \\ \tilde{H}_{31} & \tilde{H}_{32} & \tilde{H}_{33} \end{bmatrix} \tag{3.14}$$

and where the block dimensions are conformable to the dimensions of $w_{\mathcal{Q}}$, w_o and $w_{\mathcal{U}}$ respectively. As a result, (3.11) can be rewritten as

$$\begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \end{bmatrix} = \begin{bmatrix} \breve{G}_{\mathcal{Q}\mathcal{Q}} & \breve{G}_{\mathcal{Q}o} & \breve{G}_{\mathcal{Q}\mathcal{U}} \\ \breve{G}_{o\mathcal{Q}} & \breve{G}_{oo} & \breve{G}_{o\mathcal{U}} \\ \breve{G}_{\mathcal{U}\mathcal{Q}} & \breve{G}_{\mathcal{U}o} & \breve{G}_{\mathcal{U}\mathcal{U}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \end{bmatrix} + \tilde{H} \begin{bmatrix} \tilde{\xi}_{\mathcal{Q}} \\ \tilde{\xi}_o \\ \tilde{\xi}_{\mathcal{U}} \end{bmatrix}. \tag{3.15}$$

By denoting

$$\begin{bmatrix} \check{H}_{13} \\ \check{H}_{23} \end{bmatrix} := \begin{bmatrix} \tilde{H}_{13}\tilde{H}_{33}^{-1} \\ \tilde{H}_{23}\tilde{H}_{33}^{-1} \end{bmatrix} \tag{3.16}$$

and premultiplying (3.15) with

$$\begin{bmatrix} I & 0 & -\check{H}_{13} \\ 0 & I & -\check{H}_{23} \\ 0 & 0 & I \end{bmatrix} \tag{3.17}$$

while only keeping the identity terms on the left hand side, we obtain an equivalent network equation:

$$\begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \end{bmatrix} = \begin{bmatrix} \breve{G}'_{\mathcal{Q}\mathcal{Q}} & \breve{G}'_{\mathcal{Q}o} & \breve{G}'_{\mathcal{Q}\mathcal{U}} \\ \breve{G}'_{o\mathcal{Q}} & \breve{G}'_{oo} & \breve{G}'_{o\mathcal{U}} \\ \breve{G}_{\mathcal{U}\mathcal{Q}} & \breve{G}_{\mathcal{U}o} & \breve{G}_{\mathcal{U}\mathcal{U}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \end{bmatrix} + \begin{bmatrix} \tilde{H}'_{11} & \tilde{H}'_{12} & 0 \\ \tilde{H}'_{21} & \tilde{H}'_{22} & 0 \\ \tilde{H}_{31} & \tilde{H}_{32} & \tilde{H}_{33} \end{bmatrix} \begin{bmatrix} \tilde{\xi}_{\mathcal{Q}} \\ \tilde{\xi}_o \\ \tilde{\xi}_{\mathcal{U}} \end{bmatrix}, \tag{3.18}$$

with

$$\breve{G}'_{\mathcal{Q}\mathcal{U}} = \breve{G}_{\mathcal{Q}\mathcal{U}} - \check{H}_{13}\breve{G}_{\mathcal{U}\mathcal{U}} + \check{H}_{13} \tag{3.19}$$

$$\breve{G}'_{\mathcal{Q}\star} = \breve{G}_{\mathcal{Q}\star} - \check{H}_{13}\breve{G}_{\mathcal{U}\star} \tag{3.20}$$

$$\breve{G}'_{o\star} = \breve{G}_{o\star} - \check{H}_{23}\breve{G}_{\mathcal{U}\star} \tag{3.21}$$

$$\breve{G}'_{o\mathcal{U}} = \breve{G}_{o\mathcal{U}} - \check{H}_{23}\breve{G}_{\mathcal{U}\mathcal{U}} + \check{H}_{23} \tag{3.22}$$

$$\tilde{H}'_{1\square} = \tilde{H}_{1\square} - \check{H}_{13}\tilde{H}_{3\square} \tag{3.23}$$

$$\tilde{H}'_{2\square} = \tilde{H}_{2\square} - \check{H}_{23}\tilde{H}_{3\square}. \tag{3.24}$$

where $\star \in \{\mathcal{Q} \cup \{o\}\}$ and $\square \in \{1, 2\}$.

The next step is now to show that that the block elements $\breve{G}'_{\mathcal{Q}o}$ and $\breve{G}'_{oo}$ in G can be made 0. This can be done by variable substitution as follows:

The second row in (3.18) is replaced by an explicit expression for w_o according to

$$w_o = (1 - \breve{G}'_{oo})^{-1}[\breve{G}'_{o\mathcal{Q}}w_{\mathcal{Q}} + \breve{G}'_{o\mathcal{U}}w_{\mathcal{U}} + \tilde{H}'_{21}\tilde{\xi}_{\mathcal{Q}} + \tilde{H}'_{22}\tilde{\xi}_o].$$

Additionally, this expression for w_o is substituted into the first block row of (3.18), to remove the w_o-dependent term on the right hand side, leading to

$$\begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \end{bmatrix} = \begin{bmatrix} \breve{G}''_{\mathcal{Q}\mathcal{Q}} & 0 & \breve{G}''_{\mathcal{Q}\mathcal{U}} \\ \bar{G}_{o\mathcal{Q}} & 0 & \bar{G}_{o\mathcal{U}} \\ \breve{G}_{\mathcal{U}\mathcal{Q}} & \breve{G}_{\mathcal{U}o} & \breve{G}_{\mathcal{U}\mathcal{U}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \end{bmatrix} + \begin{bmatrix} \tilde{H}''_{11} & \tilde{H}''_{12} & 0 \\ \tilde{H}''_{21} & \tilde{H}''_{22} & 0 \\ \tilde{H}_{31} & \tilde{H}_{32} & \tilde{H}_{33} \end{bmatrix} \begin{bmatrix} \tilde{\xi}_{\mathcal{Q}} \\ \tilde{\xi}_o \\ \tilde{\xi}_{\mathcal{U}} \end{bmatrix} \tag{3.25}$$

with

$$\bar{G}_{o\star} = (I - \breve{G}'_{oo})^{-1} \breve{G}'_{o\star} \tag{3.26}$$
$$\tilde{H}''_{2\star} = (I - \breve{G}'_{oo})^{-1} \tilde{H}'_{2\star} \tag{3.27}$$
$$\breve{G}''_{\mathcal{Q}\star} = \breve{G}'_{\mathcal{Q}\star} + \breve{G}'_{\mathcal{Q}o} \bar{G}_{o\star} \tag{3.28}$$
$$\tilde{H}''_{1\star} = \tilde{H}'_{1\star} + \breve{G}'_{\mathcal{Q}o} \tilde{H}''_{2\star}. \tag{3.29}$$

Since because of these operations, the matrix $\breve{G}''_{\mathcal{Q}\mathcal{Q}}$ might not be hollow, we move any diagonal terms of this matrix to the left hand side of the equation, and premultiply the first (block) equation by the diagonal matrix $(I - \mathrm{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1}$, to obtain the expression

$$\begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \end{bmatrix} = \begin{bmatrix} \bar{G}_{\mathcal{Q}\mathcal{Q}} & 0 & \bar{G}_{\mathcal{Q}\mathcal{U}} \\ \bar{G}_{o\mathcal{Q}} & 0 & \bar{G}_{o\mathcal{U}} \\ \breve{G}_{\mathcal{U}\mathcal{Q}} & \breve{G}_{\mathcal{U}o} & \breve{G}_{\mathcal{U}\mathcal{U}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \end{bmatrix} + \begin{bmatrix} \tilde{H}'''_{11} & \tilde{H}'''_{12} & 0 \\ \tilde{H}''_{21} & \tilde{H}''_{22} & 0 \\ \tilde{H}_{31} & \tilde{H}_{32} & \tilde{H}_{33} \end{bmatrix} \begin{bmatrix} \tilde{\xi}_{\mathcal{Q}} \\ \tilde{\xi}_o \\ \tilde{\xi}_{\mathcal{U}} \end{bmatrix} \tag{3.30}$$

with

$$\bar{G}_{\mathcal{Q}\mathcal{Q}} = (I - \mathrm{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1} (\breve{G}''_{\mathcal{Q}\mathcal{Q}} - \mathrm{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}})), \tag{3.31}$$
$$\bar{G}_{\mathcal{Q}\mathcal{U}} = (I - \mathrm{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1} \breve{G}''_{\mathcal{Q}\mathcal{U}} \tag{3.32}$$
$$\tilde{H}'''_{1\star} = (I - \mathrm{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1} \tilde{H}''_{1\star}. \tag{3.33}$$

As final step, we need the matrix $\tilde{H}_r := \begin{bmatrix} \tilde{H}'''_{11} & \tilde{H}'''_{12} \\ \tilde{H}''_{21} & \tilde{H}''_{22} \end{bmatrix}$ to be monic, stable and minimum phase to obtain the representation as in (3.3). To that end, we consider the stochastic process $\tilde{v}_{\mathcal{Y}} := \tilde{H}_r \tilde{\xi}_{\mathcal{Y}}$ with $\tilde{\xi}_{\mathcal{Y}} := \begin{bmatrix} \tilde{\xi}_{\mathcal{Q}}^\top & \tilde{\xi}_o^\top \end{bmatrix}^\top$. The spectral density of $\tilde{v}_{\mathcal{Y}}$ is then given by $\Phi_{\tilde{v}_{\mathcal{Y}}} = \tilde{H}_r \tilde{\Lambda}_{\mathcal{Y}} \tilde{H}_r^\star$ with $\tilde{\Lambda}_{\mathcal{Y}}$ the covariance matrix of $\tilde{\xi}_{\mathcal{Y}}$, that can be decomposed as $\tilde{\Lambda}_{\mathcal{Y}} = \tilde{\Gamma}_r \tilde{\Gamma}_r^T$. From spectral factorization [150] it follows that the spectral factor $\tilde{H}_r \tilde{\Gamma}_r$ of $\Phi_{\tilde{v}_{\mathcal{Y}}}$ satisfies

$$\tilde{H}_r \tilde{\Gamma}_r = \bar{H}_s D \tag{3.34}$$

with $\bar{H}_s$ a stable and minimum phase rational matrix, and D an "all pass" stable rational matrix satisfying $DD^\star = I$.
The signal $\tilde{v}_{\mathcal{Y}}$ can then be written as

$$\tilde{v}_{\mathcal{Y}} = \tilde{H}_r \tilde{\xi}_{\mathcal{Y}} = \bar{H}_s D \tilde{\Gamma}_r^{-1} \tilde{\xi}_{\mathcal{Y}}.$$

By defining $\bar{H}_s^{\infty} := \lim_{z\to\infty} \bar{H}_s$, this can be rewritten as

$$\tilde{v}_{\mathcal{Y}} = \tilde{H}_r\tilde{\xi}_{\mathcal{Y}} = \underbrace{\bar{H}_s(\bar{H}_s^{\infty})^{-1}}_{\bar{H}}\,\underbrace{\bar{H}_s^{\infty}D\tilde{\Gamma}_r^{-1}\tilde{\xi}_{\mathcal{Y}}}_{\xi_{\mathcal{Y}}}.$$

As a result, $\bar{H}$ is a monic stable and stably invertible rational matrix, and $\xi_{\mathcal{Y}}$ is a white noise process with spectral density given by $\bar{H}_s^{\infty}D\tilde{\Gamma}_r^{-1}\Phi_{\tilde{\xi}_{\mathcal{Y}}}\tilde{\Gamma}_r^{-T}D^{\star}(\bar{H}_s^{\infty})^T = \bar{H}_s^{\infty}(\bar{H}_s^{\infty})^T$. Therefore we can write (3.30) as,

$$\begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \end{bmatrix} = \begin{bmatrix} \bar{G}_{\mathcal{Q}\mathcal{Q}} & 0 & \bar{G}_{\mathcal{Q}\mathcal{U}} \\ \bar{G}_{o\mathcal{Q}} & 0 & \bar{G}_{\mathcal{A}\mathcal{U}} \\ \breve{G}_{\mathcal{U}\mathcal{Q}} & \breve{G}_{\mathcal{U}o} & \breve{G}_{\mathcal{U}\mathcal{U}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \end{bmatrix} + \begin{bmatrix} \bar{H}_{11} & \bar{H}_{12} & 0 \\ \bar{H}_{21} & \bar{H}_{22} & 0 \\ \bar{H}_{31} & \bar{H}_{32} & \tilde{H}_{33} \end{bmatrix} \begin{bmatrix} \xi_{\mathcal{Q}} \\ \xi_o \\ \tilde{\xi}_{\mathcal{U}} \end{bmatrix} \tag{3.35}$$

where $\begin{bmatrix}\bar{H}_{31} & \bar{H}_{32}\end{bmatrix} = \begin{bmatrix}\tilde{H}_{31} & \tilde{H}_{32}\end{bmatrix}\tilde{\Gamma}_r D^{-1}(\bar{H}_s^{\infty})^{-1}$. Let $[\bar{H}'_{31} \quad \bar{H}'_{32}] = [\bar{H}_{31} \quad \bar{H}_{32}]\begin{bmatrix}\bar{H}_{11} & \bar{H}_{12} \\ \bar{H}_{21} & \bar{H}_{22}\end{bmatrix}^{-1}$. Pre-multiplying (4.36) with $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -\bar{H}'_{31} & -\bar{H}'_{32} & I \end{bmatrix}$ while only keeping the identity terms on the left hand side, we obtain an equivalent network equation:

$$\begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \end{bmatrix} = \begin{bmatrix} \bar{G}_{\mathcal{Q}\mathcal{Q}} & 0 & \bar{G}_{\mathcal{Q}\mathcal{U}} \\ \bar{G}_{o\mathcal{Q}} & 0 & \bar{G}_{\mathcal{A}\mathcal{U}} \\ \breve{G}'_{\mathcal{U}\mathcal{Q}} & \breve{G}'_{\mathcal{U}o} & \breve{G}'_{\mathcal{U}\mathcal{U}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{Q}} \\ w_o \\ w_{\mathcal{U}} \end{bmatrix} + \begin{bmatrix} \bar{H}_{11} & \bar{H}_{12} & 0 \\ \bar{H}_{21} & \bar{H}_{22} & 0 \\ 0 & 0 & \tilde{H}_{33} \end{bmatrix} \begin{bmatrix} \xi_{\mathcal{Q}} \\ \xi_o \\ \tilde{\xi}_{\mathcal{U}} \end{bmatrix} \tag{3.36}$$

where $\breve{G}'_{\mathcal{U}\mathcal{Q}} = \breve{G}_{\mathcal{U}\mathcal{Q}} - \bar{H}'_{31}\breve{G}''''_{\mathcal{Q}\mathcal{Q}} - \breve{H}'_{32}\bar{G}''_{o\mathcal{Q}} + \bar{H}'_{31}$, $\breve{G}'_{\mathcal{U}o} = \breve{G}_{\mathcal{U}o} + \bar{H}'_{32}$ and $\breve{G}'_{\mathcal{U}\mathcal{U}} = \breve{G}_{\mathcal{U}\mathcal{U}} - \bar{H}'_{31}\breve{G}''''_{\mathcal{Q}\mathcal{U}} - \bar{H}'_{32}\breve{G}''_{\mathcal{A}\mathcal{U}}$. In order to make $\breve{G}'_{\mathcal{U}\mathcal{U}}$ hollow, we move any diagonal terms of this matrix to the left hand side of the equation, and pre-multiply the third (block) equation by the diagonal matrix $(I - \mathrm{diag}(\breve{G}'_{\mathcal{U}\mathcal{U}}))^{-1}$. This will modify (3,3) (block) element of the H matrix to $(I - \mathrm{diag}(\breve{G}'_{\mathcal{U}\mathcal{U}}))^{-1}\tilde{H}_{33}$, which we need to be monic, stable and stably invertible. Applying spectral factorization as before [150], we can write the term $(I - \mathrm{diag}(\breve{G}'_{\mathcal{U}\mathcal{U}}))^{-1}\tilde{H}_{33}\tilde{\xi}_{\mathcal{U}}$ as $\bar{H}_{33}\xi_{\mathcal{U}}$ where $\bar{H}_{33}$ is monic, stable and stably invertible and $\xi_{\mathcal{U}}$ is a white noise process with covariance Λ_{33}. This completes the proof for obtaining (3.3).

The absence of confounding variables for the estimation problem $w_{\mathcal{U}} \to w_{\mathcal{Y}}$ can be proved as follows. Since all non-measured nodes $w_{\mathcal{Z}}$ are removed in the network represented by (3.3), the only non-measured signals in the network are the noise signals in ξ_m and they do not have any unmeasured paths to any nodes in the network (i.e. to w_m). Due to the block-diagonal structure of $\bar{H}_m$ in (3.3), the only non-measured signals that have direct paths to $w_{\mathcal{U}}$ originate from $\xi_{\mathcal{U}}$, while the only non-measured signals that have direct paths to $w_{\mathcal{Y}}$ originate from $[\xi_{\mathcal{Q}}^T \ \xi_o]^T$. Therefore there does not exist an element of ξ_m that has simultaneous unmeasured paths or direct paths to both $w_{\mathcal{U}}$ and $w_{\mathcal{Y}}$. □

3.B Proof of Theorem 3.1

In order to prove Theorem 3.1, we first present three preparatory Lemmas.

Lemma 3.1 *Consider a dynamic network as defined in (3.2), a vector $e_{\mathcal{X}}$ of white noise sources with $\mathcal{X} \subseteq \mathcal{L}$, and two subsets of nodes w_Φ and w_Ω, $\Phi, \Omega \subset \mathcal{L}\backslash\mathcal{Z}$. If in $e_{\mathcal{X}}$ there is no confounding variable for the estimation problem $w_\Phi \to w_\Omega$, then*

$$\breve{H}_{\Omega\mathcal{X}}\breve{H}^*_{\Phi\mathcal{X}} = \breve{H}_{\Phi\mathcal{X}}\breve{H}^*_{\Omega\mathcal{X}} = 0,$$

where $\breve{H}_{\Omega\mathcal{X}}$, $\breve{H}_{\Phi\mathcal{X}}$ are the noise model transfer functions in the immersed network (3.11) related to the appropriate variables.

Proof: If in $e_{\mathcal{X}}$ there is no confounding variable for the formulated estimation problem, then for all e_x, $x \in \mathcal{X}$ there do not exist simultaneous paths from e_x to w_Φ and w_Ω, that are direct or pass through nodes in $\mathcal{Z}$ only.
For the network where signals $w_{\mathcal{Z}}$ are immersed, it follows from (3.13), that $\breve{H}_{k\ell} = H_{k\ell} + G_{k\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}H_{\mathcal{Z}\ell}$ where $k \in \Phi$ and $\ell \in \mathcal{X}$. The first term in the sum (i.e. H_{kl}) is the noise model transfer in the direct path from e_ℓ to w_k and the second part of the sum is the transfer function in the unmeasured paths (i.e. paths through $w_{\mathcal{Z}}$ only) from e_ℓ to w_k. If all paths from a node signal e_x to w_Φ pass through a node in $w_{\mathcal{L}\backslash\mathcal{Z}}$, then there are no direct or unmeasured paths from e_x to nodes in w_Φ. This implies that $\breve{H}_{kx} = \breve{H}^*_{kx} = 0$ for all $k \in \Phi$ (i.e $\breve{H}_{\Phi x} = 0$). A dual reasoning applies to paths from e_x to w_Ω. Consider $e_{\mathcal{X}} = [e_{x_1} \quad e_{x_2} \quad \dots \quad e_{x_n}]^\top$. Then we have $\breve{H}_{\Phi\mathcal{X}}\breve{H}^*_{\Omega\mathcal{X}} = \breve{H}_{\Phi x_1}\breve{H}^*_{\Omega x_1} + \dots + \breve{H}_{\Phi x_n}\breve{H}^*_{\Omega x_n}$. If the condition in the lemma is satisfied, implying that there do not exist simultaneous paths, then in each of the product terms we either have $\breve{H}_{\Phi x_k} = 0$ or $\breve{H}^*_{\Omega x_k} = 0$ where $k = \{1, 2, \dots, n\}$. This proves the result of lemma 3.1. □

Lemma 3.2 *Consider a dynamic network as defined in (3.11) with target module G_{ji}, where the non-measured node signals w_z are immersed, while the node sets $\{o, \mathcal{Q}, \mathcal{U}\}$ are chosen according to the specifications in Section 3.4.*
Then $\bar{G}_{ji}$ is given by the following expressions:

$$\text{If } i \in \mathcal{Q}: \ \bar{G}_{ji} = (I - \breve{G}_{jj} + \breve{H}_{j3}\breve{G}_{\mathcal{U}j})^{-1}(\breve{G}_{ji} - \breve{H}_{j3}\breve{G}_{\mathcal{U}i}) \tag{3.37}$$

$$\text{If } i \in \mathcal{U}: \bar{G}_{ji} = (I - \breve{G}_{jj} + \breve{H}_{j3}\breve{G}_{\mathcal{U}j})^{-1}(\breve{G}_{ji} - \breve{H}_{j3}\breve{G}_{\mathcal{U}i} + \breve{H}_{ji}) \tag{3.38}$$

where $\breve{H}_{j3}$ is the row vector corresponding to the row of node signal j in $\breve{H}_{13}$ (if $j \in \mathcal{Q}$) or in $\breve{H}_{23}$ (if $j \in o$), and $\breve{H}_{ji}$ is the element corresponding to the column of node signal i in $\breve{H}_{j3}$.

Proof: For the target module G_{ji} we have the following cases that can occur:

1. $j = o$ and $i \in \mathcal{U}$. From (3.26) we have $\bar{G}_{ji} = (I - \breve{G}'_{jj})^{-1}\breve{G}'_{ji}$ where $\breve{G}'_{jj}$ is given by (3.21) and $\breve{G}'_{ji}$ is given by (3.22). This directly leads to (10.52).

2. $j = o$ and $i \in \mathcal{Q}$. From (3.26) we have $\bar{G}_{ji} = (I - \breve{G}'_{jj})^{-1}\breve{G}'_{ji}$ where $\breve{G}'_{jj}$ and $\breve{G}'_{ji}$ are given by (3.21), leading to (10.51).

3. $j \in \mathcal{Q}$, o is void and $i \in \mathcal{U}$. From (4.31) we have $\bar{G}_{ji} = (I - \breve{G}''_{jj})^{-1}\breve{G}''_{ji}$ where $\breve{G}''_{jj}$ and $\breve{G}''_{ji}$ are given by (4.25). Since o is void, (4.25) leads to $G''_{\mathcal{Q}\star} = \breve{G}'_{\mathcal{Q}\star}$. Therefore $\breve{G}''_{jj} = \breve{G}'_{jj}$ which is specified by (3.20), and $\breve{G}''_{ji} = \breve{G}'_{ji}$ which is given by (3.19). This leads to (10.52).

4. $j \in \mathcal{Q}$, o is void and $i \in \mathcal{Q}$. Since $j \neq i$ it follows from (4.30) that $\bar{G}_{ji} = (I - \breve{G}''_{jj})^{-1}\breve{G}''_{ji}$ where $\breve{G}''_{jj}$ and $\breve{G}''_{ji}$ are given by (4.25). Since o is void, (4.25) leads to $G''_{\mathcal{Q}\star} = \breve{G}'_{\mathcal{Q}\star}$. Therefore for this case, $\breve{G}''_{jj} = \breve{G}'_{jj}$ and $\breve{G}''_{ji} = \breve{G}'_{ji}$, which are given by (3.20). This leads to (10.51).

Lemma 3.3 *Consider a dynamic network as defined in (3.11) where the non-measured node signals $w_{\mathcal{Z}}$ are immersed, and let $\mathcal{U}$ be decomposed in sets $\mathcal{A}$ and $\mathcal{B}$ satisfying Condition 3.2. Then the spectral density $\Phi_{\breve{v}}$ has the unique spectral factorization $\Phi_{\breve{v}} = \tilde{H}\Lambda\tilde{H}^*$ with Λ constant and $\tilde{H}$ monic, stable, minimum phase, and of the form*

$$\Lambda = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} & \Lambda_{13} & 0 \\ \Lambda_{21} & \Lambda_{22} & \Lambda_{23} & 0 \\ \Lambda_{31} & \Lambda_{32} & \Lambda_{33} & 0 \\ 0 & 0 & 0 & \Lambda_{44} \end{bmatrix}, \quad \tilde{H} = \begin{bmatrix} \tilde{H}_{11} & \tilde{H}_{12} & \tilde{H}_{\mathcal{Q}\mathcal{B}} & 0 \\ \tilde{H}_{21} & \tilde{H}_{22} & \tilde{H}_{o\mathcal{B}} & 0 \\ \tilde{H}_{\mathcal{B}\mathcal{Q}} & \tilde{H}_{\mathcal{B}o} & \tilde{H}_{\mathcal{B}\mathcal{B}} & 0 \\ 0 & 0 & 0 & \tilde{H}_{\mathcal{A}\mathcal{A}} \end{bmatrix}, \tag{3.39}$$

where the block dimensions are conformable to the dimensions of $w_{\mathcal{Q}}$, w_o, $w_{\mathcal{B}}$ and $w_{\mathcal{A}}$ respectively.

Proof: On the basis of (3.11) we write $w_{\mathcal{U}} = [w_{\mathcal{B}}^\top \quad w_{\mathcal{A}}^\top]^\top$ and

$$\breve{v} = \breve{H}\begin{bmatrix} e_{\mathcal{Q}} \\ e_o \\ e_{\mathcal{B}} \\ e_{\mathcal{A}} \\ e_{\mathcal{Z}} \end{bmatrix} = \begin{bmatrix} \breve{H}_{\mathcal{Q}\mathcal{Q}} & \breve{H}_{\mathcal{Q}o} & \breve{H}_{\mathcal{Q}\mathcal{B}} & \breve{H}_{\mathcal{Q}\mathcal{A}} & \breve{H}_{\mathcal{Q}\mathcal{Z}} \\ \breve{H}_{o\mathcal{Q}} & \breve{H}_{oo} & \breve{H}_{o\mathcal{B}} & \breve{H}_{o\mathcal{A}} & \breve{H}_{o\mathcal{Z}} \\ \breve{H}_{\mathcal{B}\mathcal{Q}} & \breve{H}_{\mathcal{B}o} & \breve{H}_{\mathcal{B}\mathcal{B}} & \breve{H}_{\mathcal{B}\mathcal{A}} & \breve{H}_{\mathcal{B}\mathcal{Z}} \\ \breve{H}_{\mathcal{A}\mathcal{Q}} & \breve{H}_{\mathcal{A}o} & \breve{H}_{\mathcal{A}\mathcal{B}} & \breve{H}_{\mathcal{A}\mathcal{A}} & \breve{H}_{\mathcal{A}\mathcal{Z}} \end{bmatrix}\begin{bmatrix} e_{\mathcal{Q}} \\ e_o \\ e_{\mathcal{B}} \\ e_{\mathcal{A}} \\ e_{\mathcal{Z}} \end{bmatrix} \tag{3.40}$$

with $cov(e) = I$ and the components of $\breve{H}$ as specified in (3.13). Starting from the expression (4.38), the spectral density $\Phi_{\breve{v}}$ can be written as $\breve{H}\breve{H}^*$ while it is denoted as

$$\Phi_{\breve{v}} = \begin{bmatrix} \Phi_{\breve{v}_{\mathcal{Q}}} & \Phi_{\breve{v}_{\mathcal{Q}}\breve{v}_o} & \Phi_{\breve{v}_{\mathcal{Q}}\breve{v}_{\mathcal{B}}} & \Phi_{\breve{v}_{\mathcal{Q}}\breve{v}_{\mathcal{A}}} \\ \Phi^*_{\breve{v}_{\mathcal{Q}}\breve{v}_o} & \Phi_{\breve{v}_o} & \Phi_{\breve{v}_o\breve{v}_{\mathcal{B}}} & \Phi_{\breve{v}_o\breve{v}_{\mathcal{A}}} \\ \Phi^*_{\breve{v}_{\mathcal{Q}}\breve{v}_{\mathcal{B}}} & \Phi^*_{\breve{v}_o\breve{v}_{\mathcal{B}}} & \Phi_{\breve{v}_{\mathcal{B}}} & \Phi_{\breve{v}_{\mathcal{B}}\breve{v}_{\mathcal{A}}} \\ \Phi^*_{\breve{v}_{\mathcal{Q}}\breve{v}_{\mathcal{A}}} & \Phi^*_{\breve{v}_o\breve{v}_{\mathcal{A}}} & \Phi^*_{\breve{v}_{\mathcal{B}}\breve{v}_{\mathcal{A}}} & \Phi_{\breve{v}_{\mathcal{A}}} \end{bmatrix}. \tag{3.41}$$

In this structure we are particularly going to analyse the elements

$$\begin{aligned}
\Phi_{\breve{v}_{\mathcal{Q}}\breve{v}_{\mathcal{A}}} &= \breve{H}_{\mathcal{Q}\mathcal{Q}}\breve{H}^*_{\mathcal{A}\mathcal{Q}} + \breve{H}_{\mathcal{Q}o}\breve{H}^*_{\mathcal{A}o} + \breve{H}_{\mathcal{Q}\mathcal{B}}\breve{H}^*_{\mathcal{A}\mathcal{B}} + \breve{H}_{\mathcal{Q}\mathcal{A}}\breve{H}^*_{\mathcal{A}\mathcal{A}} + \breve{H}_{\mathcal{Q}\mathcal{Z}}\breve{H}^*_{\mathcal{A}\mathcal{Z}} \\
\Phi_{\breve{v}_o\breve{v}_{\mathcal{A}}} &= \breve{H}_{o\mathcal{Q}}\breve{H}^*_{\mathcal{A}\mathcal{Q}} + \breve{H}_{oo}\breve{H}^*_{\mathcal{A}o} + \breve{H}_{o\mathcal{B}}\breve{H}^*_{\mathcal{A}\mathcal{B}} + \breve{H}_{o\mathcal{A}}\breve{H}^*_{\mathcal{A}\mathcal{A}} + \breve{H}_{o\mathcal{Z}}\breve{H}^*_{\mathcal{A}\mathcal{Z}} \qquad (3.42) \\
\Phi_{\breve{v}_{\mathcal{B}}\breve{v}_{\mathcal{A}}} &= \breve{H}_{\mathcal{B}\mathcal{Q}}\breve{H}^*_{\mathcal{A}\mathcal{Q}} + \breve{H}_{\mathcal{B}o}\breve{H}^*_{\mathcal{A}o} + \breve{H}_{\mathcal{B}\mathcal{B}}\breve{H}^*_{\mathcal{A}\mathcal{B}} + \breve{H}_{\mathcal{B}\mathcal{A}}\breve{H}^*_{\mathcal{A}\mathcal{A}} + \breve{H}_{\mathcal{B}\mathcal{Z}}\breve{H}^*_{\mathcal{A}\mathcal{Z}}
\end{aligned}$$

If $\mathcal{A}$ and $\mathcal{B}$ satisfy Condition 3.2, then none of the white noise terms e_x, $x \in \mathcal{L}$ will be a confounding variable for the estimation problems $w_{\mathcal{A}} \to w_{\mathcal{Q}}$, $w_{\mathcal{A}} \to w_o$ or $w_{\mathcal{A}} \to w_{\mathcal{B}}$. Then it follows from Lemma 3.1 that all of the terms in (3.42) are zero. As a result we can write the spectrum in equation (3.41) as,

$$\Phi_{\breve{v}} = \begin{bmatrix} \Phi_{\breve{v}_{\mathcal{Q}}} & \Phi_{\breve{v}_{\mathcal{Q}}\breve{v}_o} & \Phi_{\breve{v}_{\mathcal{Q}}\breve{v}_{\mathcal{B}}} & 0 \\ \Phi^*_{\breve{v}_{\mathcal{Q}}\breve{v}_o} & \Phi_{\breve{v}_o} & \Phi_{\breve{v}_o\breve{v}_{\mathcal{B}}} & 0 \\ \Phi^*_{\breve{v}_{\mathcal{Q}}\breve{v}_{\mathcal{B}}} & \Phi^*_{\breve{v}_o\breve{v}_{\mathcal{B}}} & \Phi_{\breve{v}_{\mathcal{B}}} & 0 \\ 0 & 0 & 0 & \Phi_{\breve{v}_{\mathcal{A}}} \end{bmatrix} \qquad (3.43)$$

Then the spectral density $\Phi_{\breve{v}}$ has the unique spectral factorization [150]

$$\Phi_{\breve{v}} = \begin{bmatrix} F_{11}\Lambda_1 F_{11}^* & 0 \\ 0 & F_{22}\Lambda_2 F_{22}^* \end{bmatrix} = \tilde{H}\Lambda\tilde{H}^* \qquad (3.44)$$

where $\tilde{H}$ is of the form in (4.37), and monic, stable and minimum phase. □

Next we proceed with the proof of Theorem 3.1.

With Lemma 10.4 it follows that $\bar{G}_{ji}$ is given by either (10.51) or (10.52). For analysing these two expressions, we first are going to specify $\breve{G}_{ji}$ and $\breve{G}_{jj}$. From (3.12), we have

$$\begin{aligned}
\breve{G}_{ji} &= G_{ji} + G_{j\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}G_{\mathcal{Z}i} \qquad (3.45) \\
\breve{G}_{jj} &= G_{jj} + G_{j\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}G_{\mathcal{Z}j}, \qquad (3.46)
\end{aligned}$$

where the first terms on the right hand sides reflect the direct connections from w_i to w_j (respectively from w_j to w_j) and the second terms reflect the connections that pass only through nodes in $\mathcal{Z}$. By definition, $G_{jj} = 0$ since the G matrix in the network in (3.2) is hollow. Under the parallel path and loop condition 3.1, the second terms on the right hand sides of (4.65), (4.66) are zero, so that $\breve{G}_{ji} = G_{ji}$ and $\breve{G}_{jj} = 0$.

What remains to be shown is that in (10.51) and (10.52), it holds that

$$\breve{H}_{j3}\breve{G}_{\mathcal{U}j} = \breve{H}_{j3}\breve{G}_{\mathcal{U}i} = 0 \qquad (3.47)$$

while additionally for $i \in \mathcal{U}$, it should hold that

$$\breve{H}_{ji} = 0. \qquad (3.48)$$

With definition (3.16) for $\breve{H}$ and the special structure of $\tilde{H}_{13}$ and $\tilde{H}_{23}$ in (3.14) that is implied by the result (4.37) of Lemma 3.3, we can write

$$\begin{bmatrix} \check{H}_{13} \\ \check{H}_{23} \end{bmatrix} = \begin{bmatrix} \tilde{H}_{\mathcal{Q}\mathcal{B}} & 0 \\ \tilde{H}_{\mathcal{O}\mathcal{B}} & 0 \end{bmatrix} \begin{bmatrix} \tilde{H}_{\mathcal{B}\mathcal{B}} & 0 \\ 0 & \tilde{H}_{\mathcal{A}\mathcal{A}} \end{bmatrix}^{-1} = \begin{bmatrix} \check{H}_{\mathcal{Q}\mathcal{B}} & 0 \\ \check{H}_{\mathcal{O}\mathcal{B}} & 0 \end{bmatrix}, \tag{3.49}$$

implying that columns in this matrix related to inputs $k \in \mathcal{A}$ are zero.
In order to satisfy (4.68) we need the condition that: if $i \in \mathcal{U}$ then $i \in \mathcal{A}$. This is equivalently formulated as $i \in \mathcal{Q} \cup \mathcal{A}$ (conditon 2b).
In order to satisfy (4.67) we note that $\check{H}_{j3}$ is a row vector, of which the second part (the columns related to signals in $\mathcal{A}$) is equal to 0, according to (4.40). Consequently, (4.67) is satisfied if for every $k \in \mathcal{B}$ it holds that $\breve{G}_{kj} = \breve{G}_{ki} = 0$. On the basis of (3.12), this condition is satisfied if for every $w_k \in w_{\mathcal{B}}$ there do not exist direct or unmeasured paths from w_i to w_k and from w_j to w_k (condition 2c). □

3.C Proof of Theorem 3.2

Expression (3.4) can be written as

$$w_{\mathcal{Y}} = \bar{G}^o w_{\mathcal{D}} + \bar{H}^o \xi_{\mathcal{Y}}.$$

Substituting this into the expression for the prediction error (4.5), leads to

$$\varepsilon(t,\theta) := \bar{H}(q,\theta)^{-1} \left[\Delta\bar{G}(q,\theta) w_{\mathcal{D}} + \Delta\bar{H}(q,\theta)\xi_{\mathcal{Y}}\right] + \xi_{\mathcal{Y}} \tag{3.50}$$

where $\Delta\bar{G}(q,\theta) = \bar{G}^o - \bar{G}(q,\theta)$ and $\Delta\bar{H}(q,\theta) = \bar{H}^o - \bar{H}(q,\theta)$. The proof of consistency involves two steps.

1. To show that $\mathbb{E}\varepsilon^T(t,\theta)W\varepsilon(t,\theta)$ achieves its minimum for $\Delta\bar{G}(\theta) = 0$ and $\Delta\bar{H}(\theta) = 0$,
2. To show the conditions under which the minimum is unique.

Step 1: With Proposition 3.1 it follows that our data generating system can always be written in the form (3.3), such that $w_m = T(q)\xi_m$. We denote T_1 as the matrix composed of the first and third (block) row of T, such that $w_{\mathcal{D}} = T_1(q)\xi_m$. Substituting this into (4.43) gives

$$\varepsilon(t,\theta) := \bar{H}(q,\theta)^{-1} \left[\Delta\bar{G}(q,\theta)T_1 + \begin{bmatrix}\Delta\bar{H}(\theta) & 0\end{bmatrix}\right] \xi_m + \xi_{\mathcal{Y}},$$

where ξ_m is (block) structured as $[\xi_{\mathcal{Y}}^\top \; \xi_{\mathcal{U}}^\top]^\top$.
In order to prove that the minimum of $\mathbb{E}\left[\varepsilon^T(t,\theta)W\varepsilon(t,\theta)\right]$ is attained for $\Delta\bar{G}(\theta) = 0$ and $\Delta\bar{H}(\theta) = 0$, it is sufficient to show that

$$\left[\Delta\bar{G}(\theta)T_1(q) + \begin{bmatrix}\Delta\bar{H}(\theta) & 0 & 0\end{bmatrix}\right] \xi_m(t) \tag{3.51}$$

is uncorrelated to $\xi_{\mathcal{Y}}(t)$. In order to show this, let $\mathcal{F}_n = \mathcal{U}\backslash\mathcal{F}$, with $\mathcal{F}$ as defined in the Theorem, while we decompose ξ_m according to $\xi_m = [\xi_{\mathcal{Y}}^\top \quad \xi_{\mathcal{F}}^\top \quad \xi_{\mathcal{F}_n}^\top]^\top$. Using a similar block-structure notation for $\Delta\bar{G}, T$ and $\Delta\bar{H}$, (4.46) can then be written as

$$\begin{aligned}&\left(\Delta\bar{G}_{\mathcal{Y}\mathcal{Q}}(\theta)T_{\mathcal{Q}\mathcal{Y}} + \Delta\bar{G}_{\mathcal{Y}\mathcal{F}}(\theta)T_{\mathcal{F}\mathcal{Y}} + \Delta\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)T_{\mathcal{F}_n\mathcal{Y}} + \Delta\bar{H}_{\mathcal{Y}\mathcal{Y}}(\theta)\right)\xi_{\mathcal{Y}}+\\ &+\left(\Delta\bar{G}_{\mathcal{Y}\mathcal{Q}}(\theta)T_{\mathcal{Q}\mathcal{F}} + \Delta\bar{G}_{\mathcal{Y}\mathcal{F}}(\theta)T_{\mathcal{F}\mathcal{F}} + \Delta\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)T_{\mathcal{F}_n\mathcal{F}}\right)\xi_{\mathcal{F}}\\ &+\left(\Delta\bar{G}_{\mathcal{Y}\mathcal{Q}}(\theta)T_{\mathcal{Q}\mathcal{F}_n} + \Delta\bar{G}_{\mathcal{Y}\mathcal{F}}(\theta)T_{\mathcal{F}\mathcal{F}_n} + \Delta\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)T_{\mathcal{F}_n\mathcal{F}_n}\right)\xi_{\mathcal{F}_n}.\end{aligned} \tag{3.52}$$

Since, by definition, $\xi_{\mathcal{F}_n}(t)$ is statically uncorrelated to $\xi_{\mathcal{Y}}(t)$, the $\xi_{\mathcal{F}_n}$-dependent term in (4.47) cannot create any static correlation with $\xi_{\mathcal{Y}}(t)$. Then it needs to be shown that the $\xi_{\mathcal{Y}}$- and $\xi_{\mathcal{F}}$-dependent terms in (4.47) all reflect strictly proper filters. i.e. that they all contain at least a delay.
$\Delta\bar{H}(\theta)$ is strictly proper since both $\bar{H}(\theta)$ and $\bar{H}^o$ are monic. Therefore, $\Delta\bar{H}_{\mathcal{Y}\mathcal{Y}}(\theta)$ will have at least a delay in each of its transfers.
If all paths from $w_{\mathcal{Y}\cup\mathcal{F}}$ to $w_{\mathcal{Y}}$ in the *transformed network* and in its parameterized model have at least a delay (as per Condition c in the theorem), then all terms $\Delta\bar{G}_{\mathcal{Y}\mathcal{Q}}(\theta)$ and $\Delta\bar{G}_{\mathcal{Y}\mathcal{F}}(\theta)$ will have a delay.
We then need to consider the two remaining terms, $\Delta\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)T_{\mathcal{F}_n\mathcal{Y}}$ and $\Delta\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)T_{\mathcal{F}_n\mathcal{F}}$. From the definition of $\Delta\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)$, each of the two terms can be represented as the sum of two terms. $\bar{G}_{\mathcal{Y}\mathcal{F}_n}T_{\mathcal{F}_n\mathcal{Y}}$ and $\bar{G}_{\mathcal{Y}\mathcal{F}_n}T_{\mathcal{F}_n\mathcal{F}}$ represent paths from $w_{\mathcal{Y}}$ to $w_{\mathcal{Y}}$ and from $w_{\mathcal{F}}$ to $w_{\mathcal{Y}}$ respectively in the *transformed network*. Whereas, $\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)T_{\mathcal{F}_n\mathcal{Y}}$ and $\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)T_{\mathcal{F}_n\mathcal{F}}$ is partly induced by the parameterized model and partly by the paths from $w_{\mathcal{Y}}$ to $w_{\mathcal{F}_n}$ and from $w_{\mathcal{F}}$ to $w_{\mathcal{F}_n}$ respectively in the *transformed network*. According to condition c of the theorem (delay conditions), these transfer functions are strictly proper. This implies that (4.47) is statically uncorrelated to $\xi_{\mathcal{Y}}(t)$. Therefore we have, $\bar{\mathbb{E}}\left[\varepsilon^T(t,\theta)W\varepsilon(t,\theta)\right] = \bar{\mathbb{E}}\left[||\Delta X(\theta)\xi_m||_W\right] + \bar{\mathbb{E}}\left[\xi_{\mathcal{Y}}^\top W\xi_{\mathcal{Y}}\right]$ where $\Delta X(\theta) = \bar{H}(\theta)^{-1}\left[\Delta\bar{G}(\theta)T_1(q) + \left[\Delta\bar{H}(\theta) \quad 0 \quad 0\right]\right]$. As a result, the minimum of $\bar{\mathbb{E}}\left[\varepsilon^T(t,\theta)W\varepsilon(t,\theta)\right]$, which is $\bar{\mathbb{E}}\left[\xi_{\mathcal{Y}}^\top W\xi_{\mathcal{Y}}\right]$, is achieved for $\Delta\bar{G}(\theta) = 0$ and $\Delta\bar{H}(\theta) = 0$.

Step 2: When the minimum is achieved, we have $\bar{\mathbb{E}}\left[||\Delta X(\theta)\xi_m||_W\right]$ to be zero. From (4.43), we have $\Delta X(\theta)\xi_m = \bar{H}(q,\theta)^{-1}\left[\left[\Delta\bar{G}(q,\theta) \quad \Delta\bar{H}(q,\theta)\right]\left[w_{\mathcal{D}}^\top \quad \xi_{\mathcal{Y}}^\top\right]^\top\right]$. Using the expression of ξ_o from (3.4) and substituting it in the expression of $\Delta X(\theta)\xi_m$ we get, $\Delta X(\theta)\xi_m = \bar{H}(q,\theta)^{-1}\left[\left[\Delta\bar{G}(q,\theta) \quad \Delta\bar{H}(q,\theta)\right]J\kappa(t)\right] = \Delta x(\theta)J\kappa(t)$ where,

$$J = \begin{bmatrix} I & 0 & 0\\ 0 & I & 0\\ -(\bar{H}_{oo})^{-1}\bar{G}_{\mathcal{O}\mathcal{D}} & -(\bar{H}_{oo})^{-1}\bar{H}_{\mathcal{O}\mathcal{Q}} & (\bar{H}_{oo})^{-1}\end{bmatrix}; \bar{G}_{\mathcal{O}\mathcal{D}}^\top = \begin{bmatrix}\bar{G}_{\mathcal{O}\mathcal{Q}}^\top\\ \bar{G}_{\mathcal{O}\mathcal{U}}^\top\end{bmatrix}.$$

Writing $\bar{\mathbb{E}}\left[||\Delta X(\theta)\xi_m||_W\right] = \bar{\mathbb{E}}\left[||\Delta x(\theta)J\kappa(t)||_W\right] = 0$ using Parseval's theorem in the frequency domain, we have

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \Delta x(e^{j\omega},\theta)^\top J\Phi_\kappa(\omega)J^*\Delta x(e^{-j\omega},\theta)d\omega = 0. \tag{3.53}$$

The standard reasoning for showing uniqueness of the identification result is to show that if $\bar{\mathbb{E}}\left[||\Delta X(\theta)\xi_m||_W\right]$ equals 0 (i.e. when the minimum power is

achieved), this should imply that $\Delta\bar{G}(\theta) = 0$ and $\Delta\bar{H}(\theta) = 0$. Since J is full rank and positive definite, the above mentioned implication will be fulfilled only if $\Phi_\kappa(\omega) > 0$ for a sufficiently high number of frequencies. On condition 2 of Theorem 3.2 being satisfied along with the other conditions in Theorem 1, it ensures that the minimum value is achieved only for $\bar{G}(\theta) = \bar{G}^0$ and $\bar{H}(\theta) = \bar{H}^0$.
□

3.D Proof of Proposition 3.2

The disturbances in the original network are characterized by $\breve{v}$ (3.11). From the results of Lemma 3.3, we can infer that the spectral density $\Phi_{\breve{v}}$ has the unique spectral factorization $\Phi_{\breve{v}} = \tilde{H}\Lambda\tilde{H}^*$ where $\tilde{H}$ is monic, stable, minimum phase, and of the form given in (4.37). Together with the form of Λ in (4.37) it follows that $\xi_{\mathcal{A}}$ is uncorrelated with $\xi_{\mathcal{Y}}$. As a result, the set $\mathcal{A}$ satisfies the properties of $\mathcal{F}_n$, so that in Condition c we can replace $\mathcal{F}$ by $\mathcal{B}$. What remains to be shown is that the delay in path/loop conditions in the transformed network (3.4) can be reformulated into the same conditions on the original network (3.2). To this end we will need two Lemma's.

Lemma 3.4 *Consider a dynamic network as dealt with in Theorem 3.2, with reference to eq. (3.4), where a selection of node signals is decomposed into sets $\mathcal{D} = \mathcal{Q} \cup \mathcal{U}$, $\mathcal{Y} = \mathcal{Q} \cup \{o\}$, and which is obtained after immersion of nodes in $\mathcal{Z}$. Let i be any element $i \in \mathcal{Y} \cup \mathcal{U}$, and let k be any element $k \in \mathcal{Y}$.*
If in the original network the direct path, as well as all paths that pass through non-measured nodes only, from w_i to w_k have a delay, then $\bar{G}_{ki}$ is strictly proper.

Proof: We will show that $\bar{G}_{ki}$ is strictly proper if all paths from w_i to w_k have a delay. For any $k \in \mathcal{Y}$, $i \in \mathcal{D}$, $\bar{G}_{ki}$ is given by either (10.51) or (10.52) with $j = k$. The situation that is not covered by (10.51), (10.52) is the case where $i = \{o\}$, but from (3.30) it follows that $\bar{G}_{ko} = 0$, for $k \in \mathcal{Y}$. So for this situation strictly properness is guaranteed.
We will now use (10.51) and (10.52) for j given by any $k \in \mathcal{Y}$. In (10.51) and (10.52), it will hold that $\breve{H}_{k3}$ is given by the appropriate component of (3.16), which, by the fact that (3.14) is monic, will imply that $\breve{H}_{k3}$ is strictly proper. By the same reasoning this also holds for $\breve{H}_{ki}$.
From (10.51) and (10.52) it then follows that strictly properness of $\bar{G}_{ki}$ follows from strictly properness of $\breve{G}_{ki}$ if the inverse expression $(I - \breve{G}_{kk} + \breve{H}_{k3}\breve{G}_{\mathcal{U}k})^{-1}$ is proper. This latter condition is guaranteed by the fact that $\breve{H}_{k3}$ is strictly proper and $\breve{G}_{kk}$ and $(I - \breve{G}_{kk})^{-1}$ are proper as they reflect a module and network transfer function in the immersed network [141, 149]. Finally, strictly properness of $\breve{G}_{ki}$ follows from strictly properness of G_{ki} and the presence of a delay in all paths from w_i to w_k that pass through unmeasured nodes.

Lemma 3.5 *Consider the transformed network and let j, k be any elements $j, k \in \mathcal{Y} \cup \mathcal{U}$. If in the original network all paths from w_k to w_j have a delay, then all paths from w_k to w_j in the transformed network have a delay.*

Proof: This is proved using the Lemma 3 in [124] and Lemma 3.4. Let $\bar{G}(\infty)$ denote $\lim_{z\to\infty} \bar{G}(z)$. From Lemma 3.4 we know $\bar{G}_{jk}$ is strictly proper if all paths from w_k to w_j in the original network have a delay. Therefore,

$$\bar{G}_m(\infty) = \begin{bmatrix} * & 0 \\ * & * \end{bmatrix}, \tag{3.54}$$

where the 0 represents $\bar{G}_{jk}(\infty)$. Using inverse rule of block matrices we have,

$$(I - \bar{G}_m(\infty))^{-1} = \begin{bmatrix} * & 0 \\ * & * \end{bmatrix} \tag{3.55}$$

Considering (3.3) we can write $w_m = \bar{G}_m w_m + v_m$ where $v_m = \bar{H}_m \xi_m$. So have $w_m = (I - \bar{G}_m)^{-1} v_m$ where $(I - \bar{G}_m)^{-1}$ represents the transfer from v_m to w_m. Having 0 in (3.55) represents that the transfer function from v_k to w_j has a delay. Since v_k has path only to w_k with unit transfer function, w_k to w_j has a delay. □

We now look into the proof of Proposition 3.2. For this we need to generalize the result we have achieved in Lemma 3.5 in terms of scalar node signals to set of node signals. If all existing paths/loops from $w_{\mathcal{Y}\cup\mathcal{F}}$ to $w_{\mathcal{Y}}$ in the original network have at least a delay, then all existing paths/loops from $w_k, k \in \mathcal{Y} \cup \mathcal{F}$ to $w_j, j \in \mathcal{Y}$ in the original network have at least a delay. If all existing paths/loops from $w_k, k \in \mathcal{Y} \cup \mathcal{F}$ to $w_j, j \in \mathcal{Y}$ in the original network have at least a delay, then as a result of Lemma 3.5, all existing paths/loops from $w_k, k \in \mathcal{Y} \cup \mathcal{F}$ to $w_j, j \in \mathcal{Y}$ in the transformed network have at least a delay. This implies that all existing paths/loops from $w_k, k \in \mathcal{Y} \cup \mathcal{F}$ to $w_j, j \in \mathcal{Y}$ in the transformed network have at least a delay. Following the above reasoning, we can also show that if all existing paths from $w_{\mathcal{Y}\cup\mathcal{F}}$ to $w_k, k \in \mathcal{F}_n$ in the original network have at least a delay, all existing paths from $w_{\mathcal{Y}\cup\mathcal{F}}$ to $w_k, k \in \mathcal{F}_n$ in the transformed network have at least a delay.

4 CHAPTER

A Generalized method for flexible signal selection

For the problem of identifying a target module that is embedded in a dynamic network with known interconnection structure, different sets of conditions are available for the set of node signals to be measured and the set of excitation signals to be applied at particular node locations. In previous work these conditions have typically been derived from either an indirect identification approach, considering external excitation signals as predictor inputs, or from a direct identification approach, considering measured node signals as predictor inputs. While both approaches lead to different sets of (sufficient) conditions, in this chapter we extend the flexibility in the sufficient conditions for selection of excitation and measured node signals, by combining both direct and indirect approaches. As a result we will show the benefits of using both external excitation signals and node signals as predictor inputs. The provided conditions allow us to design sensor selection and actuation schemes with considerable freedom for consistent identification of a target module.

4.1 Introduction

An important condition in the works that use the direct method [124], [29], [30], [104] is that all parallel paths from the input of the target module to its output

This chapter is based on the preliminary work: K.R. Ramaswamy, P.M.J. Van den Hof and A.G. Dankers, "Generalized sensing and actuation schemes for local module identification in dynamic networks", in *Proc. 58th IEEE Conf. Decision and Control (CDC)*, Nice, France, 11-13 December 2019, pp. 5519-5524.

and all loops through the output node should pass through a measured node signal that is included as a predictor input (see Condition 3.1 in Chapter 3). This requirement ensures that the identified module after immersion of the non-measured nodes converges to the target module. However, in practical situations, there can be parallel paths and loops that might have all nodes non-measured. This creates a restriction for the selection of measured node signals and application of the direct method.

In the *indirect method* as in [54], [7], external excitation signals are used as predictor inputs for an open loop MIMO identification problem. These methods involve two steps: (1) First obtain consistent estimates of a transfer function from external excitation signals to measured node signals; (2) Using these estimates to obtain consistent estimates of the target module (we call this step *post-processing*). However, since they rely on external excitation signals, the indirect methods have restrictive conditions on requirement of external excitation signals.

In the work of this chapter, we increase the flexibility in the sufficient conditions for the selection of excitation and measured node signals for consistent target module estimates and thereby generalize the sensing and actuation schemes. We relax the above-discussed condition on the parallel paths and loops around the output node. This relaxation in the condition is achieved by combining elements of both direct and indirect approaches. We use both the node signals and external excitation signals as predictor inputs (element of direct method) and allow post-processing of module estimates (element of indirect method), thereby mixing both direct and indirect methods. The provided conditions allow us to design sensor selection and actuation schemes with considerably more freedom for consistent identification of a target module.

4.2 Motivating example

In this section, we highlight the motivation of the work in this chapter using a suitable example. In Chapter 2 of this thesis, it has been shown that we can identify a target module G_{ji}^0 consistently provided that we choose the selection of predictor input signals to satisfy particular conditions. One of the main conditions is the parallel path/loop Condition 2.1. When this condition is satisfied, using a MISO identification setup with w_j as predicted output and $w_{\mathcal{D}_j}$ as predictor inputs, the *MISO direct method* as discussed in Section 2.5.1 provides a consistent estimate of the target module, if in addition there are no confounding variables for the estimation problem $w_{\mathcal{D}_j} \rightarrow w_j$. In [30] additional conditions have been formulated for the selection of $\mathcal{D}_j$ so as to avoid the presence of confounding variables, typically by choosing additional predictor inputs defined by the set $\mathcal{B}_j \subset \mathcal{L}\backslash\{\mathcal{D}_j \cup \{j\}\}$. This situation has been analyzed for the case where all disturbance signals v are mutually uncorrelated, i.e. its spectral density Φ_v being diagonal. In this situation it is still required that $\mathcal{D}_j$ satisfies condition 2.1. This restrictive condition is required for the target module in the dynamic network to be invariant in an immersed network where all non-measured signals are being removed [29].

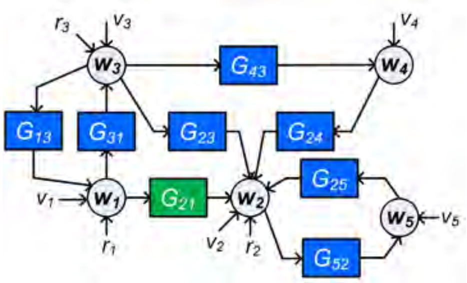


Figure 4.1: Example network

Example 4.1 *Consider a dynamic network as represented in Figure 4.1 with all noises in v uncorrelated with each other i.e. Φ_v is diagonal. For identifying the target module G_{21} (in green box), we have $j = 2$, and in order to satisfy condition 2.1 we need $\mathcal{D}_j = \{1, 3, 5\}$ where w_3 is included to block the parallel path from w_1 to w_2, and w_5 is included to block the loop through w_2. Using this set of measured nodes, we arrive at an immersed network after removing the non-measured node as in Figure 4.2. We can observe that the module between w_1 and w_2 (the green box) is G_{21} and remains invariant.*

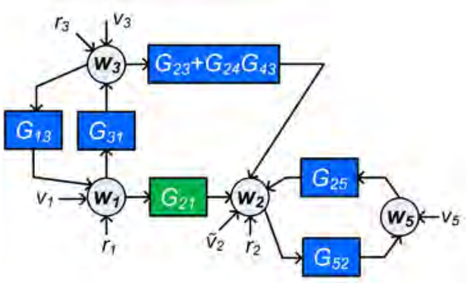


Figure 4.2: Immersed network of network in Figure 4.1 [29] where the nonmeasured node w_4 has been removed (immersed), and where $\tilde{v}_2 = v_2 + G_{24}v_4$.

If w_3 and w_5 are not selected in $\mathcal{D}_j$, and so $\mathcal{D}_j = \{1\}$, we arrive at an immersed network after removing all non-measured nodes, as depicted in Figure 4.3. We can now observe that the dynamic module between w_1 and w_2 (the green box in Figure 4.3) is not equal to G_{21}. The terms $G'_{23}G_{31}$ and $(1-G_{25}G_{52})^{-1}$ are due to the fact that in this situation the parallel path and loop condition 2.1 is not satisfied. In the work of this chapter we are going to relax the restrictive condition 2.1 and increase the freedom in the selection of measured node signals.

For the approach based on the indirect identification method, in [54] a method has been presented to identify a target module using external signals as predictor inputs, along the following reasoning.

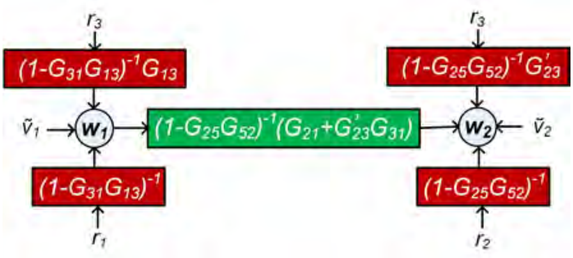

r_3
$(1-G_{31}G_{13})^{-1}G_{13}$
r_3
$(1-G_{25}G_{52})^{-1}G'_{23}$
$\tilde{v}_1$
w_1
$(1-G_{25}G_{52})^{-1}(G_{21}+G'_{23}G_{31})$
w_2
$\tilde{v}_2$
$(1-G_{31}G_{13})^{-1}$
$(1-G_{25}G_{52})^{-1}$
r_1
r_2

Figure 4.3: Immersed network of network in figure 4.1[29] where the non-measured nodes w_3, w_4, w_5 have been removed (immersed), and where $G'_{23} = (G_{23} + G_{24}G_{43})$, $\tilde{v}_1 = (1 - G_{31}G_{13})^{-1}(v_1 + G_{13}v_3)$ and $\tilde{v}_2 = (1 - G_{25}G_{52})^{-1}\,(v_2 + (G_{23} + G_{24}G_{43})v_3 + G_{24}v_4 + G_{25}v_5)$.

Proposition 4.1 (from [54]) *In order to identify a target module G_{ji}, perform the following experiment:*

1. *Excite node w_i and all its out-neighbors with sufficiently rich signals. Include these excitation signals as predictor inputs;*
2. *Measure the out-neighbors of w_i. Include them as predicted outputs.*

Under these conditions and using full order models for the elements of T (where T relates the r signals to node signals, see Chapter 2), consistent estimates $\hat{T}_{\mathcal{N}_i^+\mathcal{N}_i^+}$, $\hat{T}_{\mathcal{N}_i^+i}$ of $\bar{T}_{\mathcal{N}_i^+\mathcal{N}_i^+}$ and $\bar{T}_{\mathcal{N}_i^+i}$ can be obtained using an open loop MIMO identification method as given in Section 2.5.2. Then a consistent estimate of $\hat{G}_{\mathcal{N}_i^+i}$ (which includes the target module) is obtained by,

$$\hat{G}_{\mathcal{N}_i^+i} = [\hat{T}_{\mathcal{N}_i^+\mathcal{N}_i^+}]^{-1}\hat{T}_{\mathcal{N}_i^+i} \qquad \square \qquad (4.1)$$

A dual of this proposition with w-in-neighbors of w_j, replacing the w-out neighbors of w_i, is also provided in [54] (provided in Section 2.5.2 of this thesis). It can be observed that a consistent estimate of the target module is obtained from consistent estimates of elements of T. We will refer to this step (4.1) of manipulating identified objects, as *post-processing*. Considering the earlier Example 4.1, we can now consistently identify our target module using an open loop MIMO identification setup with $\{r_1, r_2, r_3\}$ as inputs and $\{w_2, w_3\}$ as outputs. However this requires restrictive conditions on the nodes to be excited and nodes to be measured, i.e. measured excitation signals r_1, r_2, r_3. Further relaxations of these restrictive conditions on excitation and measured node signals will be addressed in the sequel.

4.3 Illustration of the Generalized method

In this section we illustrate the developed method in this chapter with suitable examples. In the work of this chapter, we combine the ideas of both the direct and indirect method such that we increase flexibility in the selection of excitation and measured node signals, and exploit all excitations. We use both the measured node signals as well as the excitation signals as predictor inputs. In addition to that, we do not restrict to the situation of invariance of our target module after immersion as in the direct method, but use the mechanism of *post-processing* from the indirect method to consistently identify the target module.

Example 4.2 *We now consider the same network as in Example 4.1 but with two constraints: (a) it is not possible to measure w_3 and w_5; (b) it is not possible to excite node w_1. It can be inferred that it is not possible to consistently estimate $G_{ji} = G_{21}$ using the direct method due to constraint (a). Similarly due to constraint (b), it is not possible with the indirect method either.*

As shown in Example 4.1, if we do not measure w_3 and w_5 our target module changes to $(1 - G_{25}G_{52})^{-1}(G_{21} + G_{23}G_{31})$ in the immersed network. However, we can see that this module also contains the target module of interest G_{21}. Therefore we might extract the target module from this term if we know (or) find the other contributions.

Consider the situation that we excite node w_3 and measure node w_4. After immersing the non-measured nodes (see Chapter 2 on immersion) we end up in a dynamic network setup as in Figure 4.4. Now consider the identification problem $\{w_1, w_4, r_2, r_3\} \rightarrow \{w_2, w_4\}$. We can infer the following from the figure:

1. *Identifying the transfer from $r_3 \rightarrow w_4$ provides G_{43} and the transfer from $w_1 \rightarrow w_4$ provide $G_{43}G_{31}$. Thus we can identify G_{31};*
2. *The transfer from $r_3 \rightarrow w_2$ provides $(1 - G_{25}G_{52})^{-1}G_{23}$. The term $(1 - G_{25}G_{52})^{-1}$ is due to the fact that in the original network there is a loop around w_2 which is not "blocked" by a measured node. This term given by the transfer from $r_2 \rightarrow w_2$. As a result, we can obtain G_{23}.*
3. *The term $G_{23}G_{31}$ is due to the fact that in the original network there is a path from w_1 to w_2 through w_3 which is not "blocked" by a measured node. Knowing G_{23} and G_{31} from the above two steps, we obtain the term $G_{23}G_{31}$. We also have $(1 - G_{25}G_{52})^{-1}$. Eventually we obtain our target module of interest from the transfer $w_1 \rightarrow w_2$ (i.e. $(1 - G_{25}G_{52})^{-1}(G_{21} + G_{23}G_{31})$).*

This shows that we can consistently identify the target module G_{21} if we know or could consistently identify the transfer from

$\{w_1, w_4, r_2, r_3\} \rightarrow \{w_2, w_4\}$. In order to achieve this we move to a MIMO identification problem using the prediction error method with $\{w_2, w_4\}$ as predicted outputs and $\{w_1, w_4, r_2, r_3\}$ as predictor inputs.

Remark 4.1 *The excitation r_2 at the output node is required as predictor input since the loops through w_j are not blocked by a measured node. However we can still relax this under certain conditions, which will be elaborated upon in this chapter.*

Remark 4.2 *The consistency results will still require some excitation conditions, which will be specified later on.*

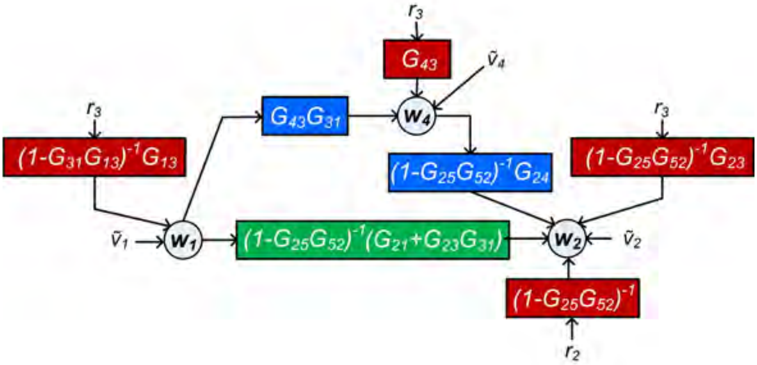


Figure 4.4: Immersed network of network in Figure 4.1 where the nonmeasured nodes w_3, w_5 have been removed (immersed), and where $\tilde{v}_1 = (1-G_{31}G_{13})^{-1}(v_1 + G_{13}v_3)$, $\tilde{v}_2 = (1 - G_{25}G_{52})^{-1}(v_2 + G_{23}v_3 + G_{25}v_5)$ and $\tilde{v}_4 = v_4 + G_{43}v_3$.

In the previous chapter, we saw that the confounding variable for an estimation problem can be handled by adding either predicted outputs or predictor inputs. In the same way, we can handle the confounding variables that occur for the estimation problem in this chapter as well. We can observe from Figure 4.4 that the noise at predictor input w_1 and at predicted outputs w_2, w_4 are correlated due to v_3. This is due to the fact that in the original network, v_3 (in turn e_3) has simultaneous paths to w_1 and w_2 (also w_1 and w_4), while these paths run through the unmeasured node w_3. Therefore e_3, which is a confounding variable, creates noise correlation between predictor inputs and predicted outputs. When using the prediction error framework with the MIMO setup as explained above (i.e. with $\{w_2, w_4\}$ as predicted outputs), we only model the noise from $\{e_2, e_4\} \rightarrow \{w_2, w_4\}$ but not from the confounding variable e_3. This leads to a lacking consistency property of the identified modules, as described in Chapter 3. If we also predict w_1 (include it also as predicted output), we now model the noise from e_3 as well. This leads to consistent estimates. Therefore for Example 2, we need the MIMO identification setup $\{w_1, w_4, r_2, r_3\} \rightarrow \{w_1, w_2, w_4\}$.

From the discussed example, we can now conjecture the following generalization:

1. Violating the parallel path condition can be handled by exciting a node in the parallel path, including the excitation signal in the predictor input, and by measuring a descendant node from the excited node, different from the output and need not belong to the parallel path, and by including this descendant node in the predicted output;
2. Violating the loop condition can be handled by either
   - exciting the output node and including the excitation signal in the predictor input; or
   - exciting a node in the loop, including the excitation signal in the predictor input, and by measuring a descendant node from the excited node, different from the output and need not belong to the loop, and by including this descendant node in the predicted output;
3. Confounding variables can be handled by including measured nodes as predicted outputs[1].

Remark 4.3 *In handling the parallel path condition it will appear that we actually have to add one additional constraint (see Property 4.1 later on). If the mentioned descendant node is the input of the target module, then this node needs to be excited with an external signal, which is included as predictor input.*

Remark 4.4 *If we consider again Example 4.1 in Figure 4.1, then the parallel path problem that occurs when w_3 can not be measured, can be compensated for by measuring a descendant from w_3, which in this case could also be w_1. Since w_1 is the input of the target module, the previous remark now leads to the situation that w_1 also needs to be excited, which is a situation that was excluded in Example 2.*

In the sequal of this chapter, we will derive the formal results that underly the above conjectured statements.

4.4 Main results - Line of reasoning

Similar to the previous chapter, we will denote $w_{\mathcal{Y}}$ as the node signals in w that serve as predicted outputs, and $w_{\mathcal{D}}$ as the node signals in w that serve as predictor

[1] Confounding variables can also be handled in other ways, for example, adding predictor inputs (see Chapter 3). In this chapter we handle using predicted outputs in order to avoid measurement of additional node signals.

inputs. Next we decompose $w_{\mathcal{Y}}$ and $w_{\mathcal{D}}$ into disjoint sets according to: $\mathcal{Y} = \mathcal{Q} \cup \mathcal{O}$ and $\mathcal{D} = \mathcal{Q} \cup \mathcal{U}$ where $w_{\mathcal{Q}}$ are the node signals that are common in $w_{\mathcal{Y}}$ and $w_{\mathcal{D}}$; $w_{\mathcal{O}}$ is the set of node signals that are only predicted outputs; $w_{\mathcal{U}}$ are the node signals that are only in $w_{\mathcal{D}}$. Additionally we denote $w_{\mathcal{Z}}$ as the node signals in w that are neither predicted output nor predictor input, i.e. $\mathcal{Z} = \mathcal{L} \backslash \{\mathcal{D} \cup \mathcal{Y}\}$, where $\mathcal{L} = \{1, 2, \cdots L\}$.

Consider a dynamic network defined by (3.1), however with $cov(e) = I$ and H^0 not necessarily monic as considered in Chapter 3 (we refer to this network as the *original network* in this chapter). Our objective is to end up with an identification problem in which we identify the dynamics from inputs $(w_{\mathcal{Q}}, w_{\mathcal{U}}, u)$ to outputs $(w_{\mathcal{Q}}, w_{\mathcal{O}})$, while our target module $G_{ji}(q)$ can be retrieved possibly through *post-processing* from the elements of the identified (MIMO) model. This can be realized by the following steps[2]:

1. Firstly, similar to (3.3), we write the system's equations for the measured variables as

$$\underbrace{\left[\begin{array}{c} w_{\mathcal{Q}} \\ w_{\mathcal{O}} \\ \hline w_{\mathcal{U}} \end{array}\right]}_{w_m} = \underbrace{\left[\begin{array}{c|c} \bar{G} & 0 \\ \hline \bar{G}_{\mathcal{UD}} & \bar{G}_{\mathcal{UO}} \end{array}\right]}_{\bar{G}_m} \left[\begin{array}{c} w_{\mathcal{Q}} \\ w_{\mathcal{U}} \\ \hline w_{\mathcal{O}} \end{array}\right] + \underbrace{\left[\begin{array}{c|c} \bar{H} & 0 \\ \hline 0 & \bar{H}_{\mathcal{UU}} \end{array}\right]}_{\bar{H}_m} \underbrace{\left[\begin{array}{c} \xi_{\mathcal{Q}} \\ \xi_{\mathcal{O}} \\ \hline \xi_{\mathcal{U}} \end{array}\right]}_{\xi_m} + \underbrace{\left[\begin{array}{c} \bar{R} \\ \hline \bar{R}_{\mathcal{U}} \end{array}\right]}_{\bar{R}_m} u \tag{4.2}$$

   with ξ_m a white noise process, while $\bar{H}$ is monic, stable and stably invertible and the components in $\bar{G}$ are zero if it concerns a mapping between identical signals. This step is made by removing the non-measured signals $w_{\mathcal{Z}}$ from the network, while maintaining the second order properties of the remaining signals. This step is referred to as immersion of the nodes in $w_{\mathcal{Z}}$ [29]. After immersion, we re-write the system's equation to structure the noise model such that there are no confounding variables for the estimation problem $w_{\mathcal{U}} \rightarrow w_{\mathcal{Y}}$.

2. As an immediate result of the previous step we can write an expression for the output variables $w_{\mathcal{Y}}$, by considering the upper part of the equation (4.2), as

$$\underbrace{\begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{O}} \end{bmatrix}}_{w_{\mathcal{Y}}} = \underbrace{\begin{bmatrix} \bar{G}_{\mathcal{QQ}} & \bar{G}_{\mathcal{QU}} \\ \bar{G}_{\mathcal{OQ}} & \bar{G}_{\mathcal{OU}} \end{bmatrix}}_{\bar{G}} \underbrace{\begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{U}} \end{bmatrix}}_{w_{\mathcal{D}}} + \underbrace{\begin{bmatrix} \bar{H}_{\mathcal{QQ}} & \bar{H}_{\mathcal{QO}} \\ \bar{H}_{\mathcal{OQ}} & \bar{H}_{\mathcal{OO}} \end{bmatrix}}_{\bar{H}} \underbrace{\begin{bmatrix} \xi_{\mathcal{Q}} \\ \xi_{\mathcal{O}} \end{bmatrix}}_{\xi_{\mathcal{Y}}} + \bar{R}u \tag{4.3}$$

   with $cov(\xi_{\mathcal{Y}}) := \bar{\Lambda}$.

3. Thirdly, it will be shown that, on the basis of (4.3), under fairly general conditions, the transfer functions $\bar{G}(q)$, $\bar{H}(q)$ and $\bar{R}(q)$ can be estimated consistently.

4. Finally, we will provide conditions to guarantee that the target module G_{ji}^0 can be obtained from the identified elements of $\bar{G}(q)$, $\bar{H}(q)$ and $\bar{R}(q)$ in (4.3) *(i.e. post-processing)*. This will require conditions on the selection of node

[2] From now on, ^{0} is dropped for convenience.

signals in $w_{\mathcal{Q}}, w_{\mathcal{O}}, w_{\mathcal{U}}$ and excitation signals in u. We will introduce two additional sets $\mathcal{Z}_r$ and $\mathcal{T}$ in the sequel of this chapter. These two sets will play a major role in extracting the target module estimate from the identification result $\bar{G}, \bar{R}$. A pictorial representation of the identification setup with the classification of different sets of signals in (4.3) is provided in Figure 4.5.

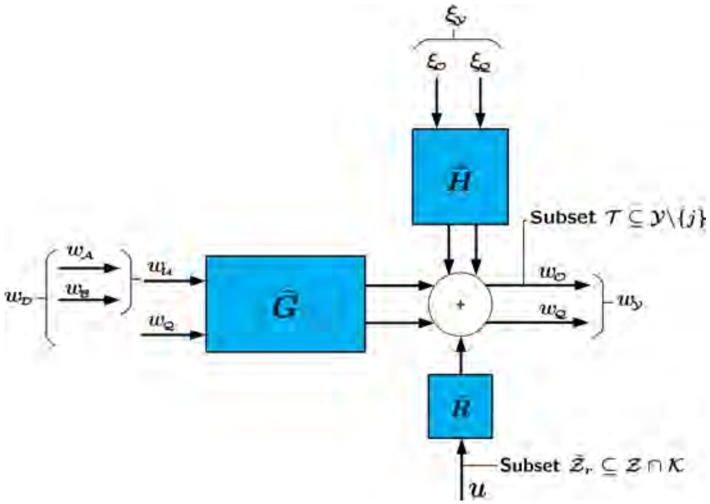


Figure 4.5: Figure to depict the identification setup and classification of different sets of signals in the input and output of the identification problem.

The combination of steps 3 and 4 will lead to a consistent estimation of the target module $G^0_{ji}(q)$.

4.5 Main results - Derivations

4.5.1 System representation after immersion (Step 1-2)

First we will show that we can write an expression for the output variables $w_{\mathcal{Y}}$ as in (4.3).

Proposition 4.2 *Consider a dynamic network defined by* (3.1)*, however with* $cov(e) = I$ *and* H^0 *not necessarily monic. Then,*

1. *there exists a representation (4.2) of the measured node signals* w_m*, with* $\bar{H}_m$ *monic, stable and stably invertible, and* ξ_m *a white noise process, and*

2. *for this representation there are no confounding variables for the estimation problem* $w_{\mathcal{U}} \rightarrow w_{\mathcal{Y}}$.

Proof: See appendix. ■

We refer to the network in (4.2) as the *transformed network*. The consequence of Proposition 4.2 is that the output node signals in $w_{\mathcal{Y}}$ can be explicitly written in the form of (4.3), in terms of input node signals $w_{\mathcal{D}}$, excitation signals u and disturbances, without relying on (unmeasured) node signals in $w_{\mathcal{Z}}$. For the representation (4.3), the structure of $\bar{R}$ will be induced by the topological properties of the network. Note that, $\bar{R}u$ does not necessarily include all external excitation signals in the network, because the effect of some of them on $w_{\mathcal{Y}}$ will be incorporated in the term $\bar{G}w_{\mathcal{D}}$. This latter group of external signals will be referred to as $u_{\bar{\mathcal{P}}}$, with $\bar{\mathcal{P}}$ to be defined later on. In order to use (4.3) for identifying $\bar{G}$ it is attractive to further explore the structure of $\bar{R}$, i.e. to determine which elements of $\bar{R}$ are fixed (e.g. 1 or 0) and which terms are dynamic. To this end, we present the Lemma 4.1.

Condition 4.1 *There exist no direct or unmeasured paths from any signals in $w_{\mathcal{O}}$ to any signals in $w_{\mathcal{B}}$.*

Lemma 4.1 *If Condition 3.2 is satisfied with $\mathcal{U} = \mathcal{A} \cup \mathcal{B}$, such that* in the original network *there are no confounding variables for the estimation problems $w_{\mathcal{A}} \to w_{\mathcal{Y}}$ and $w_{\mathcal{A}} \to w_{\mathcal{B}}$, then for the network in* (4.3)*,*

1. *$\bar{R}_{k\ell} = 0$ with $k, \ell \in \mathcal{Q}$ and $k \neq \ell$;*
2. *$\bar{R}_{k\ell} = 0$ with $k \in \mathcal{O}$ and $\ell \in \mathcal{Q}$;*
3. *$\bar{R}_{k\ell} = 0$ with $k \in \mathcal{Y}$ and $\ell \in \mathcal{A}$;*

and if Condition 4.1 is also satisfied, then

4. *$\bar{R}_{kk} = 1, k \in \mathcal{Y}$, if:*
   (a) *all loops through w_k pass through a node in $w_{\mathcal{Q}\cup\mathcal{U}}$;*
   (b) *there exist no direct or unmeasured paths from w_k to any $w_m, m \in \mathcal{B}$ when $k \in \mathcal{Q}$;*
5. *$\bar{R}_{k\ell} = 0, k, \ell \in \mathcal{O}$ and $k \neq \ell$, if:*
   (a) *all paths from w_ℓ to w_k pass through a node in $w_{\mathcal{Q}\cup\mathcal{U}}$;*
6. *$\bar{R}_{k\ell} = 0, k \in \mathcal{Q}, \ell \in \mathcal{O}$, if:*
   (a) *all paths from w_ℓ to w_k pass through a node in $w_{\mathcal{Q}\cup\mathcal{U}}$;*
7. *$\bar{R}_{k\ell} = 0, k \in \mathcal{Y}, \ell \in \mathcal{Z}$, if:*

(a) all paths from w_ℓ to w_k pass through a node in $w_{\mathcal{Q}\cup\mathcal{U}}$; and

(b) there exist no direct or unmeasured paths from w_ℓ to any $w_m, m \in \mathcal{B}$.

Proof: See appendix. ■

As a result of Lemma 4.1, we can decompose (4.3) as,

$$w_{\mathcal{Y}} = \bar{G}w_{\mathcal{D}} + \bar{H}\xi_{\mathcal{Y}} + \bar{S}u_{\mathcal{P}} + \bar{J}u_{\mathcal{K}}, \tag{4.4}$$

where

- $u_{\mathcal{P}}$ and $u_{\mathcal{K}}$ indicates the excitation signals in u;
- $\bar{S}$ is a binary (selection) matrix with known elements (either 1 or 0), indicating which output node signals are excited by signals $u_{\mathcal{P}}$;
- $\bar{J}$ is a matrix that contains the dynamics, indicating which output node signals are excited by signals $u_{\mathcal{K}}$.

In the next step, we will specify which external signals in u are in $u_{\mathcal{P}}$, $u_{\mathcal{K}}$, or neither in terms of properties of the original network. We will first specify the external signals in $u_{\mathcal{P}}$. Excitation signals can only appear in $u_{\mathcal{P}}$ if they are directly added to node signals in $w_{\mathcal{Y}}$, so $u_{\mathcal{P}} \subseteq u_{\mathcal{Y}}$, as further specified in Proposition 4.3.

Proposition 4.3 *Consider that Condition 3.2 and 4.1 are satisfied. Let $u_{\mathcal{P}}$ be defined as those u-signals in the original network that are directly added to a node signal w_ℓ with $\ell \in \mathcal{Y}$, such that:*

- *if $w_\ell \in w_{\mathcal{Q}}$:*
  1. *all loops through w_ℓ pass through a node in $w_{\mathcal{Q}\cup\mathcal{U}}$; and*
  2. *there exist no direct or unmeasured paths from w_ℓ to any $w_m, m \in \mathcal{B}$;*
- *if $w_\ell \in w_{\mathcal{O}}$:*
  3. *all paths from w_ℓ to $w_k, k \in \mathcal{Y}$ and loops through w_ℓ pass through a node in $w_{\mathcal{Q}\cup\mathcal{U}}$.*

Then $\bar{S}$ is a selection matrix.

Proof: The result of the proposition is a direct result of Lemma 4.1. ■

In order to specify $u_{\mathcal{K}}$, we will follow a reasoning of exclusion. First we specify the set of excitation signals $u_{\bar{\mathcal{P}}}$, that do not appear in the equation (4.4) and then $u_{\mathcal{K}}$ is defined as the remaining excitation signals that appear in (4.4) with a dynamic term, according to $\mathcal{K} = \mathcal{L}\backslash\{\mathcal{P} \cup \bar{\mathcal{P}}\}$. For this we first define the set $\bar{\mathcal{Z}}$.

Definition 4.1 *$\bar{\mathcal{Z}} \subseteq \mathcal{Z}$ denotes the indices of signals in u (i.e. $u_\ell, \ell \in \bar{\mathcal{Z}}$) that satisfies the following property:*

- *all paths from w_ℓ to $w_k, k \in \mathcal{Y}$ pass through a node in $w_{\mathcal{Q}\cup\mathcal{U}}$;*
- *there exist no direct or unmeasured paths from w_ℓ to any $w_m, m \in \mathcal{B}$.*

Now we specify the signals in $u_\mathcal{K}$.

Proposition 4.4 *Consider that Condition 3.2 and 4.1 are satisfied. Let $u_\mathcal{K}$ be the external excitation signals in the network that has the dynamical effect on $w_\mathcal{Y}$. Then $\mathcal{K} = \mathcal{L}\backslash\{\mathcal{P} \cup \mathcal{A} \cup \bar{\mathcal{Z}}\}$.*

Proof: See appendix. ■

Remark 4.5 *If Condition 3.2 is satisfied, then $u_\mathcal{A} \in u_{\bar{\mathcal{P}}}$; and if Condition 4.1 is also satisfied, then $u_{\bar{\mathcal{Z}}} \in u_{\bar{\mathcal{P}}}$.*

The external signals in $u_\mathcal{K}$ are the external excitation signals in the network that have a dynamical effect on $w_\mathcal{Y}$ (i.e. $\bar{J}u_\mathcal{K}$). We will next discuss the identification results for the setup (4.4).

4.5.2 Identification results (Step 3)

For the system's equation (4.4) we can set up a predictor model with input $(w_\mathcal{D}, u_\mathcal{K})$ and outputs $w_\mathcal{Y}$, for the estimation of $\bar{G}$, $\bar{J}$ and $\bar{H}$. As a result we can set up a predictor model based on a parametrized model set determined by

$$\mathcal{M} := \left\{(\bar{G}(\theta), \bar{H}(\theta), \bar{J}(\theta), \bar{\Lambda}(\theta)), \theta \in \Theta\right\},$$

while the actual data generating system is represented by $\mathcal{S} = (\bar{G}(\theta_o), \bar{H}(\theta_o), \bar{J}(\theta_0), \bar{\Lambda}(\theta_0))$. The corresponding identification problem is defined by considering the one-step-ahead prediction of $w_\mathcal{Y}$, according to

$$\hat{w}_\mathcal{Y}(t|t-1) := \mathbb{E}\{w_\mathcal{Y}(t) \mid w_\mathcal{Y}^{t-1}, w_\mathcal{D}^t, u_\mathcal{P}^t, u_\mathcal{K}^t\}$$

where $w_\mathcal{D}^t, u_\mathcal{P}^t$, $u_\mathcal{K}^t$ denotes the past of $w_\mathcal{D}, u_\mathcal{P}, u_\mathcal{K}$ respectively, i.e. $\{w_\mathcal{D}(k), u_\mathcal{P}(k)$ and $u_\mathcal{K}(k), k \leqslant t\}$. The resulting prediction error becomes:

$$\begin{aligned}\varepsilon(t,\theta) &:= w_\mathcal{Y}(t) - \hat{w}_\mathcal{Y}(t|t-1;\theta) \qquad (4.5)\\ &= \bar{H}(q,\theta)^{-1}\left[w_\mathcal{Y}(t) - \bar{G}(q,\theta)w_\mathcal{D}(t) - \bar{S}u_\mathcal{P}(t) - \bar{J}(q,\theta)u_\mathcal{K}(t)\right],\end{aligned}$$

and the weighted least squares identification criterion

$$\hat{\theta}_N = \arg\min_{\theta} \frac{1}{N} \sum_{t=0}^{N-1} \varepsilon^T(t,\theta) W \varepsilon(t,\theta), \tag{4.6}$$

with W any positive definite weighting matrix. This parameter estimate then leads to an estimated subnetwork $\bar{G}_{\mathcal{YD}}(q,\hat{\theta}_N)$, with the estimated module $\bar{G}_{ji}(q,\hat{\theta}_N)$ as one of its scalar entries.

Theorem 4.1 (Consistency) *Consider a (MIMO) network identification setup with predictor inputs $w_{\mathcal{D}}, u_{\mathcal{P}}, u_{\mathcal{K}}$ and predicted outputs $w_{\mathcal{Y}}$, according to (4.4). Let $\mathcal{F}_n \subseteq \mathcal{U}$ be the set of node signals k for which ξ_k is statically uncorrelated with $\xi_{\mathcal{Y}}$ and let $\mathcal{F} := \mathcal{U}\backslash\mathcal{F}_n$. Then a direct prediction error identification method according to (4.5)-(4.6), applied to a parametrized model set $\mathcal{M}$ will provide consistent estimates of $\bar{G}, \bar{J}$ and $\bar{H}$ if:*

a. $\mathcal{M}$ is chosen to satisfy $\mathcal{S} \in \mathcal{M}$;

b. $\Phi_{\kappa}(\omega) > 0$ for a sufficiently high number of frequencies, where $\kappa(t) := \begin{bmatrix} w_{\mathcal{D}}^\top(t) & \xi_{\mathcal{Y}}^\top(t) & u_{\mathcal{K}}(t) \end{bmatrix}^\top$;
(data-informativity condition).

c. The following paths/loops should have at least a delay:

- *All paths/loops from $w_{\mathcal{Y}\cup\mathcal{F}}$ to $w_{\mathcal{Y}}$ in the network (3.4) and in its parametrized model; and*
- *For every $w_k \in \mathcal{F}_n$, all paths from $w_{\mathcal{Y}\cup\mathcal{F}}$ to w_k in the network (3.4), or all paths from w_k to $w_{\mathcal{Y}}$ in the parametrized model.*

(delay in path/loop condition.)

d. All signals in u are uncorrelated to $\xi_{\mathcal{Y}}$.

Proof: The proof is provided in the appendix. ■

It is important to note that Theorem 4.1 is formulated in terms of conditions on the network in (4.3), which we refer to as the *transformed network*. However, it is essential to formulate the conditions in terms of properties of signals in the *original network*, represented by (3.1). This can be done using the result of Proposition 4.5.

Proposition 4.5 *If in the original network, $\mathcal{U}$ is decomposed in two disjoint sets $\mathcal{A}$ and $\mathcal{B}$ satisfying Condition 3.2, then Condition c of Theorem 4.1 can be reformulated as:*

c. The following paths/loops should have at least a delay:

- *All paths/loops from $w_{\mathcal{Y}\cup\mathcal{B}}$ to $w_{\mathcal{Y}}$ in the original network and in the* *parametrized model; and*

- *For every $w_k \in \mathcal{A}$, all paths from $w_{\mathcal{Y}\cup\mathcal{B}}$ to w_k in the original network, or all paths from w_k to $w_\mathcal{Y}$ in the parametrized model.*

Proof: See appendix. ■

4.6 Post-processing (Step 4)

The estimate of $\bar{G}$ contains the estimate of $\bar{G}_{ji}$ as one of its elements since $w_i \in w_\mathcal{D}$ and $w_j \in w_\mathcal{Y}$. However our final goal is to estimate our target module G^0_{ji}. As addressed in the previous chapter, under certain conditions (module invariance conditions), the module $\bar{G}^0_{ji} = G^0_{ji}$, i.e. the module between w_i and w_j in network (4.4) is the target module G^0_{ji} of the original network. Therefore, no post-processing is required in this case. We will first consider these conditions under which no post-processing is required. However, the work in this chapter explores the situation when the module invariance result is not satisfied. We will also explore post-processing of the estimates of the identification to get back the target module. This will be discussed later in this chapter. First, we provide the module invariance result.

Theorem 4.2 (Module invariance result) *Let G_{ji} be the target network module. In the transformed system's equation (3.4) (also* (4.4)*), it holds that $\bar{G}_{ji} = G^0_{ji}$ under the following conditions:*

1. *The parallel path and loop Condition 3.1 is satisfied, and*
2. *The following three conditions are satisfied:*
   a. *$\mathcal{U}$ is decomposed in $\mathcal{A}$ and $\mathcal{B}$, satisfying Condition 3.2, and*
   b. *$i \in \{\mathcal{A} \cup \mathcal{Q}\}$, and*
   c. *Every path from $\{w_i, w_j, w_{\mathcal{O}\setminus\{j\}}\}$ to $w_\mathcal{B}$ passes through a measured node in $w_{\mathcal{L}\setminus\mathcal{Z}}$.*

Proof: See appendix.

Remark 4.6 *In retrospect, the above module invariance result is the generalized result of the module invariance result (Theorem 3.1) from the previous chapter. In the previous chapter, we considered the set $\mathcal{O}$ to be void or with only one element $\{j\}$, however, in this chapter we allow more elements in set $\mathcal{O}$. This is reflected in condition 2(c) of Theorem 3.1.*

The above module invariance condition involves the parallel path and loop condition to be satisfied. In this work, we consider the situation when the parallel path and loop condition cannot be satisfied, i.e. the module invariance result does not hold. In this situation, the target module needs to be extracted from $\bar{G}_{ji}$ through *post-processing*. For this post-processing step we will require two additional sets:

- A set $\mathcal{Z}_r \subseteq \mathcal{Z} \cap \mathcal{K}$ which represents externally excited nodes (i.e. with signals in u) in unmeasured paths from w_i to w_j and in loops around w_j; and
- A set $\mathcal{T} \subseteq \mathcal{Y}\backslash\{j\}$ which, for each of the nodes in $w_{\mathcal{T}}$ represents a measured descendant node that has an unmeasured path from $w_{\mathcal{Z}_r}$, while w_j is excluded from $\mathcal{T}$. Note that for each node $w_k, k \in \mathcal{Z}_r$, the corresponding element in $\mathcal{T}$ is a measured node, and therefore cannot be in the corresponding unmeasured path from w_i to w_j or loops around w_j, that passes through w_k. Therefore the descendant in $\mathcal{T}$ typically breaks out of these unmeasured parallel paths/loops, as illustrated in Figure 4.6.

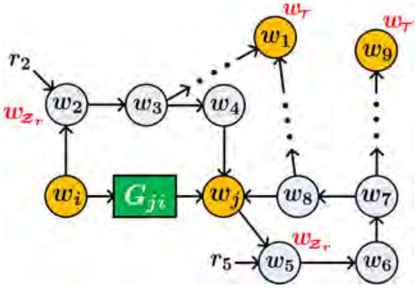


Figure 4.6: Example network with all measured nodes in yellow and $u_2 = r_2, u_5 = r_5$. Modules and noise are not shown for convenience. Arrows with dots indicate unmeasured path.

These two sets will play a major role in extracting the target module estimate from the identification result $\bar{G}, \bar{J}$. The properties that $\mathcal{Z}_r$ and $\mathcal{T}$ need to satisfy in order to realize this post-processing step are formulated next.

For this we formulate the following conditions.

Condition 4.2 In the original network *there are no direct or unmeasured paths from* $w_k, k \in \mathcal{O}\backslash\{j\}$ *to* $(w_j, w_{\mathcal{B}})$*. Also,* $i \in \mathcal{Q} \cup \mathcal{A}$*.*

Property 4.1 (Properties of $\mathcal{Z}_r$ and $\mathcal{T}$) *Let* $\mathcal{Z}_r$ *and* $\mathcal{T}$ *satisfy the following properties:*

1. *All unmeasured paths from* w_i *to* w_j *pass through a node* $w_k, k \in \mathcal{Z}_r$ *that has an unmeasured path to a node* $w_\ell, \ell \in \mathcal{T}$*;*
2. *All unmeasured paths from* w_i *to* $w_{\mathcal{T}}$ *pass through a node* $w_k, k \in \mathcal{Z}_r$

and $G_{\mathcal{T}i} = 0$;

3. *If $i \in \mathcal{T}$, then w_i is excited by an external excitation signal u_i;*
4. *If there exist unmeasured loops through w_j and w_j is not excited by an external excitation signal u_j, then:*
   (a) *The unmeasured loops through w_j pass through a node $w_k, k \in \mathcal{Z}_r$ that has an unmeasured path to a node $w_n, n \in \mathcal{T}$;*
   (b) *All unmeasured paths from w_j to $w_{\mathcal{T}}$ pass through a node $w_k, k \in \mathcal{Z}_r$ and $G_{\mathcal{T}j} = 0$;*
5. *Every $w_k, k \in \mathcal{Z}_r$ is excited by an external excitation signal u_k;*
6. *There exists $|\mathcal{Z}_r|$ vertex-disjoint paths from nodes in $w_{\mathcal{Z}_r}$ to $w_{\mathcal{T}}$;*
7. *All unmeasured paths from w_i, w_j to $w_{\mathcal{B}}$ pass through a node $w_k, k \in \mathcal{Z}_r$ and $G_{\mathcal{B}j} = G_{\mathcal{B}i} = 0$;*
8. *All unmeasured paths from w_i, w_j to $w_{\mathcal{O}}$ pass through a node $w_k, k \in \mathcal{Z}_r$ and $G_{\mathcal{O}j} = G_{\mathcal{O}i} = 0$;*

Theorem 4.3 *Consider the situation of Theorem 4.1, and let Condition 3.2 and Condition 4.2 be satisfied. Let $i \in \mathcal{D}$ and let the sets $\mathcal{Z}_r$ and $\mathcal{T}$ satisfy Property 4.1. Then a consistent estimate of target module G_{ji} is obtained as*

$$G_{ji}(\hat{\theta}_N) = C_{jj}^{-1}(\hat{\theta}_N)\Big(\bar{G}_{ji}(\hat{\theta}_N) - \bar{J}_{j\mathcal{Z}_r}(\hat{\theta}_N) J^{\dagger}_{\mathcal{T}\mathcal{Z}_r}(\hat{\theta}_N) C_{\mathcal{T}i}(\hat{\theta}_N)\Big)$$

where[a]

1. *$C_{jj} = \bar{J}_{jj}$ if w_j is excited by an external signal u_j;*
2. *$C_{jj} = \Big(1 - \bar{J}_{j\mathcal{Z}_r}\bar{J}^{\dagger}_{\mathcal{T}\mathcal{Z}_r}\bar{G}_{\mathcal{T}j}(1 + \bar{J}_{j\mathcal{Z}_r}\bar{J}^{\dagger}_{\mathcal{T}\mathcal{Z}_r}\bar{G}_{\mathcal{T}j})^{-1}\Big)^{-1}$ if w_j is not excited by an external signal u_j;*
3. *$C_{\mathcal{T}i} = \bar{G}_{\mathcal{T}i}$ if $i \notin \mathcal{T}$;*
4. *$C_{\mathcal{T}i} = (\bar{G}_{\mathcal{T}i} + C_{ii})$ if $i \in \mathcal{T}$, where C_{ii} is a column matrix with every element zero except the element corresponding to node w_i which is $\bar{J}_{ii}(1 - \bar{J}_{ii}^{-1})$* □

[a] notation $(\hat{\theta}_N)$ is dropped in the following expressions.

Proof: The proof is provided in the appendix. □

Here $[\ .\ ]^{\dagger}$ correspond to the left inverse of the matrix. The left inverse exists if set $\mathcal{Z}_r$ and $\mathcal{T}$ satisfy Property 4.1.

We interpret Property 4.1 using the network in Figure 4.6. We have one unmeasured parallel path from w_i to w_j and one unmeasured loop through w_j. Considering the parallel path, the excited node w_2 and its measured descendant w_1 ensures that (1)-(2) in Property 4.1 are satisfied with w_2 in $w_{\mathcal{Z}_r}$ and w_1 in $w_{\mathcal{T}}$. Similarly, considering the unmeasured loop through w_j, the excited node w_5 and its measured descendant w_1 ensures that (4) in Property 4.1 is satisfied with w_5 in $w_{\mathcal{Z}_r}$. Property (5) in Property 4.1 is satisfied with both w_2 and w_5 being excited by external signals. However, (6) in Property 4.1 is not satisfied if $w_{\mathcal{Z}_r} = \{w_2, w_5\}$ and $w_{\mathcal{T}} = w_1$. We need 2 vertex-disjoint paths from $w_{\mathcal{Z}_r}$ to $w_{\mathcal{T}}$. Since, we have only w_1 in $w_{\mathcal{T}}$, this property cannot be satisfied. Hence we choose w_9 in $w_{\mathcal{T}}$, which is a descendant of w_5 and ensure that (6) in Property 4.1 is satisfied. Property (3) in Property 4.1 is redundant for this case since $i \notin \mathcal{T}$. It is important to note that $w_{\mathcal{T}}$ can be any node in the network that satisfies the Property 4.1 and thus relaxes the sensor placement scheme. (7) and (8) in Property 4.1 can be similarly interpreted as (2), even though we did not explain it using the figure.

We have now provided the general theory for signal selection and consistency results for the generalized method. The theory for signal selection can lead to multiple predictor models that can provide consistent estimates, similar to the local direct method in the previous chapter. Therefore, we can come up with many algorithms for signal selection that satisfy the general theory. In the sequel, we provide an algorithm for signal selection, which is similar to the minimum measurement case algorithm that has been discussed in the previous chapter, but now applied to a generalized situation.

> **Remark 4.7** *We consider that the input and output of the target module to be measured. However, this requirement is not strict. It is possible to further generalize to situations where either the input or output of the target module is not measured or both the input and output are not measured.*

4.7 Algorithm for signal selection

In this section, we provide the algorithm for signal selection that provides the required identification setup for the introduced identification method in this chapter. This algorithm ensures that Conditions 3.2, 4.1 and 4.2 are satisfied, and $\mathcal{Z}_r$ and $\mathcal{T}$ are chosen according to Property 4.1. Also, the algorithm ensures that $u_{\mathcal{P}}$ and $u_{\mathcal{K}}$ are appropriately chosen as discussed in Section 4.5.1.

Algorithm A

1. Select target module G_{ji};

2. Include j in the index set $\mathcal{Y}$ of measured node variables that are to be predicted.

3. Let $\mathcal{D}_j$ be a set of measured nodes that includes w_i and (some)nodes that block the parallel paths from w_i to w_j and loops through w_j. Then

- include $\mathcal{D}_j$ in $\mathcal{D}$;

4. Select a set $\mathcal{Z}_r$, $\mathcal{T}$ that satisfies Property 4.1 and include it in $\mathcal{K}$ and $\mathcal{Y}$ respectively;

5. Determine $\mathcal{Z} = \mathcal{L}\backslash\{\mathcal{D} \cup \mathcal{Y}\}$;

6. For every $k \in \mathcal{D}$:

   (a) if there exist a confounding variable e_ℓ in e for the estimation problem $w_k \to w_{\mathcal{Y}}$, then include k in $\mathcal{Y}$;

7. If $\mathcal{Y}$ has changed, return to step 6;

8. For every $k \in \mathcal{Y}\backslash\{\mathcal{D} \cup j\}$:

   (a) if there exists an unmeasured path from w_k to a node in w_j, include k in $\mathcal{D}$;

9. Determine $\mathcal{Q} = \mathcal{Y} \cap \mathcal{D}$;

10. Determine $\mathcal{U} = \mathcal{D}\backslash\mathcal{Q}$ and $\mathcal{O} = \mathcal{Y}\backslash\mathcal{Q}$;

11. For every $\ell \in \mathcal{L}\backslash\mathcal{U}$:

    (a) **For** $\ell \in \mathcal{Q}$**:** if all loops through w_ℓ pass through a node in $w_{\mathcal{Q}\cup\mathcal{U}}$, then include ℓ in $\mathcal{P}$ else include in $\mathcal{K}$;

    (b) **For** $\ell \in \mathcal{O}$**:** if all paths from w_ℓ to $w_k, k \in \mathcal{Y}$ and loops through w_ℓ pass through a node in $w_{\mathcal{Q}\cup\mathcal{U}}$, then include ℓ in $\mathcal{P}$ else include in $\mathcal{K}$;

    (c) **For** $\ell \in \mathcal{Z}$**:** if any path from w_ℓ to $w_k, k \in \mathcal{Y}$ does not pass through a node in $w_{\mathcal{Q}\cup\mathcal{U}}$, then include ℓ in $\mathcal{K}$.

When this algorithm finishes, we acquire sets $\mathcal{Y}, \mathcal{D}, \mathcal{P}, \mathcal{K}, \mathcal{T}, \mathcal{Z}_r$ such that we can write the system's output equation as (4.4), the properties of $\mathcal{Z}_r, \mathcal{T}$ are satisfied and we can estimate the target module consistently using the result of Theorem 4.3.

Remark 4.8 *In this chapter, we considered the situation that the input w_i and output w_j of the target module are in the predictor model and thus measured. However, extensions can be made to consider the situations where the input w_i or the output w_j or both not included in the predictor model.*

Remark 4.9 *The method introduced in this chapter, which is referred to as the* generalized method *uses the ingredients of both the direct and indirect method. When the module invariance result in Theorem 4.2 is satisfied, then the* generalized method *simplifies to the* local direct method *(since no post-processing is required to extract the target module) for a dynamic network* *setup with measured external excitation signals u.*

Remark 4.10 *Note that in order to satisfy the data informativity condition and achieve consistency, additional excitation signals might be required to be present.*

4.8 Conclusions

A new local module identification method has been introduced that consistently identifies the target module under known topology, with a generalized scheme for selection of measured node signals and excitation of nodes. We provide flexibility in the sufficient conditions to identify a target module which creates considerable freedom in sensor selection and actuation schemes. This is achieved by combining elements of the direct and indirect identification approaches. We use both node signals and external excitation signals as predictor inputs, allow post-processing of module estimates, and use a MIMO identification setting, thereby mixing both the approaches. With this step we remove restrictive conditions on measured node signals and excitation signals that are present in the currently available methods, e.g. concerning parallel paths and loops around the output.

Appendices

4.A Proof of Proposition 4.2

First we present the following preparatory lemma.

Lemma 4.2 *Let $\mathcal{C} \subset \mathcal{L}$ with cardinality c and $\mathcal{V} = \mathcal{L} \backslash \mathcal{C}$. The system equations for node variables $w_{\mathcal{C}}$ can be written as,*

$$w_{\mathcal{C}} = \breve{G}_{\mathcal{CC}} w_{\mathcal{C}} + \tilde{H} \tilde{\xi}_{\mathcal{C}} + \breve{R} u \,, \tag{4.7}$$

where $\tilde{H}$ is stable, monic and minimum phase rational $c \times c$ matrix, and $\tilde{\xi}_c$ is a $c \times 1$ white noise process with cov($\tilde{\xi}_{\mathcal{C}}$) = $\tilde{\Lambda} \in \mathbb{R}^{c \times c}$, and

$$\begin{aligned} \breve{G}_{\mathcal{CC}} &= G_{\mathcal{CC}} + G_{\mathcal{CV}}(I - G_{\mathcal{VV}})^{-1} G_{\mathcal{VC}} \,, && (4.8) \\ \breve{R}_{\mathcal{CV}} &= G_{\mathcal{CV}}(I - G_{\mathcal{VV}})^{-1} && (4.9) \\ \breve{R} &= \begin{bmatrix} I & \breve{R}_{\mathcal{CV}} \end{bmatrix} . && (4.10) \end{aligned}$$

Proof: On the basis of the decomposition of node signals as defined in the lemma, we are going to rewrite the system's equations (3.1) in the following structured form:

$$\begin{bmatrix} w_{\mathcal{C}} \\ w_{\mathcal{V}} \end{bmatrix} = \begin{bmatrix} G_{\mathcal{CC}} & G_{\mathcal{CV}} \\ G_{\mathcal{VC}} & G_{\mathcal{VV}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{C}} \\ w_{\mathcal{V}} \end{bmatrix} + \begin{bmatrix} H_{\mathcal{CC}} & H_{\mathcal{CV}} \\ H_{\mathcal{VC}} & H_{\mathcal{VV}} \end{bmatrix} \begin{bmatrix} e_{\mathcal{C}} \\ e_{\mathcal{V}} \end{bmatrix} + \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} u_{\mathcal{C}} \\ u_{\mathcal{V}} \end{bmatrix} . \tag{4.11}$$

where $u_{\mathcal{C}} = R_{\mathcal{C}} r$ and $u_{\mathcal{V}} = R_{\mathcal{V}} r$. We can eliminate the node variables $w_{\mathcal{V}}$ from the equation, by writing the last (block) row of (4.58) into an explicit expression for $w_{\mathcal{V}}$:

$$w_{\mathcal{V}} = (I - G_{\mathcal{VV}})^{-1} \left[G_{\mathcal{VC}} w_{\mathcal{C}} + H_{\mathcal{VV}} e_{\mathcal{V}} + H_{\mathcal{VC}} e_{\mathcal{C}} + u_{\mathcal{V}} \right] ,$$

and by substituting this $w_{\mathcal{V}}$ into the expressions for $w_{\mathcal{C}}$-variable. As a result, we have

$$w_{\mathcal{C}} = \breve{G}_{\mathcal{CC}} w_{\mathcal{C}} + \breve{H}_{\mathcal{CC}} e_{\mathcal{C}} + \breve{H}_{\mathcal{CV}} e_{\mathcal{V}} + u_{\mathcal{C}} + \breve{R}_{\mathcal{CV}} u_{\mathcal{V}} \,, \tag{4.12}$$

where $\breve{G}_{\mathcal{CC}}$ and $\breve{R}_{\mathcal{CV}}$ are given by (4.54) and (4.61) respectively, and

$$\breve{H}_{\mathcal{CC}} = H_{\mathcal{CC}} + G_{\mathcal{CV}}(I - G_{\mathcal{VV}})^{-1} H_{\mathcal{VC}} \,, \tag{4.13}$$

$$\breve{H}_{\mathcal{CV}} \quad = \quad H_{\mathcal{CV}} + G_{\mathcal{CV}}(I - G_{\mathcal{VV}})^{-1} H_{\mathcal{VV}} \,. \tag{4.14}$$

From (4.12), let $\breve{v} = \breve{H}e = \begin{bmatrix}\breve{H}_{\mathcal{CC}} & \breve{H}_{\mathcal{CV}}\end{bmatrix} \begin{bmatrix}e_{\mathcal{C}}^{\top} & e_{\mathcal{V}}^{\top}\end{bmatrix}^{\top}$. The spectral density of $\breve{v}$ is given by $\Phi_{\breve{v}} = \breve{H}\breve{H}^{*}$. Applying a spectral factorization [150] to $\Phi_{\breve{v}}$ will deliver $\Phi_{\breve{v}} = \tilde{H}\tilde{\Lambda}\tilde{H}^{*}$ with $\tilde{H}$ a monic, stable and minimum phase rational matrix, and $\tilde{\Lambda}$ a positive definite (constant) matrix. Then there exists a white noise process $\tilde{\xi}_c$ defined by $\tilde{\xi}_c := \tilde{H}^{-1}\breve{H}e$ such that $\tilde{H}\tilde{\xi}_c = \breve{v}$, with $\mathrm{cov}(\tilde{\xi}) = \tilde{\Lambda}$. Thus we get the result of the lemma. ■

Having presented the preparatory lemmas and corollaries, we now provide the proof of proposition 4.2. Considering $\mathcal{C} = \mathcal{D} \cup \mathcal{Y} = \mathcal{Q} \cup \mathcal{O} \cup \mathcal{U}$ and $\mathcal{V} = \mathcal{Z}$, using the result of Lemma 4.2 we can write

$$\begin{aligned}
\begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{O}} \\ w_{\mathcal{U}} \end{bmatrix} \quad = \quad & \begin{bmatrix} \breve{G}_{\mathcal{QQ}} & \breve{G}_{\mathcal{QO}} & \breve{G}_{\mathcal{QU}} \\ \breve{G}_{\mathcal{OQ}} & \breve{G}_{\mathcal{OO}} & \breve{G}_{\mathcal{OU}} \\ \breve{G}_{\mathcal{UQ}} & \breve{G}_{\mathcal{UO}} & \breve{G}_{\mathcal{UU}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{O}} \\ w_{\mathcal{U}} \end{bmatrix} + \begin{bmatrix} \tilde{H}_{\mathcal{QQ}} & \tilde{H}_{\mathcal{QO}} & \tilde{H}_{\mathcal{QU}} \\ \tilde{H}_{\mathcal{OQ}} & \tilde{H}_{\mathcal{OO}} & \tilde{H}_{\mathcal{OU}} \\ \tilde{H}_{\mathcal{UQ}} & \tilde{H}_{\mathcal{UO}} & \tilde{H}_{\mathcal{UU}} \end{bmatrix} \begin{bmatrix} \tilde{\xi}_{\mathcal{Q}} \\ \tilde{\xi}_{\mathcal{O}} \\ \tilde{\xi}_{\mathcal{U}} \end{bmatrix} \\
+ \quad & \begin{bmatrix} I & 0 & 0 & \breve{R}_{\mathcal{QZ}} \\ 0 & I & 0 & \breve{R}_{\mathcal{OZ}} \\ 0 & 0 & I & \breve{R}_{\mathcal{UZ}} \end{bmatrix} \begin{bmatrix} u_{\mathcal{Q}} \\ u_{\mathcal{O}} \\ u_{\mathcal{U}} \\ u_{\mathcal{Z}} \end{bmatrix} .
\end{aligned} \tag{4.15}$$

By denoting $\grave{H}_{\mathcal{QU}} = \tilde{H}_{\mathcal{QU}}\tilde{H}_{\mathcal{UU}}^{-1}$ and $\grave{H}_{\mathcal{OU}} = \tilde{H}_{\mathcal{OU}}\tilde{H}_{\mathcal{UU}}^{-1}$ and Premultiplying (4.15) with $\begin{bmatrix} I & 0 & -\grave{H}_{\mathcal{QU}} \\ 0 & I & -\grave{H}_{\mathcal{OU}} \\ 0 & 0 & I \end{bmatrix}$ while only keeping the identity terms on the left hand side, we obtain an equivalent network equation:

$$\begin{aligned}
\begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{O}} \\ w_{\mathcal{U}} \end{bmatrix} \quad = \quad & \begin{bmatrix} \breve{G}'_{\mathcal{QQ}} & \breve{G}'_{\mathcal{QO}} & \breve{G}'_{\mathcal{QU}} \\ \breve{G}'_{\mathcal{OQ}} & \breve{G}'_{\mathcal{OO}} & \breve{G}'_{\mathcal{OU}} \\ \breve{G}_{\mathcal{UQ}} & \breve{G}_{\mathcal{UO}} & \breve{G}_{\mathcal{UU}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{O}} \\ w_{\mathcal{U}} \end{bmatrix} + \begin{bmatrix} \tilde{H}'_{\mathcal{QQ}} & \tilde{H}'_{\mathcal{QO}} & 0 \\ \tilde{H}'_{\mathcal{OQ}} & \tilde{H}'_{\mathcal{OO}} & 0 \\ \tilde{H}_{\mathcal{UQ}} & \tilde{H}_{\mathcal{UO}} & \tilde{H}_{\mathcal{UU}} \end{bmatrix} \begin{bmatrix} \tilde{\xi}_{\mathcal{Q}} \\ \tilde{\xi}_{\mathcal{O}} \\ \tilde{\xi}_{\mathcal{U}} \end{bmatrix} \\
+ \quad & \begin{bmatrix} I & 0 & \breve{R}'_{\mathcal{QU}} & \breve{R}'_{\mathcal{QZ}} \\ 0 & I & \breve{R}'_{\mathcal{OU}} & \breve{R}'_{\mathcal{OZ}} \\ 0 & 0 & I & \breve{R}_{\mathcal{UZ}} \end{bmatrix} \begin{bmatrix} u_{\mathcal{Q}} \\ u_{\mathcal{O}} \\ u_{\mathcal{U}} \\ u_{\mathcal{Z}} \end{bmatrix} ,
\end{aligned} \tag{4.16}$$

where

$$\breve{G}'_{k\ell} \quad = \quad \breve{G}_{k\ell} - \grave{H}_{k\mathcal{U}}\breve{G}_{\mathcal{U}\ell} \,, \tag{4.17}$$

$$\breve{G}'_{k\mathcal{U}} \quad = \quad \breve{G}_{k\mathcal{U}} - \grave{H}_{k\mathcal{U}}\breve{G}_{\mathcal{UU}} + \grave{H}_{k\mathcal{U}} \,, \tag{4.18}$$

$$\tilde{H}'_{k\ell} \quad = \quad \tilde{H}_{k\ell} - \grave{H}_{k\mathcal{U}}\tilde{H}_{\mathcal{U}\ell} \,, \tag{4.19}$$

$$\breve{R}'_{k\mathcal{U}} \quad = \quad -\grave{H}_{k\mathcal{U}} \,, \tag{4.20}$$

$$\breve{R}'_{kh} \quad = \quad \breve{R}_{kh} - \grave{H}_{k\mathcal{U}}\breve{R}_{\mathcal{U}h} \,, \tag{4.21}$$

with $k, \ell \in \mathcal{Q} \cup \mathcal{O}$ and $h \in \mathcal{Z}$.

The next step is now to show that that the block elements $\breve{G}'_{\mathcal{QO}}$, $\breve{G}'_{\mathcal{OO}}$ in G can be made 0. This can be done by variable substitution as follows:

The second row in (4.15) is replaced by an explicit expression for w_o according to

$$\begin{aligned} w_{\mathcal{O}} &= (I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}[\breve{G}'_{\mathcal{O}\mathcal{Q}} w_{\mathcal{Q}} + \breve{G}'_{\mathcal{O}\mathcal{U}} w_{\mathcal{U}} + \tilde{H}'_{\mathcal{O}\mathcal{Q}} \tilde{\xi}_{\mathcal{Q}} \\ &+ \tilde{H}'_{\mathcal{O}\mathcal{O}} \tilde{\xi}_{\mathcal{O}} + u_{\mathcal{O}} + \breve{R}'_{\mathcal{O}\mathcal{U}} u_{\mathcal{U}} + \breve{R}'_{\mathcal{O}\mathcal{Z}} u_{\mathcal{Z}}]. \end{aligned}$$

Additionally, this expression for w_o is substituted into the first block row of (4.16), to remove the w_o-dependent term on the right hand side, leading to

$$\begin{aligned} \begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{O}} \\ w_{\mathcal{U}} \end{bmatrix} &= \begin{bmatrix} \breve{G}''_{\mathcal{Q}\mathcal{Q}} & 0 & \breve{G}''_{\mathcal{Q}\mathcal{U}} \\ \bar{G}_{\mathcal{O}\mathcal{Q}} & 0 & \bar{G}_{\mathcal{O}\mathcal{U}} \\ \breve{G}_{\mathcal{U}\mathcal{Q}} & \breve{G}_{\mathcal{U}\mathcal{O}} & \breve{G}_{\mathcal{U}\mathcal{U}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{O}} \\ w_{\mathcal{U}} \end{bmatrix} + \begin{bmatrix} \tilde{H}''_{\mathcal{Q}\mathcal{Q}} & \tilde{H}''_{\mathcal{Q}\mathcal{O}} & 0 \\ \tilde{H}''_{\mathcal{O}\mathcal{Q}} & \tilde{H}''_{\mathcal{O}\mathcal{O}} & 0 \\ \tilde{H}_{\mathcal{U}\mathcal{Q}} & \tilde{H}_{\mathcal{U}\mathcal{O}} & \tilde{H}_{\mathcal{U}\mathcal{U}} \end{bmatrix} \begin{bmatrix} \tilde{\xi}_{\mathcal{Q}} \\ \tilde{\xi}_{\mathcal{O}} \\ \tilde{\xi}_{\mathcal{U}} \end{bmatrix} \\ &+ \begin{bmatrix} I & \breve{R}''_{\mathcal{Q}\mathcal{O}} & \breve{R}''_{\mathcal{Q}\mathcal{U}} & \breve{R}''_{\mathcal{Q}\mathcal{Z}} \\ 0 & \bar{R}_{\mathcal{O}\mathcal{O}} & \bar{R}_{\mathcal{O}\mathcal{U}} & \bar{R}_{\mathcal{O}\mathcal{Z}} \\ 0 & 0 & I & \breve{R}_{\mathcal{U}\mathcal{Z}} \end{bmatrix} \begin{bmatrix} u_{\mathcal{Q}} \\ u_{\mathcal{O}} \\ u_{\mathcal{U}} \\ u_{\mathcal{Z}} \end{bmatrix}, \end{aligned} \tag{4.22}$$

with

$$\bar{G}_{\mathcal{O}\star} = (I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1} \breve{G}'_{\mathcal{O}\star} \tag{4.23}$$

$$\tilde{H}''_{\mathcal{O}\star} = (I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1} \tilde{H}'_{\mathcal{O}\star} \tag{4.24}$$

$$\breve{G}''_{\mathcal{Q}\star} = \breve{G}'_{\mathcal{Q}\star} + \breve{G}'_{\mathcal{Q}\mathcal{O}} \bar{G}_{\mathcal{O}\star} \tag{4.25}$$

$$\tilde{H}''_{\mathcal{Q}\star} = \tilde{H}'_{\mathcal{Q}\star} + \breve{G}'_{\mathcal{Q}\mathcal{O}} \tilde{H}''_{\mathcal{O}\star} \tag{4.26}$$

$$\bar{R}_{\mathcal{O}\mathcal{O}} = (I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1} \ ; \ \bar{R}_{\mathcal{O}\star} = (I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1} \breve{R}'_{\mathcal{O}\star} \tag{4.27}$$

$$\breve{R}''_{\mathcal{Q}\mathcal{O}} = \breve{G}'_{\mathcal{Q}\mathcal{O}} \bar{R}_{\mathcal{O}\mathcal{O}} \ ; \ \breve{R}''_{\mathcal{Q}\star} = \breve{R}'_{\mathcal{Q}\star} + \breve{G}'_{\mathcal{Q}\mathcal{O}} \bar{R}_{\mathcal{O}\star}. \tag{4.28}$$

Since because of these operations, the matrix $\breve{G}''_{\mathcal{Q}\mathcal{Q}}$ might not be hollow, we move any diagonal terms of this matrix to the left hand side of the equation, and premultiply the first (block) equation by the diagonal matrix $(I - \operatorname{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1}$, to obtain the expression

$$\begin{aligned} \begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{O}} \\ w_{\mathcal{U}} \end{bmatrix} &= \begin{bmatrix} \bar{G}_{\mathcal{Q}\mathcal{Q}} & 0 & \bar{G}_{\mathcal{Q}\mathcal{U}} \\ \bar{G}_{\mathcal{O}\mathcal{Q}} & 0 & \bar{G}_{\mathcal{O}\mathcal{U}} \\ \breve{G}_{\mathcal{U}\mathcal{Q}} & \breve{G}_{\mathcal{U}\mathcal{O}} & \breve{G}_{\mathcal{U}\mathcal{U}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{O}} \\ w_{\mathcal{U}} \end{bmatrix} + \begin{bmatrix} \tilde{H}'''_{\mathcal{Q}\mathcal{Q}} & \tilde{H}'''_{\mathcal{Q}\mathcal{O}} & 0 \\ \tilde{H}''_{\mathcal{O}\mathcal{Q}} & \tilde{H}''_{\mathcal{O}\mathcal{O}} & 0 \\ \tilde{H}_{\mathcal{U}\mathcal{Q}} & \tilde{H}_{\mathcal{U}\mathcal{O}} & \tilde{H}_{\mathcal{U}\mathcal{U}} \end{bmatrix} \begin{bmatrix} \tilde{\xi}_{\mathcal{Q}} \\ \tilde{\xi}_{\mathcal{O}} \\ \tilde{\xi}_{\mathcal{U}} \end{bmatrix} \\ &+ \underbrace{\begin{bmatrix} \bar{R}_{\mathcal{Q}\mathcal{Q}} & \bar{R}_{\mathcal{Q}\mathcal{O}} & \bar{R}_{\mathcal{Q}\mathcal{U}} & \bar{R}_{\mathcal{Q}\mathcal{Z}} \\ 0 & \bar{R}_{\mathcal{O}\mathcal{O}} & \bar{R}_{\mathcal{O}\mathcal{U}} & \bar{R}_{\mathcal{O}\mathcal{Z}} \\ 0 & 0 & I & \breve{R}_{\mathcal{U}\mathcal{Z}} \end{bmatrix} \begin{bmatrix} u_{\mathcal{Q}} \\ u_{\mathcal{O}} \\ u_{\mathcal{U}} \\ u_{\mathcal{Z}} \end{bmatrix}}_{\bar{u}}, \end{aligned} \tag{4.29}$$

with

$$\bar{G}_{\mathcal{Q}\mathcal{Q}} = (I - \operatorname{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1}(\breve{G}''_{\mathcal{Q}\mathcal{Q}} - \operatorname{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}})), \tag{4.30}$$

$$\bar{G}_{\mathcal{Q}\mathcal{U}} = (I - \operatorname{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1} \breve{G}''_{\mathcal{Q}\mathcal{U}} \tag{4.31}$$

$$\begin{aligned} \tilde{H}'''_{\mathcal{Q}\star} &= (I - \mathrm{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1}\tilde{H}''_{\mathcal{Q}\star} && (4.32)\\ \bar{R}_{\mathcal{Q}\mathcal{Q}} &= (I - \mathrm{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1} && (4.33)\\ \bar{R}_{\mathcal{Q}\star} &= (I - \mathrm{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1}\breve{R}''_{\mathcal{Q}\star}. && (4.34) \end{aligned}$$

As final step, we need the matrix $\tilde{H}_r := \begin{bmatrix} \tilde{H}'''_{\mathcal{Q}\mathcal{Q}} & \tilde{H}'''_{\mathcal{Q}\mathcal{O}} \\ \tilde{H}''_{\mathcal{O}\mathcal{Q}} & \tilde{H}''_{\mathcal{O}\mathcal{O}} \end{bmatrix} = \begin{bmatrix} \check{H}_{\mathcal{Q}\mathcal{Q}} & \check{H}_{\mathcal{Q}\mathcal{O}} \\ \check{H}_{\mathcal{O}\mathcal{Q}} & \check{H}_{\mathcal{O}\mathcal{O}} \end{bmatrix}$ to be monic, stable and minimum phase. To that end, we consider the stochastic process $\tilde{v}_{\mathcal{Y}} := \tilde{H}_r\tilde{\xi}_{\mathcal{Y}}$ with $\tilde{\xi}_{\mathcal{Y}} := \begin{bmatrix}\tilde{\xi}_{\mathcal{Q}}^{\top} & \tilde{\xi}_{\mathcal{O}}^{\top}\end{bmatrix}^{\top}$. The spectral density of $\tilde{v}_{\mathcal{Y}}$ is then given by $\Phi_{\tilde{v}_{\mathcal{Y}}} = \tilde{H}_r\tilde{\Lambda}_{\mathcal{Y}}\tilde{H}_r^{\star}$ with $\tilde{\Lambda}_{\mathcal{Y}}$ the covariance matrix of $\tilde{\xi}_{\mathcal{Y}}$, that can be decomposed as $\tilde{\Lambda}_{\mathcal{Y}} = \tilde{\Gamma}_r\tilde{\Gamma}_r^T$. From spectral factorization [150] it follows that the spectral factor $\tilde{H}_r\tilde{\Gamma}_r$ of $\Phi_{\tilde{v}_{\mathcal{Y}}}$ satisfies

$$\tilde{H}_r\tilde{\Gamma}_r = \bar{H}_s D \tag{4.35}$$

with $\bar{H}_s$ a stable and minimum phase rational matrix, and D an "all pass" stable rational matrix satisfying $DD^{\star} = I$.
The signal $\tilde{v}_{\mathcal{Y}}$ can then be written as

$$\tilde{v}_{\mathcal{Y}} = \tilde{H}_r\tilde{\xi}_{\mathcal{Y}} = \bar{H}_s D\tilde{\Gamma}_r^{-1}\tilde{\xi}_{\mathcal{Y}}.$$

By defining $\bar{H}_s^{\infty} := \lim_{z\to\infty}\bar{H}_s$, this can be rewritten as

$$\tilde{v}_{\mathcal{Y}} = \tilde{H}_r\tilde{\xi}_{\mathcal{Y}} = \underbrace{\bar{H}_s(\bar{H}_s^{\infty})^{-1}}_{\bar{H}}\underbrace{\bar{H}_s^{\infty}D\tilde{\Gamma}_r^{-1}\tilde{\xi}_{\mathcal{Y}}}_{\xi_{\mathcal{Y}}}.$$

As a result, $\bar{H}$ is a monic stable and stably invertible rational matrix, and $\xi_{\mathcal{Y}}$ is a white noise process with spectral density given by $\bar{H}_s^{\infty}D\tilde{\Gamma}_r^{-1}\Phi_{\tilde{\xi}_{\mathcal{Y}}}\tilde{\Gamma}_r^{-T}D^{\star}(\bar{H}_s^{\infty})^T = \bar{H}_s^{\infty}(\bar{H}_s^{\infty})^T = \bar{\Sigma}$. Therefore we can write (4.29) as,

$$\underbrace{\begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{O}} \\ w_{\mathcal{U}} \end{bmatrix}}_{w_m} = \begin{bmatrix} \bar{G}_{\mathcal{Q}\mathcal{Q}} & 0 & \bar{G}_{\mathcal{Q}\mathcal{U}} \\ \bar{G}_{\mathcal{O}\mathcal{Q}} & 0 & \bar{G}_{\mathcal{O}\mathcal{U}} \\ \breve{G}_{\mathcal{U}\mathcal{Q}} & \breve{G}_{\mathcal{U}o} & \breve{G}_{\mathcal{U}\mathcal{U}} \end{bmatrix}\begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{O}} \\ w_{\mathcal{U}} \end{bmatrix} + \begin{bmatrix} \bar{H}_{\mathcal{Q}\mathcal{Q}} & \bar{H}_{\mathcal{Q}\mathcal{O}} & 0 \\ \bar{H}_{\mathcal{O}\mathcal{Q}} & \bar{H}_{\mathcal{O}\mathcal{O}} & 0 \\ \bar{H}_{\mathcal{U}\mathcal{Q}} & \bar{H}_{\mathcal{U}\mathcal{O}} & \tilde{H}_{\mathcal{U}\mathcal{U}} \end{bmatrix}\underbrace{\begin{bmatrix} \xi_{\mathcal{Q}} \\ \xi_{\mathcal{O}} \\ \tilde{\xi}_{\mathcal{U}} \end{bmatrix}}_{\xi_m} + \bar{u}, \tag{4.36}$$

where $\begin{bmatrix}\bar{H}_{\mathcal{U}\mathcal{Q}} & \bar{H}_{\mathcal{U}\mathcal{O}}\end{bmatrix} = \begin{bmatrix}\tilde{H}_{\mathcal{U}\mathcal{Q}} & \tilde{H}_{\mathcal{U}\mathcal{O}}\end{bmatrix}\tilde{\Gamma}_r D^{-1}(\bar{H}_s^{\infty})^{-1}$.

With the first two (block) rows representing the equation for $w_{\mathcal{Y}}$, we get the result of the proposition. ■

4.B Proof of Lemma 4.1

First we present the following preparatory lemmas and corollaries

Lemma 4.3 *Consider the situation in Lemma 4.2. Let $\mathcal{C}_1 \subseteq \mathcal{C}$ and $\mathcal{C}_2 = \mathcal{C}\backslash\mathcal{C}_1$ belong to sets of measured node signals in w and set $\mathcal{V}$ belong to unmeasured* *node signals in w. If there are no confounding variables for the estimation problems $w_{\mathcal{C}_1} \rightarrow w_{\mathcal{C}_2}$, then the spectral density $\Phi_{\breve{v}}$ has the unique spectral factorization $\Phi_{\breve{v}} = \tilde{H}\tilde{\Lambda}\tilde{H}^*$ with $\tilde{\Lambda}$ constant and $\tilde{H}$ monic, stable, minimum phase, and of the form*

$$\Lambda = \begin{bmatrix} \Lambda_{11} & 0 \\ 0 & \Lambda_{22} \end{bmatrix}, \; \tilde{H} = \begin{bmatrix} \tilde{H}_{\mathcal{C}_1\mathcal{C}_1} & 0 \\ 0 & \tilde{H}_{\mathcal{C}_2\mathcal{C}_2} \end{bmatrix}, \tag{4.37}$$

where the block dimensions are conformable to the dimensions of $w_{\mathcal{C}_1}$ and $w_{\mathcal{C}_2}$ respectively.

Proof: On the basis of (4.63), we can write

$$\breve{v} = \underbrace{\begin{bmatrix} \breve{H}_{\mathcal{C}_1\mathcal{C}_1} & \breve{H}_{\mathcal{C}_1\mathcal{C}_2} & \breve{H}_{\mathcal{C}_1\mathcal{V}} \\ \breve{H}_{\mathcal{C}_2\mathcal{C}_1} & \breve{H}_{\mathcal{C}_2\mathcal{C}_2} & \breve{H}_{\mathcal{C}_2\mathcal{V}} \end{bmatrix}}_{\breve{H}} \begin{bmatrix} e_{\mathcal{C}_1} \\ e_{\mathcal{C}_2} \\ e_{\mathcal{V}} \end{bmatrix}. \tag{4.38}$$

Starting from the expression (4.38), the spectral density $\Phi_{\breve{v}}$ can be written as $\breve{H}\breve{H}^*$ while it is denoted as

$$\Phi_{\breve{v}} = \begin{bmatrix} \Phi_{w_{\mathcal{C}_1}} & \Phi_{\breve{w}_{\mathcal{C}_1}\breve{w}_{\mathcal{C}_2}} \\ \Phi^*_{\breve{w}_{\mathcal{C}_1}\breve{w}_{\mathcal{C}_2}} & \Phi_{\breve{w}_{\mathcal{C}_2}} \end{bmatrix}. \tag{4.39}$$

In this structure we are particularly going to analyse the element $\Phi_{\breve{w}_{\mathcal{C}_1}\breve{w}_{\mathcal{C}_2}} = \breve{H}_{\mathcal{C}_1\mathcal{C}_2}\breve{H}^*_{\mathcal{C}_2\mathcal{C}_1} + \breve{H}_{\mathcal{C}_1\mathcal{C}_2}\breve{H}^*_{\mathcal{C}_2\mathcal{C}_2} + \breve{H}_{\mathcal{C}_1\mathcal{V}}\breve{H}^*_{\mathcal{C}_2\mathcal{V}}$. If there are no confounding variables for the estimation problems $w_{\mathcal{C}_1} \rightarrow w_{\mathcal{C}_2}$, none of the white noise terms e_x, $x \in \mathcal{L}$ have simultaneous paths to node signals $w_k, k \in \mathcal{C}_1$ and $w_\ell, \ell \in \mathcal{C}_2$, while these paths are unmeasured paths. Then it follows from Lemma 3.1 that $\Phi_{\breve{w}_{\mathcal{C}_1}\breve{w}_{\mathcal{C}_2}}$ is zero. As a result the spectrum in equation (4.39) will be block diagonal. Then the spectral density $\Phi_{\breve{v}}$ has the unique spectral factorization [150] $\Phi_{\breve{v}} = \tilde{H}\tilde{\Lambda}\tilde{H}^*$, where $\tilde{\Lambda}$ and $\tilde{H}$ is of the form in (4.37) with $\tilde{H}$ being monic, stable and minimum phase. ■

Corollary 4.1 *Consider the situation in Lemma 4.3. Then $\tilde{H}_{\mathcal{C}_1\mathcal{C}_2} = \tilde{H}_{\mathcal{C}_2\mathcal{C}_1} = 0$. If $k \in \mathcal{C}_1$ and $\ell \in \mathcal{C}_2$ or vice versa, then $\tilde{H}_{kl} = \tilde{H}_{lk} = 0$. Also, if $k \in \mathcal{C}_1$, $\tilde{H}_{k\mathcal{C}_2} = 0$.*

Lemma 4.4 *If condition 3.2 is satisfied, then $\grave{H}_{k\ell} = 0$ if $k \in \mathcal{Y}, \ell \in \mathcal{A}$.*

Proof: Using the result in Corollary 4.1, considering $\mathcal{C} = \mathcal{Y} \cup \mathcal{U}$, $\mathcal{C}_1 = \mathcal{Y} \cup \mathcal{B}$ and $\mathcal{C}_2 = \mathcal{A}$, we have $\tilde{H}_{k\mathcal{A}} = \tilde{H}_{\mathcal{A}k} = 0$ and $\tilde{H}_{\mathcal{Y}\mathcal{A}} = \tilde{H}_{\mathcal{A}\mathcal{Y}} = \tilde{H}_{\mathcal{B}\mathcal{A}} = \tilde{H}_{\mathcal{A}\mathcal{B}} = 0$. Let $\grave{H}_{k\mathcal{U}}$ is the row vector corresponding to the row of node signal k in $\grave{H}_{\mathcal{Q}\mathcal{U}}$ (if $k \in \mathcal{Q}$) or in $\grave{H}_{\mathcal{O}\mathcal{U}}$ (if $k \in \mathcal{O}$), and $\grave{H}_{k\ell}$ is the element corresponding to the column of node signal w_ℓ in $\grave{H}_{k\mathcal{U}}$.

From the definition of $\grave{H}_{k\mathcal{U}}$, we can write

$$\grave{H}_{k\mathcal{U}} = \begin{bmatrix} \tilde{H}_{k\mathcal{B}} & 0 \end{bmatrix} \begin{bmatrix} \tilde{H}_{\mathcal{B}\mathcal{B}} & 0 \\ 0 & \tilde{H}_{\mathcal{A}\mathcal{A}} \end{bmatrix}^{-1} = \begin{bmatrix} \grave{H}_{k\mathcal{B}} & 0 \end{bmatrix}, \tag{4.40}$$

implying that columns in this matrix related to inputs $\ell \in \mathcal{A}$ are zero. ■

Having presented the preparatory lemma, we now provide the proof of proposition 4.1. From (4.29) it follows that

$$\bar{u} = \begin{bmatrix} \bar{R}_{\mathcal{Q}\mathcal{Q}} & \bar{R}_{\mathcal{Q}\mathcal{O}} & \bar{R}_{\mathcal{Q}\mathcal{U}} & \bar{R}_{\mathcal{Q}\mathcal{Z}} \\ 0 & \bar{R}_{\mathcal{O}\mathcal{O}} & \bar{R}_{\mathcal{O}\mathcal{U}} & \bar{R}_{\mathcal{O}\mathcal{Z}} \\ 0 & 0 & I & \breve{R}_{\mathcal{U}\mathcal{Z}} \end{bmatrix} \begin{bmatrix} u_{\mathcal{Q}} \\ u_{\mathcal{O}} \\ u_{\mathcal{U}} \\ u_{\mathcal{Z}} \end{bmatrix}. \tag{4.41}$$

Directly from (4.41), we can infer that $\bar{R}_{\mathcal{O}\mathcal{Q}} = 0$.

The term $\bar{R}_{k\ell}$ with $k \in \mathcal{Q}, \ell \in \mathcal{O}$:
First we evaluate the term $\bar{R}_{\mathcal{Q}\mathcal{O}}$. With (4.34) and (4.28) it follows that

$$\bar{R}_{\mathcal{Q}\mathcal{O}} = (I - \text{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1} \breve{G}'_{\mathcal{Q}\mathcal{O}} (I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1},$$

while with (4.17):

$$\breve{G}'_{\mathcal{Q}\mathcal{O}} = \breve{G}_{\mathcal{Q}\mathcal{O}} - \grave{H}_{\mathcal{Q}\mathcal{B}} \breve{G}_{\mathcal{B}\mathcal{O}}, \qquad \breve{G}'_{\mathcal{O}\mathcal{O}} = \breve{G}_{\mathcal{O}\mathcal{O}} - \grave{H}_{\mathcal{O}\mathcal{B}} \breve{G}_{\mathcal{B}\mathcal{O}}.$$

Therefore, with $k \in \mathcal{Q}$ and $\ell \in \mathcal{O}$, we have,

$$\bar{R}_{k\ell} = (I - \breve{G}''_{kk})^{-1} \breve{G}'_{k\mathcal{O}} [(I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}]_{(:,\ell)},$$

where $[\quad]_{(:,\ell)}$ denotes the ℓ^{th} column of the matrix. From Condition 4.1 it follows that $\breve{G}_{\mathcal{B}\mathcal{O}} = 0$, and from Condition 6(a) in Lemma 4.1 it follows that $\breve{G}_{k\mathcal{O}}[(I - \breve{G}_{\mathcal{O}\mathcal{O}})^{-1}]_{(:,\ell)} = 0$. As a result, $\breve{G}'_{k\mathcal{O}}[(I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}]_{(:,\ell)} = 0$ leading to $\bar{R}_{k\ell} = 0$ with $k \in \mathcal{Q}, \ell \in \mathcal{O}$ if Condition 6(a) in Lemma 4.1 and Condition 4.1 are satisfied.

The term $\bar{R}_{k\ell}$ with $k \in \mathcal{O}, \ell \in \mathcal{O}$:
For the term $\bar{R}_{\mathcal{O}\mathcal{O}}$, we have according to (4.27) that $\bar{R}_{\mathcal{O}\mathcal{O}} = (I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}$. From Condition 4.1 it follows that $\breve{G}_{\mathcal{B}\mathcal{O}} = 0$. Therefore, with $k \in \mathcal{O}$ and $\ell \in \mathcal{O}$, we have $R_{k\ell} = [(I - \breve{G}_{\mathcal{O}\mathcal{O}})^{-1}]_{(k,\ell)}$. From condition 5(a) in Lemma 4.1 it follows that $[(I - \breve{G}_{\mathcal{O}\mathcal{O}})^{-1}]_{(k,\ell)} = 0$ when $k \neq \ell$ and from Condition 4(a) in Lemma 4.1 it follows that $[(I - \breve{G}_{\mathcal{O}\mathcal{O}})^{-1}]_{(k,k)} = 1$. As a result, we have $\bar{R}_{k\ell} = 0$ with $k, \ell \in \mathcal{O}$ and $k \neq \ell$ if Condition 5(a) in Lemma 4.1 and Condition 4.1 are satisfied, and $\bar{R}_{kk} = 1$ with $k \in \mathcal{O}$ if Condition 4(a) in Lemma 4.1 and Condition 4.1 are satisfied.

The term $\bar{R}_{k\ell}$ with $k \in \mathcal{Q}, \ell \in \mathcal{Q}$:
Next, for the term $\bar{R}_{\mathcal{Q}\mathcal{Q}}$ we consider according to (4.33), $\bar{R}_{\mathcal{Q}\mathcal{Q}} = (I - \text{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1}$ and from (4.25):

$$\breve{G}''_{\mathcal{Q}\mathcal{Q}} = \breve{G}'_{\mathcal{Q}\mathcal{Q}} + \breve{G}'_{\mathcal{Q}\mathcal{O}} \bar{G}_{\mathcal{O}\mathcal{Q}}.$$

Since $\bar{R}_{\mathcal{Q}\mathcal{Q}}$ is a diagonal matrix, this implies that $\bar{R}_{k\ell} = 0$ if $k, \ell \in \mathcal{Q}$ and $k \neq \ell$. $\bar{R}_{kk}$

for $k \in \mathcal{Q}$ is given by $\bar{R}_{kk} = (I - \breve{G}''_{kk})^{-1}$ and from (4.25):

$$\breve{G}''_{kk} = \breve{G}'_{kk} + \breve{G}'_{k\mathcal{O}}\bar{G}_{\mathcal{O}k},$$

which when substituting (4.17) and (4.23) becomes for $k \in \mathcal{Q}$:

$$\breve{G}''_{kk} \quad = \quad \breve{G}_{kk} - \grave{H}_{k\mathcal{B}}\breve{G}_{\mathcal{B}k} + (\breve{G}_{k\mathcal{O}} - \breve{H}_{k\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}})(I - \breve{G}_{\mathcal{O}\mathcal{O}} + \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}})^{-1}(\breve{G}_{\mathcal{O}k} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}k}).$$

From conditions 4(a), 4(b) in Lemma 4.1 and Condition 4.1 it follows that $\breve{G}_{kk} = 0$, $\breve{G}_{\mathcal{B}\mathcal{O}} = 0$ and $\breve{G}_{\mathcal{B}k} = 0$. Substituting this in the previous equation leads to $\breve{G}''_{kk} = \breve{G}_{k\mathcal{O}}(I - \breve{G}_{\mathcal{O}\mathcal{O}})^{-1}\breve{G}_{\mathcal{O}k}$. Since condition 4(a) in Lemma 4.1 excludes the possibility that there is a loop through $w_k, k \in \mathcal{Q}$ that passes through a node in $w_{\mathcal{O}}$, it follows that $\breve{G}''_{kk} = 0$. Therefore $\bar{R}_{kk} = 1$ for $k \in \mathcal{Q}$ if Conditions 4(a), 4(b) in Lemma 4.1 and Condition 4.1 are satisfied.

The term $\bar{R}_{\mathcal{Y}\mathcal{A}}$:
For the term $\bar{R}_{\mathcal{O}\mathcal{U}}$, we have according to (4.27) that $\bar{R}_{\mathcal{O}\mathcal{U}} = (I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}\breve{R}'_{\mathcal{O}\mathcal{U}}$. From (4.17) and (4.20) we have $\breve{G}'_{\mathcal{O}\mathcal{O}} = \breve{G}_{\mathcal{O}\mathcal{O}} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}}$ and $\breve{R}'_{\mathcal{O}\mathcal{U}} = [-\grave{H}_{\mathcal{O}\mathcal{B}} \quad -\grave{H}_{\mathcal{O}\mathcal{A}}]$ with $\grave{H}_{\mathcal{O}\mathcal{A}} = 0$. This implies that $\bar{R}_{\mathcal{O}\mathcal{A}} = 0$. For the term $\bar{R}_{\mathcal{Q}\mathcal{U}}$, we have according to (4.34) that $\bar{R}_{\mathcal{Q}\mathcal{U}} = (I - \operatorname{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1}\breve{R}''_{\mathcal{Q}\mathcal{U}}$. With (4.28) it follows that

$$\bar{R}_{\mathcal{Q}\mathcal{U}} = (I - \operatorname{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1}(\breve{R}'_{\mathcal{Q}\mathcal{U}} + \breve{G}'_{\mathcal{Q}\mathcal{O}}(I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}\breve{R}'_{\mathcal{O}\mathcal{U}}),$$

while with (4.17):

$$\breve{G}'_{\mathcal{Q}\mathcal{O}} = \breve{G}_{\mathcal{Q}\mathcal{O}} - \grave{H}_{\mathcal{Q}\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}}, \qquad \breve{G}'_{\mathcal{O}\mathcal{O}} = \breve{G}_{\mathcal{O}\mathcal{O}} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}},$$

and with (4.20) we have $\breve{R}'_{\mathcal{O}\mathcal{U}} = [-\grave{H}_{\mathcal{O}\mathcal{B}} \quad -\grave{H}_{\mathcal{O}\mathcal{A}}]$ and $\breve{R}'_{\mathcal{Q}\mathcal{U}} = [-\grave{H}_{\mathcal{Q}\mathcal{B}} \quad -\grave{H}_{\mathcal{Q}\mathcal{A}}]$ with $\grave{H}_{\mathcal{O}\mathcal{A}} = \grave{H}_{\mathcal{Q}\mathcal{A}} = 0$. Therefore, we have $R_{\mathcal{Q}\mathcal{A}} = 0$ and combining with the result of $\bar{R}_{\mathcal{O}\mathcal{A}}$, it implies that $\bar{R}_{\mathcal{Y}\mathcal{A}} = 0$.

The term $\bar{R}_{k\ell}$ with $k \in \mathcal{Y}, \ell \in \mathcal{Z}$:
For the term $\bar{R}_{\mathcal{O}\mathcal{Z}}$, we have according to (4.27) that $\bar{R}_{\mathcal{O}\mathcal{Z}} = (I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}\breve{R}'_{\mathcal{O}\mathcal{Z}}$. From (4.17) and (4.21) we have $\breve{G}'_{\mathcal{O}\mathcal{O}} = \breve{G}_{\mathcal{O}\mathcal{O}} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}}$ and $\breve{R}'_{\mathcal{O}\mathcal{Z}} = \breve{R}_{\mathcal{O}\mathcal{Z}} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{R}_{\mathcal{B}\mathcal{Z}}$. If Condition 4.1 is satisfied, we have $\breve{G}_{\mathcal{B}\mathcal{O}} = 0$. From (4.61), we have that $\breve{R}_{\mathcal{O}\mathcal{Z}} = G_{\mathcal{O}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}$ and $\breve{R}_{\mathcal{B}\mathcal{Z}} = G_{\mathcal{B}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}$. This implies that for $k \in \mathcal{O}$ and $\ell \in \mathcal{Z}$, we have

$$\bar{R}_{k\ell} = [(I - \breve{G}_{\mathcal{O}\mathcal{O}})^{-1}]_{(k,:)}(G_{\mathcal{O}\mathcal{Z}}[(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}]_{(:,\ell)} - \grave{H}_{\mathcal{O}\mathcal{B}}G_{\mathcal{B}\mathcal{Z}}[(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}]_{(:,\ell)}).$$

If condition 7(b) in Lemma 4.1 is satisfied, we have $G_{\mathcal{B}\mathcal{Z}}[(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}]_{(:,\ell)} = 0$ and if condition 7(a) in Lemma 4.1 is satisfied, we have $[(I - \breve{G}_{\mathcal{O}\mathcal{O}})^{-1}]_{(k,:)}G_{\mathcal{O}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}]_{(:,\ell)} = 0$. Therefore, $\bar{R}_{k\ell} = 0$ if conditions 7(a), 7(b) in Lemma 4.1 and Condition 4.1 are satisfied for $k \in \mathcal{O}$ and $\ell \in \mathcal{Z}$.

For the term $\bar{R}_{\mathcal{Q}\mathcal{Z}}$, we have according to (4.34) that $\bar{R}_{\mathcal{Q}\mathcal{Z}} = (I - \operatorname{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1}\breve{R}''_{\mathcal{Q}\mathcal{Z}}$. With (4.28) it follows that

$$\bar{R}_{\mathcal{Q}\mathcal{Z}} = (I - \operatorname{diag}(\breve{G}''_{\mathcal{Q}\mathcal{Q}}))^{-1}(\breve{R}'_{\mathcal{Q}\mathcal{Z}} + \breve{G}'_{\mathcal{Q}\mathcal{O}}(I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}\breve{R}'_{\mathcal{O}\mathcal{Z}}),$$

while with (4.17):

$$\breve{G}'_{\mathcal{Q}\mathcal{O}} = \breve{G}_{\mathcal{Q}\mathcal{O}} - \grave{H}_{\mathcal{Q}\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}}, \qquad \breve{G}'_{\mathcal{O}\mathcal{O}} = \breve{G}_{\mathcal{O}\mathcal{O}} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}},$$

and with (4.21) we have $\breve{R}'_{\mathcal{O}\mathcal{Z}} = \breve{R}_{\mathcal{O}\mathcal{Z}} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{R}_{\mathcal{B}\mathcal{Z}}$ and $\breve{R}'_{\mathcal{Q}\mathcal{Z}} = \breve{R}_{\mathcal{Q}\mathcal{Z}} - \grave{H}_{\mathcal{Q}\mathcal{B}}\breve{R}_{\mathcal{B}\mathcal{Z}}$. If Condition 4.1 is satisfied, we have $\breve{G}_{\mathcal{B}\mathcal{O}} = 0$. From (4.61), we have that $\breve{R}_{\mathcal{O}\mathcal{Z}} = G_{\mathcal{O}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}$, $\breve{R}_{\mathcal{Q}\mathcal{Z}} = G_{\mathcal{Q}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}$ and $\breve{R}_{\mathcal{B}\mathcal{Z}} = G_{\mathcal{B}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}$. This implies that for $k \in \mathcal{Q}$ and $\ell \in \mathcal{Z}$, we have

$$\begin{aligned}\bar{R}_{k\ell} &= (I - \breve{G}_{kk})^{-1}\big(G_{k\mathcal{Z}}[(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}]_{(:,\ell)} - \grave{H}_{k\mathcal{B}}G_{\mathcal{B}\mathcal{Z}}[(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}]_{(:,\ell)} \\ &\quad + \breve{G}_{k\mathcal{O}}(I - \breve{G}_{\mathcal{O}\mathcal{O}})^{-1}(G_{\mathcal{O}\mathcal{Z}}[(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}]_{(:,\ell)} - \grave{H}_{\mathcal{O}\mathcal{B}}G_{\mathcal{B}\mathcal{Z}}[(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}]_{(:,\ell)})\big) \qquad (4.42)\end{aligned}$$

If condition 7(b) in Lemma 4.1 is satisfied, we have $G_{\mathcal{B}\mathcal{Z}}[(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}]_{(:,\ell)} = 0$ and if condition 7(a) in Lemma 4.1 is satisfied, we have $G_{k\mathcal{Z}}[(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}]_{(:,\ell)} = 0$, $\breve{G}_{k\mathcal{O}}(I - \breve{G}_{\mathcal{O}\mathcal{O}})^{-1}G_{\mathcal{O}\mathcal{Z}}[(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}]_{(:,\ell)} = 0$. Therefore, $\bar{R}_{k\ell} = 0$ if conditions 7(a), 7(b) in Lemma 4.1 and Condition 4.1 are satisfied for $k \in \mathcal{Q}$ and $\ell \in \mathcal{Z}$. Consolidating the results, we have $\bar{R}_{k\ell} = 0$ if conditions 7(a), 7(b) in Lemma 4.1 and Condition 4.1 are satisfied for $k \in \mathcal{Y}$ and $\ell \in \mathcal{Z}$.

This concludes the final part of the Proof. ■

4.C Proof of Proposition 4.4

We will start by specifying the signals in $u_{\bar{\mathcal{P}}}$ considering all the excitation signals $u_{\mathcal{L}}$. We know $\mathcal{L} = \mathcal{Y} \cup \mathcal{U} \cup \mathcal{Z}$. It is implicit that the excitation signals in the set $\mathcal{Y}$ cannot contribute to $\bar{\mathcal{P}}$. We will first present the excitation signals in the set $\mathcal{U}$ that contribute to $\bar{\mathcal{P}}$. From the result of Lemma 4.1, we have $\bar{R}_{k\ell} = 0$ for all $k \in \mathcal{Y}$ and $\ell \in \mathcal{A}$ if Condition 3.2 is satisfied. Also, from the proof of Lemma 4.1, we can have non-zero dynamics in $\bar{R}_{k\ell}0$ for all $k \in \mathcal{Y}$ and $\ell \in \mathcal{B}$. Therefore, $u_{\mathcal{A}}$ belongs to $u_{\bar{\mathcal{P}}}$ if Condition 3.2 is satisfied. Now we will specify the signals in $\mathcal{Z}$ that contribute to $\bar{\mathcal{P}}$. For this we define the set $\bar{\mathcal{Z}}$.

Definition 4.2 *$\bar{\mathcal{Z}} \subseteq \mathcal{Z}$ denotes the indices of signals in u (i.e. $u_\ell, \ell \in \bar{\mathcal{Z}}$) that satisfies the following property:*

- *all paths from w_ℓ to $w_k, k \in \mathcal{Y}$ pass through a node in $w_{\mathcal{Q}\cup\mathcal{U}}$;*
- *there exist no direct or unmeasured paths from w_ℓ to any $w_m, m \in \mathcal{B}$.*

From the result of Lemma 4.1, we have $\bar{R}_{\mathcal{Y}\ell} = 0$ for $\ell \in \bar{\mathcal{Z}}$ if Condition 3.2 and 4.1 are satisfied. Therefore, $u_{\bar{\mathcal{Z}}}$ belongs to $u_{\bar{\mathcal{P}}}$ if Condition 3.2 and 4.1 are satisfied. After specifying $\mathcal{P}$ and $\bar{\mathcal{P}}$, we can now specify the external signals in u that belong to $u_{\mathcal{K}}$ by excluding $\mathcal{P}$ and $\bar{\mathcal{P}}$ from the list of excitation signals. Therefore, $\mathcal{K} = \mathcal{L}\backslash\{\mathcal{P} \cup \mathcal{A} \cup \bar{\mathcal{Z}}\}$. ■

4.D Proof of Theorem 4.1

Substituting expression (4.4) into the expression for the prediction error (4.5), leads to

$$\varepsilon(t,\theta): \quad = \quad \bar{H}(q,\theta)^{-1}\Big[\Delta\bar{G}(q,\theta)w_{\mathcal{D}} + \Delta\bar{J}(q,\theta)u_{\mathcal{K}} + \Delta\bar{H}(q,\theta)\xi_{\mathcal{Y}}\Big] + \xi_{\mathcal{Y}} \quad (4.43)$$

where $\Delta\bar{G}(q,\theta) = \bar{G}^0 - \bar{G}(q,\theta)$, $\Delta\bar{J}(q,\theta) = \bar{J}^0 - \bar{J}(q,\theta)$ and $\Delta\bar{H}(q,\theta) = \bar{H}^0 - \bar{H}(q,\theta)$. The proof of consistency involves two steps.

1. To show that $\bar{\mathbb{E}}\varepsilon^T(t,\theta)W\varepsilon(t,\theta)$ achieves its minimum for $\Delta\bar{G}(\theta) = 0$, $\Delta\bar{T}(\theta) = 0$ and $\Delta\bar{H}(\theta) = 0$,
2. To show the conditions under which the minimum is unique.

Step 1: From (4.36) it follows that our data generating system can always be written such that $w_m = T^{\xi}(q)\xi_m + T^u(q)u$. We denote T_1^{ξ}, T_1^u as the matrix composed of the first and third (block) row of T^{ξ} and T^u respectively, such that $w_{\mathcal{D}} = T_1^{\xi}(q)\xi_m + T_1^u(q)u$. Substituting this into (4.43) gives

$$\begin{aligned}\varepsilon(t,\theta) &:= \bar{H}(q,\theta)^{-1}\left[\Delta\bar{G}(q,\theta)T_1^{\xi} + \begin{bmatrix}\Delta\bar{H}(q,\theta) & 0\end{bmatrix}\right]\xi_m \\ &+H(q,\theta)^{-1}\left[\Delta\bar{G}(q,\theta)T_1^{u} + \begin{bmatrix}\Delta\bar{J}(q,\theta) & 0\end{bmatrix}\right]u + \xi_{\mathcal{Y}},\end{aligned} \quad (4.44)$$

where ξ_m is (block) structured as $[\xi_{\mathcal{Y}}^{\top}\ \xi_{\mathcal{U}}^{\top}]^{\top}$ and u is (block) structured as $[u_{\mathcal{K}}^{\top}\ r_{\mathcal{L}\backslash\mathcal{K}}^{\top}]^{\top}$. In order to prove that the minimum of $\bar{\mathbb{E}}\left[\varepsilon^T(t,\theta)W\varepsilon(t,\theta)\right]$ is attained for $\Delta\bar{G}(\theta) = 0$, $\Delta\bar{J}(\theta) = 0$ and $\Delta\bar{H}(\theta) = 0$, it is sufficient to show that

$$\left[\Delta\bar{G}(\theta)T_1^{\xi}(q) + \begin{bmatrix}\Delta\bar{H}(\theta) & 0\end{bmatrix}\right]\xi_m(t) + \left[\Delta\bar{G}(q,\theta)T_1^{u}(q) + \begin{bmatrix}\Delta\bar{J}(\theta) & 0\end{bmatrix}\right]u \quad (4.45)$$

is uncorrelated to $\xi_{\mathcal{Y}}(t)$. As a result of Condition d in the theorem, u is uncorrelated to $\xi_{\mathcal{Y}}$. Therefore, it is sufficient to shoe that

$$\left[\Delta\bar{G}(\theta)T_1^{\xi}(q) + \begin{bmatrix}\Delta\bar{H}(\theta) & 0 & 0\end{bmatrix}\right]\xi_m(t) \quad (4.46)$$

is uncorrelated to $\xi_{\mathcal{Y}}(t)$. In order to show this, let $\mathcal{F}_n = \mathcal{U}\backslash\mathcal{F}$, with $\mathcal{F}$ as defined in the Theorem, while we decompose ξ_m according to $\xi_m = [\xi_{\mathcal{Y}}^{\top} \quad \xi_{\mathcal{F}}^{\top} \quad \xi_{\mathcal{F}_n}^{\top}]^{\top}$. Using a similar block-structure notation for $\Delta\bar{G}$, $\Delta\bar{J}$, T_1^{ξ} and $\Delta\bar{H}$, (4.46) can then be written as

$$\begin{aligned}&\left(\Delta\bar{G}_{\mathcal{Y}\mathcal{Q}}(\theta)T_{\mathcal{Q}\mathcal{Y}} + \Delta\bar{G}_{\mathcal{Y}\mathcal{F}}(\theta)T_{\mathcal{F}\mathcal{Y}} + \Delta\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)T_{\mathcal{F}_n\mathcal{Y}} + \Delta\bar{H}_{\mathcal{Y}\mathcal{Y}}(\theta)\right)\xi_{\mathcal{Y}} + \\ &+ \left(\Delta\bar{G}_{\mathcal{Y}\mathcal{Q}}(\theta)T_{\mathcal{Q}\mathcal{F}} + \Delta\bar{G}_{\mathcal{Y}\mathcal{F}}(\theta)T_{\mathcal{F}\mathcal{F}} + \Delta\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)T_{\mathcal{F}_n\mathcal{F}}\right)\xi_{\mathcal{F}} \\ &+ \left(\Delta\bar{G}_{\mathcal{Y}\mathcal{Q}}(\theta)T_{\mathcal{Q}\mathcal{F}_n} + \Delta\bar{G}_{\mathcal{Y}\mathcal{F}}(\theta)T_{\mathcal{F}\mathcal{F}_n} + \Delta\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)T_{\mathcal{F}_n\mathcal{F}_n}\right)\xi_{\mathcal{F}_n}.\end{aligned} \quad (4.47)$$

Since, by definition, $\xi_{\mathcal{F}_n}(t)$ is statically uncorrelated to $\xi_{\mathcal{Y}}(t)$, the $\xi_{\mathcal{F}_n}$-dependent term in (4.47) cannot create any static correlation with $\xi_{\mathcal{Y}}(t)$. Then it needs to be shown that the $\xi_{\mathcal{Y}}$- and $\xi_{\mathcal{F}}$-dependent terms in (4.47) all reflect strictly proper

filters. i.e. that they all contain at least a delay.
$\Delta\bar{H}(\theta)$ is strictly proper since both $\bar{H}(\theta)$ and $\bar{H}^o$ are monic. Therefore, $\Delta\bar{H}_{\mathcal{Y}\mathcal{Y}}(\theta)$ will have at least a delay in each of its transfers.
If all paths from $w_{\mathcal{Y}\cup\mathcal{F}}$ to $w_{\mathcal{Y}}$ in the *transformed network* and in its parameterized model have at least a delay (as per Condition c in the theorem), then all terms $\Delta\bar{G}_{\mathcal{Y}\mathcal{Q}}(\theta)$ and $\Delta\bar{G}_{\mathcal{Y}\mathcal{F}}(\theta)$ will have a delay.
We then need to consider the two remaining terms, $\Delta\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)T_{\mathcal{F}_n\mathcal{Y}}$ and $\Delta\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)T_{\mathcal{F}_n\mathcal{F}}$. From the definition of $\Delta\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)$, each of the two terms can be represented as the sum of two terms. $\bar{G}_{\mathcal{Y}\mathcal{F}_n}T_{\mathcal{F}_n\mathcal{Y}}$ and $\bar{G}_{\mathcal{Y}\mathcal{F}_n}T_{\mathcal{F}_n\mathcal{F}}$ represent paths from $w_{\mathcal{Y}}$ to $w_{\mathcal{Y}}$ and from $w_{\mathcal{F}}$ to $w_{\mathcal{Y}}$ respectively in the *transformed network*. Whereas, $\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)T_{\mathcal{F}_n\mathcal{Y}}$ and $\bar{G}_{\mathcal{Y}\mathcal{F}_n}(\theta)T_{\mathcal{F}_n\mathcal{F}}$ is partly induced by the parameterized model and partly by the paths from $w_{\mathcal{Y}}$ to $w_{\mathcal{F}_n}$ and from $w_{\mathcal{F}}$ to $w_{\mathcal{F}_n}$ respectively in the *transformed network*. According to condition c of the theorem (delay conditions), these transfer functions are strictly proper. This implies that (4.47) is statically uncorrelated to $\xi_{\mathcal{Y}}(t)$. Therefore we have, $\bar{\mathbb{E}}\left[\varepsilon^T(t,\theta)W\varepsilon(t,\theta)\right] = \bar{\mathbb{E}}\left[||\Delta X_1(\theta)\xi_m + \Delta X_2(\theta)u||_W^2\right] + \bar{\mathbb{E}}\left[\xi_{\mathcal{Y}}^\top W\xi_{\mathcal{Y}}\right]$ where $\Delta X_1(\theta) = \bar{H}(\theta)^{-1}\left[\Delta\bar{G}(\theta)T_1^{\xi}(q) + \left[\Delta\bar{H}(\theta) \quad 0 \quad 0\right]\right]$ and $\Delta X_2(\theta) = \bar{H}(\theta)^{-1}\left[\Delta\bar{G}(\theta)T_1^{u}(q) + \left[\Delta\bar{J}(\theta) \quad 0\right]\right]$. As a result, the minimum of $\bar{\mathbb{E}}\left[\varepsilon^T(t,\theta)W\varepsilon(t,\theta)\right]$, which is $\bar{\mathbb{E}}\left[\xi_{\mathcal{Y}}^\top W\xi_{\mathcal{Y}}\right]$, is achieved for $\Delta\bar{G}(\theta) = 0$, $\Delta\bar{J}(\theta) = 0$ and $\Delta\bar{H}(\theta) = 0$.

Step 2: When the minimum is achieved, we have $\bar{\mathbb{E}}\left[||\Delta X_1(\theta)\xi_m + \Delta X_2(\theta)u||_W^2\right]$ to be zero. From (4.43), we have $\Delta X_1(\theta)\xi_m + \Delta X_2(\theta)u = \bar{H}(q,\theta)^{-1}\left[\left[\Delta\bar{G}(q,\theta) \;\; \Delta\bar{J}(q,\theta) \;\; \Delta\bar{H}(q,\theta)\right]\left[w_{\mathcal{D}}^\top \;\; u_{\mathcal{K}}^\top \;\; \xi_{\mathcal{Y}}^\top\right]^\top\right] = \Delta x(\theta)\kappa(t)$. Writing $\bar{\mathbb{E}}\left[||\Delta X_1(\theta)\xi_m + \Delta X_2(\theta)u||_W^2\right] = \bar{\mathbb{E}}\left[||\Delta x(\theta)\kappa(t)||_W^2\right] = 0$ using Parseval's theorem in the frequency domain, we have

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \Delta x(e^{j\omega},\theta)^\top\Phi_\kappa(\omega)\Delta x(e^{-j\omega},\theta)d\omega = 0. \tag{4.48}$$

The standard reasoning for showing uniqueness of the identification result is to show that if $\bar{\mathbb{E}}\left[||\Delta X_1(\theta)\xi_m + \Delta X_2(\theta)u||_W^2\right]$ equals 0 (i.e. when the minimum power is achieved), this should imply that $\Delta\bar{G}(\theta) = 0, \Delta\bar{J}(\theta) = 0$ and $\Delta\bar{H}(\theta) = 0$. Looking into equation (4.48), this implication will be fulfilled only if $\Phi_\kappa(\omega) > 0$ for a sufficiently high number of frequencies. On condition 2 of Theorem 4.1 being satisfied along with the other conditions in Theorem 1, it ensures that the minimum value is achieved only for $\bar{G}(\theta) = \bar{G}^0$, $\bar{J}(\theta) = \bar{J}^0$ and $\bar{H}(\theta) = \bar{H}^0$. ■

4.E Proof of Proposition 4.5

Considering $\mathcal{C}_1 = \mathcal{Y} \cup \mathcal{B}, \mathcal{C}_2 = \mathcal{A}$ and $\mathcal{V} = \mathcal{Z}$, the disturbances in the original network are characterized by $\breve{v}$ (4.38). From the results of Lemma 4.3 and since Condition 3.2 is satisfied, we can infer that the spectral density $\Phi_{\breve{v}}$ has the unique spectral factorization $\Phi_{\breve{v}} = \tilde{H}\Lambda\tilde{H}^*$ where $\tilde{H}$ is monic, stable, minimum phase, and of the form given in (4.37). Together with the form of Λ in (4.37) it follows that $\xi_{\mathcal{A}}$ is uncorrelated with $\xi_{\mathcal{Y}}$. As a result, the set $\mathcal{A}$ satisfies the properties of $\mathcal{F}_n$, so that in Condition c we can replace $\mathcal{F}$ by a $\mathcal{B}$. What remains to be shown is that the delay in path/loop conditions in the transformed network (3.4) can be reformulated into the same conditions on the original network. Following the same approach as in the proof of proposition 3.2 we end up with the result of the proposition. ■

4.F Proof of Theorem 4.2

In order to prove Theorem 4.2, we first present a preparatory Lemma.

Lemma 4.5 *Consider the network in (3.4) (also (4.4)). If condition 3.2 is satisfied, then $\bar{G}_{j\ell}$ is given by the following expressions:*

$$\text{If } j \in \mathcal{O}, \ell \in \mathcal{D} : \bar{G}_{j\ell}=[(I-\breve{G}_{\mathcal{O}\mathcal{O}} + \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}})^{-1}]_{(j,:)}(\breve{G}_{\mathcal{O}\ell}-\grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\ell}+\grave{H}_{\mathcal{O}\ell}). \quad (4.49)$$

$$\text{If } j \in \mathcal{Q}, \ell \in \mathcal{D} : \bar{G}_{j\ell}=\Big(I-\breve{G}_{jj}+\grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}j} - \breve{G}'_{j\mathcal{O}}(I - \breve{G}_{\mathcal{O}\mathcal{O}})^{-1}(\breve{G}_{\mathcal{O}j} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}j})\Big)^{-1}$$
$$\Big(\breve{G}_{j\ell}-\grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}\ell}+\grave{H}_{j\ell} + \breve{G}'_{j\mathcal{O}}(I - \breve{G}_{\mathcal{O}\mathcal{O}})^{-1}(\breve{G}_{\mathcal{O}\ell}-\grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\ell}+\grave{H}_{\mathcal{O}\ell})\Big)(4.50)$$

where $\grave{H}_{\mathcal{O}\ell} = \grave{H}_{j\ell} = 0$ if $\ell \in \mathcal{Q} \cup \mathcal{A}$.

Proof: Using the result in Lemma 4.4, if Condition 3.2 is satisfied, then $\grave{H}_{\mathcal{Y}\mathcal{A}} = 0$. We have the following cases that can occur:

1. $j = \mathcal{O}$ and $\ell \in \mathcal{U}$. From (4.23) we have $\bar{G}_{j\ell} = [(I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}]_{(j,:)}\breve{G}'_{\mathcal{O}\ell}$ where $\breve{G}'_{\mathcal{O}\mathcal{O}} = \breve{G}_{\mathcal{O}\mathcal{O}} - \grave{H}_{\mathcal{O}\mathcal{U}}\breve{G}_{\mathcal{U}\mathcal{O}}$ is given by (4.17) and $\breve{G}'_{\mathcal{O}\ell} = \breve{G}_{\mathcal{O}\ell} - \grave{H}_{\mathcal{O}\mathcal{U}}\breve{G}_{\mathcal{U}\ell} + \grave{H}_{\mathcal{O}\ell}$ is given by (4.18). Since $\grave{H}_{\mathcal{O}\mathcal{A}} = 0$, we have $\breve{G}'_{\mathcal{O}\mathcal{O}} = \breve{G}_{\mathcal{O}\mathcal{O}} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}}$ and $\breve{G}'_{\mathcal{O}\ell} = \breve{G}_{\mathcal{O}\ell} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\ell} + \grave{H}_{\mathcal{O}\ell}$ with $\grave{H}_{\mathcal{O}\ell} = 0$ if $\ell \in \mathcal{A}$. This leads to (10.52).

2. $j = \mathcal{O}$ and $\ell \in \mathcal{Q}$. From (4.23) we have $\bar{G}_{j\ell} = [(I-\breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}]_{(j,:)}\breve{G}'_{\mathcal{O}\ell}$ where $\breve{G}'_{\mathcal{O}\mathcal{O}}$ and $\breve{G}'_{\mathcal{O}\ell}$ are given by (4.17). Since $\grave{H}_{\mathcal{O}\mathcal{A}} = 0$, we have $\breve{G}'_{\mathcal{O}\mathcal{O}} = \breve{G}_{\mathcal{O}\mathcal{O}} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}}$ and $\breve{G}'_{\mathcal{O}\ell} = \breve{G}_{\mathcal{O}\ell} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\ell}$. This leads to (10.52).

3. $j \in \mathcal{Q}$ and $\ell \in \mathcal{U}$. From (4.31) we have $\bar{G}_{j\ell} = (I - \breve{G}''_{jj})^{-1}\breve{G}''_{j\ell}$ where $\breve{G}''_{jj} = \breve{G}'_{jj} + \breve{G}'_{j\mathcal{O}}(I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}\breve{G}'_{\mathcal{O}j}$ and $\breve{G}''_{j\ell} = \breve{G}'_{j\ell} + \breve{G}'_{j\mathcal{O}}(I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}\breve{G}'_{\mathcal{O}\ell}$ are given by (4.25). If Condition 3.2 is satisfied, using (4.17), we have $\breve{G}'_{jj} = \breve{G}_{jj} - \grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}j}$, $\breve{G}'_{j\mathcal{O}} = \breve{G}_{j\mathcal{O}} - \grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}}$, $\breve{G}'_{\mathcal{O}j} = \breve{G}_{\mathcal{O}j} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}j}$ and $\breve{G}'_{\mathcal{O}\mathcal{O}} = \breve{G}_{\mathcal{O}\mathcal{O}} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}}$. If condition (3.2) is satisfied, using (4.18), we have $\breve{G}'_{j\ell} = \breve{G}_{j\ell} - \grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}\ell} + \grave{H}_{j\ell}$ and $\breve{G}'_{\mathcal{O}\ell} = \breve{G}_{\mathcal{O}\ell} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\ell} + \grave{H}_{\mathcal{O}\ell}$ where $\grave{H}_{\mathcal{O}\ell} = \grave{H}_{j\ell} = 0$ if $\ell \in \mathcal{A}$. This leads to (4.50).

4. $j \in \mathcal{Q}$ and $\ell \in \mathcal{Q}$. Since $j \neq \ell$ it follows from (4.30) that $\bar{G}_{j\ell} = (I - \breve{G}''_{jj})^{-1}\breve{G}''_{j\ell}$ where $\breve{G}''_{jj}$ and $\breve{G}''_{j\ell}$ are given by (4.25). If Condition 3.2 is satisfied, using (4.17), we have $\breve{G}'_{jj} = \breve{G}_{jj} - \grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}j}$, $\breve{G}'_{j\mathcal{O}} = \breve{G}_{j\mathcal{O}} - \grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}}$, $\breve{G}'_{\mathcal{O}j} = \breve{G}_{\mathcal{O}j} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}j}$ and $\breve{G}'_{\mathcal{O}\mathcal{O}} = \breve{G}_{\mathcal{O}\mathcal{O}} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}}$. If condition (3.2) is satisfied, using (4.17), we have $\breve{G}'_{j\ell} = \breve{G}_{j\ell} - \grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}\ell}$ and $\breve{G}'_{\mathcal{O}\ell} = \breve{G}_{\mathcal{O}\ell} - \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\ell}$. This leads to (4.50).

■

Next we proceed with the proof of Theorem 4.2. With Lemma 10.4 it follows that $\bar{G}_{ji}$ is given by either (10.52) or (4.50) if $\ell = i$. For analysing the expression, we first are going to specify $\breve{G}_{ji}$ and $\breve{G}_{jj}$. From (4.8), we have $\breve{G}_{ji} = G_{ji} + G_{j\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}G_{\mathcal{Z}i}$ and $\breve{G}_{jj} = G_{jj} + G_{j\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}G_{\mathcal{Z}j}$. The first terms on the right hand sides reflect the direct connections from w_i to w_j (similarly w_j to w_j) and the second terms reflect the connections that pass only through nodes in $\mathcal{Z}$. By definition, $G_{jj} = 0$ since the G matrix in the network in (2.2) is hollow. Under condition 1 in Theorem 4.2 being satisfied, the second terms on the right hand sides are zero, so that $\breve{G}_{ji} = G_{ji}$ and $\breve{G}_{jj} = 0$.

Under condition 2(c) in Theorem 4.2 being satisfied, we have $\breve{G}_{\mathcal{B}j} = \breve{G}_{\mathcal{B}i} = 0$ and $\breve{G}_{\mathcal{B}\mathcal{O}} = 0$ irrespective of whether j belongs to $\mathcal{O}$ or not. Since condition 2(b) in Theorem 4.2 is satisfied, we have $\grave{H}_{\mathcal{O}\ell} = \grave{H}_{j\ell} = 0$. Under condition 1 in Theorem 4.2 being satisfied, the term $\breve{G}_{j\mathcal{O}}(I - \breve{G}_{\mathcal{O}\mathcal{O}})^{-1}\breve{G}_{\mathcal{O}j}$ and $\breve{G}_{j\mathcal{O}}(I - \breve{G}_{\mathcal{O}\mathcal{O}})^{-1}\breve{G}_{\mathcal{O}i}$ are zero. This implies that $\bar{G}_{ji} = G_{ji}$ if $j \in \mathcal{Q}$ and $\ell \in \mathcal{D}$. Similarly, if parallel path condition 1 in Theorem 4.2 is satisfied, we have $\bar{G}_{ji} = [(I - \breve{G}_{\mathcal{O}\mathcal{O}})^{-1}]_{(j,:)}\breve{G}_{\mathcal{O}i} = (I - \breve{G}_{jj})^{-1}\breve{G}_{ji}$. Since $\breve{G}_{jj} = 0$ and $\breve{G}_{ji} = G_{ji}$, we have $\bar{G}_{ji} = G_{ji}$ if $j \in \mathcal{O}$ and $\ell \in \mathcal{D}$ as well. ■

4.G Proof of Theorem 4.3

We first present the following preparatory lemmas.

Lemma 4.6 *Let $\mathcal{C} \subset \mathcal{L}$ with cardinality c and $\mathcal{V} = \mathcal{L} \backslash \mathcal{C}$. Let $\mathcal{V}_1$ and $\mathcal{V}_2$ be two disjoint sets such that $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$. The system equations for node variables $w_{\mathcal{C}}$ can be written as,*

$$w_{\mathcal{C}} = \breve{G}_{\mathcal{C}\mathcal{C}}w_{\mathcal{C}} + \tilde{H}\tilde{\xi}_{\mathcal{C}} + \breve{R}u\,, \tag{4.51}$$

where $\tilde{H}$ is stable, monic and minimum phase rational $c \times c$ matrix, and $\tilde{\xi}_c$ is a $c \times 1$ white noise process with cov($\tilde{\xi}_c$) = $\tilde{\Lambda} \in \mathbb{R}^{c \times c}$, and

$$\begin{aligned}
\breve{G}_{kh} &= G_{kh} + G_{k\mathcal{V}_2}(I - G_{\mathcal{V}_2\mathcal{V}_2})^{-1}G_{\mathcal{V}_2h}\,, && (4.52)\\
\breve{R}_{k\mathcal{V}_2} &= G_{k\mathcal{V}_2}(I - G_{\mathcal{V}_2\mathcal{V}_2})^{-1} && (4.53)\\
\breve{G}_{\mathcal{C}\mathcal{C}} &= \breve{G}_{\mathcal{C}\mathcal{C}} + \breve{G}_{\mathcal{C}\mathcal{V}_1}(I - \breve{G}_{\mathcal{V}_1\mathcal{V}_1})^{-1}\breve{G}_{\mathcal{V}_1\mathcal{C}}\,, && (4.54)\\
\breve{R}_{\mathcal{C}\mathcal{V}_1} &= \breve{G}_{\mathcal{C}\mathcal{V}_1}(I - \breve{G}_{\mathcal{V}_1\mathcal{V}_1})^{-1}\,, && (4.55)
\end{aligned}$$

$$\begin{aligned} \breve{R}_{\mathcal{C}\mathcal{V}_2} &= \breve{R}_{\mathcal{C}\mathcal{V}_2} + \breve{G}_{\mathcal{C}\mathcal{V}_1}(I - \breve{G}_{\mathcal{V}_1\mathcal{V}_1})^{-1}\breve{R}_{\mathcal{V}_1\mathcal{V}_2}\,, && (4.56)\\ \breve{R} &= \begin{bmatrix} I & \breve{R}_{\mathcal{C}\mathcal{V}_1} & \breve{R}_{\mathcal{C}\mathcal{V}_2} \end{bmatrix}, && (4.57) \end{aligned}$$

with $k, h \in \mathcal{C} \cup \mathcal{V}_1$.

Proof: On the basis of the decomposition of node signals as defined in the lemma, we are going to rewrite the system's equations (3.1) in the following structured form:

$$\begin{bmatrix} w_{\mathcal{C}} \\ w_{\mathcal{V}_1} \\ w_{\mathcal{V}_2} \end{bmatrix} = \begin{bmatrix} G_{\mathcal{C}\mathcal{C}} & G_{\mathcal{C}\mathcal{V}_1} & G_{\mathcal{C}\mathcal{V}_2} \\ G_{\mathcal{V}_1\mathcal{C}} & G_{\mathcal{V}_1\mathcal{V}_1} & G_{\mathcal{V}_1\mathcal{V}_2} \\ G_{\mathcal{V}_2\mathcal{C}} & G_{\mathcal{V}_2\mathcal{V}_1} & G_{\mathcal{V}_2\mathcal{V}_2} \end{bmatrix} \begin{bmatrix} w_{\mathcal{C}} \\ w_{\mathcal{V}_1} \\ w_{\mathcal{V}_2} \end{bmatrix} + \begin{bmatrix} H_{\mathcal{C}\mathcal{C}} & H_{\mathcal{C}\mathcal{V}_1} & H_{\mathcal{C}\mathcal{V}_2} \\ H_{\mathcal{V}_1\mathcal{C}} & H_{\mathcal{V}_1\mathcal{V}_1} & H_{\mathcal{V}_1\mathcal{V}_2} \\ H_{\mathcal{V}_2\mathcal{C}} & H_{\mathcal{V}_2\mathcal{V}_1} & H_{\mathcal{V}_2\mathcal{V}_2} \end{bmatrix} \begin{bmatrix} e_{\mathcal{C}} \\ e_{\mathcal{V}_1} \\ e_{\mathcal{V}_2} \end{bmatrix} + \begin{bmatrix} I\,0\,0 \\ 0\,I\,0 \\ 0\,0\,I \end{bmatrix} \begin{bmatrix} u_{\mathcal{C}} \\ u_{\mathcal{V}_1} \\ u_{\mathcal{V}_2} \end{bmatrix} \quad (4.58)$$

We can eliminate the node variables $w_{\mathcal{V}_2}$ from the equation, by writing the last (block) row of (4.58) into an explicit expression for $w_{\mathcal{V}_2}$:

$$w_{\mathcal{V}_2} = (I - G_{\mathcal{V}_2\mathcal{V}_2})^{-1} \left[\sum_{k \in \mathcal{C} \cup \mathcal{V}_1} G_{\mathcal{V}_2 k} w_k + \sum_{k \in \mathcal{C} \cup \mathcal{V}} H_{\mathcal{V}_2 k} e_k + u_{\mathcal{V}_2} \right],$$

and by substituting this $w_{\mathcal{V}_2}$ into the expressions for other variables. As a result, we have

$$\begin{bmatrix} w_{\mathcal{C}} \\ w_{\mathcal{V}_1} \end{bmatrix} = \begin{bmatrix} \breve{G}_{\mathcal{C}\mathcal{C}} & \breve{G}_{\mathcal{C}\mathcal{V}_1} \\ \breve{G}_{\mathcal{V}_1\mathcal{C}} & \breve{G}_{\mathcal{V}_1\mathcal{V}_1} \end{bmatrix} \begin{bmatrix} w_{\mathcal{C}} \\ w_{\mathcal{V}_1} \end{bmatrix} + \begin{bmatrix} \breve{H}_{\mathcal{C}\mathcal{C}} & \breve{H}_{\mathcal{C}\mathcal{V}_1} & \breve{H}_{\mathcal{C}\mathcal{V}_2} \\ \breve{H}_{\mathcal{V}_1\mathcal{C}} & \breve{H}_{\mathcal{V}_1\mathcal{V}_1} & \breve{H}_{\mathcal{V}_1\mathcal{V}_2} \end{bmatrix} \begin{bmatrix} e_{\mathcal{C}} \\ e_{\mathcal{V}_1} \\ e_{\mathcal{V}_2} \end{bmatrix} + \begin{bmatrix} I\;0\;\breve{R}_{\mathcal{C}\mathcal{V}_2} \\ 0\;I\;\breve{R}_{\mathcal{V}_1\mathcal{V}_2} \end{bmatrix} \begin{bmatrix} u_{\mathcal{C}} \\ u_{\mathcal{V}_1} \\ u_{\mathcal{V}_2} \end{bmatrix}, \quad (4.59)$$

where

$$\begin{aligned} \breve{G}_{kh} &= G_{kh} + G_{k\mathcal{V}_2}(I - G_{\mathcal{V}_2\mathcal{V}_2})^{-1}G_{\mathcal{V}_2 h}\,, && (4.60)\\ \breve{R}_{k\mathcal{V}_2} &= G_{k\mathcal{V}_2}(I - G_{\mathcal{V}_2\mathcal{V}_2})^{-1} && (4.61)\\ \breve{H}_{k\ell} &= H_{k\ell} + G_{k\mathcal{V}_2}(I - G_{\mathcal{V}_2\mathcal{V}_2})^{-1}H_{\mathcal{V}_2\ell}\,, && (4.62) \end{aligned}$$

with $k, h \in \mathcal{C} \cup \mathcal{V}_1$ and $\ell \in \mathcal{C} \cup \mathcal{V}_1 \cup \mathcal{V}_2$. Now we eliminate the node variables $w_{\mathcal{V}_1}$ from the above equation, by writing the last (block) row of (4.59) into an explicit expression for $w_{\mathcal{V}_1}$:

$$w_{\mathcal{V}_1} = (I - \breve{G}_{\mathcal{V}_1\mathcal{V}_1})^{-1} \left[\breve{G}_{\mathcal{V}_1\mathcal{C}} w_{\mathcal{C}} + \sum_{k \in \mathcal{C} \cup \mathcal{V}} \breve{H}_{\mathcal{V}_1 k} e_k + u_{\mathcal{V}_1} + \breve{R}_{\mathcal{V}_2\mathcal{V}_2} u_{\mathcal{V}_2} \right],$$

and by substituting this $w_{\mathcal{V}_2}$ into the expressions for $w_{\mathcal{C}}$ variable. As a result, we have

$$w_{\mathcal{C}} = \breve{G}_{\mathcal{C}\mathcal{C}} w_{\mathcal{C}} + \sum_{k \in \mathcal{C} \cup \mathcal{V}} \breve{H}_{\mathcal{C}k} e_k + u_{\mathcal{C}} + \breve{R}_{\mathcal{C}\mathcal{V}_1} u_{\mathcal{V}_1} + \breve{R}_{\mathcal{C}\mathcal{V}_2} u_{\mathcal{V}_2}\,, \quad (4.63)$$

where $\breve{G}_{\mathcal{C}\mathcal{C}}$, $\breve{R}_{\mathcal{C}\mathcal{V}_1}$, $\breve{R}_{\mathcal{C}\mathcal{V}_2}$ are given by (4.54), (4.55), (4.56) respectively, and $\breve{H}_{\mathcal{C}\ell} = \breve{H}_{\mathcal{C}\ell} + \breve{G}_{\mathcal{C}\mathcal{V}_1}(I - \breve{G}_{\mathcal{V}_1\mathcal{V}_1})^{-1}\breve{H}_{\mathcal{V}_1\ell}$, with $\ell \in \mathcal{C} \cup \mathcal{V}$.

Let $\breve{v} = \breve{H}e = \begin{bmatrix} \breve{H}_{\mathcal{C}\mathcal{C}} & \breve{H}_{\mathcal{C}\mathcal{V}_1} & \breve{H}_{\mathcal{C}\mathcal{V}_2} \end{bmatrix} \begin{bmatrix} e_{\mathcal{C}}^\top & e_{\mathcal{V}_1}^\top & e_{\mathcal{V}_2}^\top \end{bmatrix}^\top$. Then the spectral density of $\breve{v}$

in (4.63) is given by $\Phi_{\breve{v}} = \breve{H}\breve{H}^*$. Applying a spectral factorization [150] to $\Phi_{\breve{v}}$ will deliver $\Phi_{\breve{v}} = \tilde{H}\tilde{\Lambda}\tilde{H}^*$ with $\tilde{H}$ a monic, stable and minimum phase rational matrix, and $\tilde{\Lambda}$ a positive definite (constant) matrix. Then there exists a white noise process $\tilde{\xi}_c$ defined by $\tilde{\xi}_c := \tilde{H}^{-1}\breve{H}e$ such that $\tilde{H}\tilde{\xi}_c = \breve{v}$, with cov($\tilde{\xi}$) = $\tilde{\Lambda}$. Thus we get the result of the lemma. ■

Lemma 4.7 *If condition 4.2 and Condition 3.2 are satisfied, then*

$$\bar{G}_{ji} = (1 - \breve{G}'_{jj})^{-1}\breve{G}'_{ji}, \tag{4.64}$$

with $\breve{G}'_{jj} = \breve{G}_{jj} - \grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}j}$ *and* $\breve{G}'_{ji} = \breve{G}_{ji} - \grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}i}$.

Proof: Using the result of lemma 10.4, we have $\bar{G}_{ji}$ given by either (10.52) or (4.50) if condition 4.2 and condition 3.2 are satisfied. Since condition 4.2 is satisfied, we have $\grave{H}_{\mathcal{O}i} = \grave{H}_{ji} = 0$. When condition 4.2 is satisfied, we have $\breve{G}_{jk} = 0, k \in \mathcal{O}\backslash\{j\}$. Substituting these results in (4.50), we get (4.64). Since condition 4.2 is satisfied, we have $\breve{G}_{jk} = 0, k \in \mathcal{O}\backslash\{j\}$ and $\breve{G}_{\mathcal{B}k} = 0, k \in \mathcal{O}\backslash\{j\}$. This implies that $[(I - \breve{G}_{\mathcal{O}\mathcal{O}} + \grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{O}})^{-1}]_{(j,:)} = (1 - \breve{G}_{jj} + \grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}j})^{-1}$. Therefore, substituting this result in (10.52), we get (4.64). ■

Now we present the proof of Theorem 4.3. The target module that is the objective of our identification is given by G_{ji}, with $w_j \in (w_{\mathcal{Q}}, w_{\mathcal{O}})$ and $w_i \in (w_{\mathcal{U}}, w_{\mathcal{Q}})$. From (4.64), we have $\bar{G}_{ji} = (1 - \breve{G}'_{jj})^{-1}\breve{G}'_{ji}$. Let $\mathcal{Z} = \mathcal{Z}_r \cup \mathcal{Z}_u$. Considering $\mathcal{C} = \mathcal{D} \cup \mathcal{Y} = \mathcal{Q} \cup \mathcal{O} \cup \mathcal{U}$, $\mathcal{V}_1 = \mathcal{Z}_r$ and $\mathcal{V}_2 = \mathcal{Z}_u$, using the result of Lemma 4.6 we can write $\bar{G}_{ji} = (I - \breve{G}'_{jj})^{-1}\breve{G}'_{ji}$ where,

$$\begin{aligned}
\breve{G}_{ji} &= G_{ji} + G_{j\mathcal{Z}_u}(I - G_{\mathcal{Z}_u\mathcal{Z}_u})^{-1}G_{\mathcal{Z}_u i} + \check{G}_{j\mathcal{Z}_r}(I - \check{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\check{G}_{\mathcal{Z}_r i} && (4.65)\\
\breve{G}_{jj} &= G_{jj} + G_{j\mathcal{Z}_u}(I - G_{\mathcal{Z}_u\mathcal{Z}_u})^{-1}G_{\mathcal{Z}_u j} + \check{G}_{j\mathcal{Z}_r}(I - \check{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\check{G}_{\mathcal{Z}_r j}, && (4.66)\\
\breve{G}_{\mathcal{B}i} &= G_{\mathcal{B}i} + G_{\mathcal{B}\mathcal{Z}_u}(I - G_{\mathcal{Z}_u\mathcal{Z}_u})^{-1}G_{\mathcal{Z}_u i} + \check{G}_{\mathcal{B}\mathcal{Z}_r}(I - \check{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\check{G}_{\mathcal{Z}_r i} && (4.67)\\
\breve{G}_{\mathcal{B}j} &= G_{\mathcal{B}j} + G_{\mathcal{B}\mathcal{Z}_u}(I - G_{\mathcal{Z}_u\mathcal{Z}_u})^{-1}G_{\mathcal{Z}_u j} + \check{G}_{\mathcal{B}\mathcal{Z}_r}(I - \check{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\check{G}_{\mathcal{Z}_r j}, && (4.68)
\end{aligned}$$

where the first terms on the right hand sides reflect the direct connections from w_i to w_j (respectively from w_j to w_j, $w_k, k \in \mathcal{B}$ to w_i, $w_k, k \in \mathcal{B}$ to w_j), the second terms reflect the connections that pass only through nodes in $\mathcal{Z}_u$ and the third terms reflect the connections that pass through nodes in both $\mathcal{Z}_u$ and $\mathcal{Z}_r$. By definition, $G_{jj} = 0$ since the G matrix in the network in (3.1) is hollow. **Condition 1 in Property 4.1** and **Condition 4(a) in Property 4.1** ensures that the second terms on the right hand sides are zero, so that $\breve{G}_{ji} = G_{ji} + \check{G}_{j\mathcal{Z}_r}(I - \check{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\check{G}_{\mathcal{Z}_r i}$ and $\breve{G}_{jj} = \check{G}_{j\mathcal{Z}_r}(I - \check{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\check{G}_{\mathcal{Z}_r j}$. **Condition 7 in Property 4.1** ensures that $G_{\mathcal{B}i} = G_{\mathcal{B}j} = 0$ and the respective second terms in the equations are zero. Therefore, $\breve{G}_{\mathcal{B}j} = \check{G}_{\mathcal{B}\mathcal{Z}_r}(I - \check{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\check{G}_{\mathcal{Z}_r j}$ and $\breve{G}_{\mathcal{B}i} = \check{G}_{\mathcal{B}\mathcal{Z}_r}(I - \check{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\check{G}_{\mathcal{Z}_r i}$. Now in the sequel we find expressions for $(I - \breve{G}_{jj})^{-1}$ using elements of (4.4) in order to extract G_{ji} from $\bar{G}_{ji}$.

Obtaining the expression of $(I - \breve{G}'_{jj})^{-1}$:

When condition 4.2 is satisfied, we have $\breve{G}_{jk} = \breve{G}_{\mathcal{B}k} = 0, k \in \mathcal{O}\backslash\{j\}$. This implies that $[(I - \breve{G}'_{\mathcal{OO}})^{-1}]_{(j,;)} = (1 - \breve{G}'_{jj})^{-1}$ when $j \in \mathcal{O}$ and $(1 - \breve{G}''_{jj})^{-1} = (1 - \breve{G}'_{jj})^{-1}$ when $j \in \mathcal{Q}$. If there are unmeasured loops through w_j, then $\breve{G}_{jj} \neq 0$. From (4.27) and (4.33), we have $\bar{J}_{jj} = (I - \breve{G}'_{jj})^{-1}$ when $j \in \mathcal{Y}$. Therefore if node j is excited by an external excitation signal u_j, then $\bar{J}_{jj} = (I - \breve{G}'_{jj})^{-1}$. Now, we look into situation when node j is not excited by an external excitation signal (i.e. $u_j = 0$).

If $j \in \mathcal{Q}$, from (4.28) we know that, $\breve{R}''_{j\mathcal{Z}_r} = \breve{R}'_{j\mathcal{Z}_r} + \breve{G}'_{j\mathcal{O}}\bar{R}_{\mathcal{O}\mathcal{Z}_r}$ where we have $\breve{R}'_{j\mathcal{Z}_r} = \breve{R}_{j\mathcal{Z}_r} - \grave{H}_{j\mathcal{B}}\breve{R}_{\mathcal{B}\mathcal{Z}_r}$. Since $\breve{G}_{jk} = \breve{G}_{\mathcal{B}k} = 0, k \in \mathcal{O}\backslash\{j\}$ and since $j \in \mathcal{Q}$, we have $\breve{G}_{j\mathcal{O}} = \breve{G}_{\mathcal{BO}} = 0$. Therefore, $\breve{G}'_{j\mathcal{O}} = \breve{G}_{j\mathcal{O}} - \grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{BO}} = 0$ when $j \in \mathcal{Q}$. This implies that $\breve{R}''_{j\mathcal{Z}_r} = \breve{R}'_{j\mathcal{Z}_r}$ when $j \in \mathcal{Q}$. Then, from (4.27) and (4.34) we have $\bar{R}_{j\mathcal{Z}_r} = (I - \breve{G}'_{jj})^{-1}\breve{R}'_{j\mathcal{Z}_r}$ for $j \in \mathcal{Y}$. Expanding $\breve{R}_{j\mathcal{Z}_r}$ using (4.55), we have $\breve{R}_{j\mathcal{Z}_r} = \breve{G}_{j\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}$. Similarly, we can write $\breve{R}_{\mathcal{B}\mathcal{Z}_r} = \breve{G}_{\mathcal{B}\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}$. If **condition 5 in Property 4.1**, then

$$(I - \breve{G}'_{jj})\bar{R}_{j\mathcal{Z}_r} = \breve{G}_{j\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1} - \grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}. \tag{4.69}$$

Post-multiplying the above equation with $\breve{G}_{\mathcal{Z}_r j}$, we get

$$(I - \breve{G}'_{jj})\bar{R}_{j\mathcal{Z}_r}\breve{G}_{\mathcal{Z}_r j} = \breve{G}'_{jj}. \tag{4.70}$$

Now we look in to getting the expression of $\breve{G}_{\mathcal{Z}_r j}$ using elements of (4.4). Let $\mathcal{T} = \mathcal{T}_{\mathcal{O}} \cup \mathcal{T}_{\mathcal{Q}}$ be two disjoint sets such that $\mathcal{T}_{\mathcal{Q}} \in \mathcal{Q}$ and $\mathcal{T}_{\mathcal{O}} \in \mathcal{O}$. Using the similar reasoning as above, if **condition 5 in Property 4.1**, we can write

$$\begin{aligned}\bar{R}_{\mathcal{T}_{\mathcal{Q}}\mathcal{Z}_r} = {} & (I - \text{diag}(\breve{G}''_{\mathcal{T}_{\mathcal{Q}}\mathcal{T}_{\mathcal{Q}}}))^{-1}(\breve{G}_{\mathcal{T}_{\mathcal{Q}}\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1} - \grave{H}_{\mathcal{T}_{\mathcal{Q}}\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1} \\ & + \breve{G}'_{\mathcal{T}_{\mathcal{Q}}\mathcal{O}}(I - \breve{G}'_{\mathcal{OO}})^{-1}\left(\breve{G}_{\mathcal{O}\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1} - \grave{H}_{\mathcal{OB}}\breve{G}_{\mathcal{B}\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\right)\end{aligned} \tag{4.71}$$

$$\bar{R}_{\mathcal{T}_{\mathcal{O}}\mathcal{Z}_r} = [(I - \breve{G}'_{\mathcal{OO}})^{-1}]_{(\mathcal{T}_{\mathcal{O}},:)}(\breve{G}_{\mathcal{O}\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1} - \grave{H}_{\mathcal{OB}}\breve{G}_{\mathcal{B}\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}). \tag{4.72}$$

Now from (4.30), $\bar{G}_{\mathcal{T}_{\mathcal{Q}}j} = (I - \text{diag}(\breve{G}''_{\mathcal{T}_{\mathcal{Q}}\mathcal{T}_{\mathcal{Q}}}))^{-1}\breve{G}''_{\mathcal{T}_{\mathcal{Q}}j}$ and from (4.23) we have $\bar{G}_{\mathcal{T}_{\mathcal{O}}j} = [(I - \breve{G}'_{\mathcal{OO}})^{-1}]_{(\mathcal{T}_{\mathcal{O}},:)}\breve{G}'_{\mathcal{O}j}$ and $\bar{G}_{\mathcal{O}j} = (I - \breve{G}'_{\mathcal{OO}})^{-1}\breve{G}'_{\mathcal{O}j}$. From (4.25), we have $\breve{G}''_{\mathcal{T}_{\mathcal{Q}}j} = \breve{G}'_{\mathcal{T}_{\mathcal{Q}}j} + \breve{G}'_{\mathcal{T}_{\mathcal{Q}}\mathcal{O}}\bar{G}_{\mathcal{O}j}$ where using (4.17) we have $\breve{G}'_{\mathcal{T}_{\mathcal{Q}}j} = \breve{G}_{\mathcal{T}_{\mathcal{Q}}j} - \grave{H}_{\mathcal{T}_{\mathcal{Q}}\mathcal{B}}\breve{G}_{\mathcal{B}j}$, $\breve{G}'_{\mathcal{T}_{\mathcal{O}}j} = \breve{G}_{\mathcal{T}_{\mathcal{O}}j} - \grave{H}_{\mathcal{T}_{\mathcal{O}}\mathcal{B}}\breve{G}_{\mathcal{B}j}$ and $\breve{G}'_{\mathcal{O}j} = \breve{G}_{\mathcal{O}j} - \grave{H}_{\mathcal{OB}}\breve{G}_{\mathcal{B}j}$. Now expanding the terms $\breve{G}_{\mathcal{T}_{\mathcal{Q}}j}, \breve{G}_{\mathcal{T}_{\mathcal{O}}j}, \breve{G}_{\mathcal{B}j}, \breve{G}_{\mathcal{O}j}$ using (4.54) we can write,

$$\begin{aligned}\bar{G}_{\mathcal{T}_{\mathcal{Q}}j} = {} & (I - \text{diag}(\breve{G}''_{\mathcal{T}_{\mathcal{Q}}\mathcal{T}_{\mathcal{Q}}}))^{-1}(\breve{G}_{\mathcal{T}_{\mathcal{Q}}j} + \breve{G}_{\mathcal{T}_{\mathcal{Q}}\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\breve{G}_{\mathcal{Z}_r j} & (4.73) \\ & - \grave{H}_{\mathcal{T}_{\mathcal{Q}}\mathcal{B}}(\breve{G}_{\mathcal{B}j} + \breve{G}_{\mathcal{B}\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\breve{G}_{\mathcal{Z}_r j}) + \breve{G}'_{\mathcal{T}_{\mathcal{Q}}\mathcal{O}}(I - \breve{G}'_{\mathcal{OO}})^{-1} \\ & (\breve{G}_{\mathcal{O}j} + \breve{G}_{\mathcal{O}\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\breve{G}_{\mathcal{Z}_r j} \\ & - \grave{H}_{\mathcal{OU}}(\breve{G}_{\mathcal{U}j} + \breve{G}_{\mathcal{U}\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\breve{G}_{\mathcal{Z}_r j})), & (4.74)\end{aligned}$$

$$\bar{G}_{\mathcal{T}_{\mathcal{O}}j} = [(I - \breve{G}'_{\mathcal{OO}})^{-1}]_{(\mathcal{T}_{\mathcal{O}},:)}(\breve{G}_{\mathcal{O}j} + \breve{G}_{\mathcal{O}\mathcal{Z}_r}(I - \breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\breve{G}_{\mathcal{Z}_r j}$$

$$-\grave{H}_{\mathcal{O}\mathcal{B}}(\breve{G}_{\mathcal{B}j}+\breve{G}_{\mathcal{B}\mathcal{Z}_r}(I-\breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}G_{\mathcal{Z}_rj})). \tag{4.75}$$

Looking into (4.71) and (4.73) we have,

$$\begin{aligned}\bar{G}_{\mathcal{T}_\mathcal{Q}j} &= \bar{R}_{\mathcal{T}_\mathcal{Q}\mathcal{Z}_r}\breve{G}_{\mathcal{Z}_rj}+(I-\text{diag}(\breve{G}''_{\mathcal{T}_\mathcal{Q}\mathcal{T}_\mathcal{Q}}))^{-1}\Big(\breve{G}_{\mathcal{T}_\mathcal{Q}j}-\grave{H}_{\mathcal{T}_\mathcal{Q}\mathcal{B}}\breve{G}_{\mathcal{B}j}\\ &\quad +\breve{G}'_{\mathcal{T}_\mathcal{Q}\mathcal{O}}(I-\breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}(\breve{G}_{\mathcal{O}j}-\grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}j})\Big).\end{aligned} \tag{4.76}$$

Similarly, looking into (4.72) and (4.75) we have

$$\bar{G}_{\mathcal{T}_\mathcal{O}j}=\bar{R}_{\mathcal{T}_\mathcal{O}\mathcal{Z}_r}\breve{G}_{\mathcal{Z}_rj}+[(I-\breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}]_{(\mathcal{T}_\mathcal{O},:)}(\breve{G}_{\mathcal{O}j}-\grave{H}_{\mathcal{O}\mathcal{B}}\breve{G}_{\mathcal{B}j}). \tag{4.77}$$

Condition 4b in Property 4.1 ensures that $\breve{G}_{\mathcal{T}_\mathcal{Q}j} = 0$, **condition 7 in Property 4.1** ensures that $\breve{G}_{\mathcal{B}j} = 0$ and **condition 8 in Property 4.1** ensures that $\breve{G}_{\mathcal{O}j} = 0$. Therefore we can write (4.76) and (4.77) as,

$$\bar{G}_{\mathcal{T}j}=\bar{R}_{\mathcal{T}\mathcal{Z}_r}\breve{G}_{\mathcal{Z}_rj}. \tag{4.78}$$

Condition 6 in Property 4.1 ensures that a left inverse of $\bar{R}_{\mathcal{T}\mathcal{Z}_r}$ exists. Then $\breve{G}_{\mathcal{Z}_rj} = \bar{R}^{\dagger}_{\mathcal{T}\mathcal{Z}_r}\bar{G}_{\mathcal{T}j}$. So we write (4.70) as $\breve{G}'_{jj} = (1-\breve{G}'_{jj})\bar{R}_{j\mathcal{Z}_r}\bar{R}^{\dagger}_{\mathcal{T}\mathcal{Z}_r}\bar{G}_{\mathcal{T}j}$. Thus we get $(1-\breve{G}'_{jj})^{-1} = \Big(1-\bar{R}_{j\mathcal{Z}_r}\bar{R}^{\dagger}_{\mathcal{T}\mathcal{Z}_r}\bar{G}_{\mathcal{T}j}(1+\bar{R}_{j\mathcal{Z}_r}\bar{R}^{\dagger}_{\mathcal{T}\mathcal{Z}_r}\bar{G}_{\mathcal{T}j})^{-1}\Big)^{-1}$.

Obtaining the expression of $\breve{G}_{j\mathcal{Z}_r}(I-\breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\breve{G}_{\mathcal{Z}_ri}-\grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{Z}_r}(I-\breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\breve{G}_{\mathcal{Z}_ri}$**:** In (4.64), we now look into $\breve{G}'_{ji} = G_{ji}+\breve{G}_{j\mathcal{Z}_r}(I-\breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\breve{G}_{\mathcal{Z}_ri}-\grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{Z}_r}(I-\breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\breve{G}_{\mathcal{Z}_ri}$ to extract the target module G_{ji}. From (4.69), we know that

$$\breve{G}_{j\mathcal{Z}_r}(I-\breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\breve{G}_{\mathcal{Z}_ri}-\grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{Z}_r}(I-\breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\breve{G}_{\mathcal{Z}_ri} = (I-\breve{G}'_{jj})\bar{R}_{j\mathcal{Z}_r}\breve{G}_{\mathcal{Z}_ri}. \tag{4.79}$$

We already know the expression for $(I-\breve{G}'_{jj})^{-1}$. Therefore, $\breve{G}_{j\mathcal{Z}_r}(I-\breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}G_{\mathcal{Z}_ri}-\grave{H}_{j\mathcal{B}}\breve{G}_{\mathcal{B}\mathcal{Z}_r}(I-\breve{G}_{\mathcal{Z}_r\mathcal{Z}_r})^{-1}\breve{G}_{\mathcal{Z}_ri} = C^{-1}_{jj}\bar{R}_{j\mathcal{Z}_r}\breve{G}_{\mathcal{Z}_ri}$. Now we differentiate two different cases to get $G_{\mathcal{Z}_ri}$: when $i \notin \mathcal{T}_\mathcal{Q}$ and when $i \in \mathcal{T}_\mathcal{Q}$. When $i \notin \mathcal{T}_\mathcal{Q}$, following the similar reasoning for $\breve{G}_{\mathcal{Z}_rj}$, we have $\breve{G}_{\mathcal{Z}_ri} = \bar{R}^{\dagger}_{\mathcal{T}\mathcal{Z}_r}\bar{G}_{\mathcal{T}i}$ provided **condition 2, 5 and 6, 7, 8 in Property 4.1** are satisfied. When $i \in \mathcal{T}_\mathcal{Q}$, then $i \in \mathcal{Q}$. Now in $\bar{G}_{\mathcal{T}i}$ column matrix we have an element $\bar{G}_{ii} = (I-\breve{G}''_{ii})^{-1}(\breve{G}''_{ii}-\text{diag}(\breve{G}''_{ii}))$. When **condition 3 in Property 4.1** is satisfied, from (4.33) we have $\bar{R}_{ii} = (1-\breve{G}''_{ii})^{-1}$. Therefore $(I-\breve{G}''_{ii})^{-1}\text{diag}(\breve{G}''_{ii}) = \bar{R}_{ii}(1-\bar{R}^{-1}_{ii})$. Let C_{ii} be a column matrix with every element as zero except the element corresponding to node w_i which is $\bar{R}_{ii}(1-\bar{R}^{-1}_{ii})$. Therefore, $\bar{G}_{\mathcal{T}i} = \bar{R}_{\mathcal{T}\mathcal{Z}_r}G_{\mathcal{Z}_ri}-C_{ii}$. This gives us $C^{-1}_{jj}\bar{R}_{j\mathcal{Z}_r}\breve{G}_{\mathcal{Z}_ri} = C^{-1}_{jj}\bar{R}_{j\mathcal{Z}_r}\bar{R}^{\dagger}_{\mathcal{T}\mathcal{Z}_r}(\bar{G}_{\mathcal{T}i}+C_{ii})$.

Since, $\mathcal{Z}_r$ will be a subset of $\mathcal{K}$, we can replace $R_{j\mathcal{Z}_r}, \bar{R}_{\mathcal{T}\mathcal{Z}_r}, \bar{R}_{ii}, \bar{R}_{jj}$ as $J_{j\mathcal{Z}_r}, \bar{J}_{\mathcal{T}\mathcal{Z}_r}, \bar{J}_{ii}, \bar{J}_{ii}$ respectively. Now, we have the expression of every element in (4.64) using elements of (4.4), except the target module G_{ji}. Thus, we can extract the target module using the expression in the result of the theorem. ■

5 Chapter

Path-based conditions for data-informativity

For consistent or minimum variance estimation of a single module in a dynamic network, a predictor model has to be chosen with selected inputs and outputs, composed of a selection of measured node signals and possibly external excitation signals. The predictor model has to be chosen in such a way that consistent estimation of the target module is possible, under the condition that we have data-informativity for the considered predictor model set. Consistent estimation of the target module is typically obtained if we follow a direct method or generalized method of identification and predictor model selection, characterized by the property that measured node signals and possibly excitation signals are the prime predictor input signals. In this chapter the concept of data-informativity for network models will be formalized, and for the direct method and the generalized method the required data-informativity conditions will be specified in terms of path-based conditions on the graph of the network model, guaranteeing data-informativity in a generic sense, i.e. independent on numerical values of the network transfer functions concerned.

5.1 Introduction

The direct, generalized, and indirect identification methods typically start from a limited set of measured node signals and a selected set of measured external

This chapter is based on the preliminary work: P.M.J. Van den Hof and K.R. Ramaswamy, "Path-based data-informativity conditions for single module identification in dynamic networks", in *Proc. 59th IEEE Conf. Decision and Control (CDC)*, Jeju Island, Republic of Korea, 14-18 December 2020, pp. 4354-4359.

excitation signals (that are used in the predictor model), to determine whether a consistent and/or minimum variance estimate of the target module can be obtained. For all of these methods, data-informativity condition need to be satisfied for arriving at consistent module estimates. This condition is typically formulated as positive definite condition on a spectrum of signals in the dynamic network. While for indirect methods these conditions can typically be phrased in terms of persistence of excitation conditions on external excitation signals, see e.g. [54], for direct methods (and the generalized method) they are typically formulated in terms of a spectral condition on node signals and external excitation signals in the network, and thereby harder to interpret for the user who has to set up an experiment. This has also been addressed in [53] where it has been highlighted that the typical spectral conditions will often be conservative in the case of modules with finite model order.

In this chapter we are going to address the situation of the direct method and the generalized method discussed in Chapter 3 and 4, and we are going to reformulate the data-informativity conditions for these methods in terms of excitation conditions on the external excitation signals, together with path-based conditions on the topology of the network model set. In this way the data-informativity conditions become verifiable by the user, rather than remaining implicit as in condition (b) of Theorem 4.1.

We will highlight the different options for selecting predictor models in Section 5.2. In Section 5.3 data-informativity conditions are specified, for which path-based conditions are being derived in Section 5.4. The results are illustrated with examples. The proofs of all technical results are collected in the appendix.

5.2 Network estimation setup

We can distinguish three main different prediction error approaches for addressing the single module identification problem, where the target module is indicated by G_{ji}.

1. A *direct method*, that is based on selecting a particular set of predictor input node signals w_k, $k \in \mathcal{D}$, a set of predictor input external excitation signals ($u_k, k \in \mathcal{P} \cup \mathcal{K}$) and a set of predicted output signals w_ℓ, $\ell \in \mathcal{Y}$, with $i \in \mathcal{D}$, $j \in \mathcal{Y}$, and estimating a dynamic model based on a prediction error:

$$\varepsilon(t,\theta) = \bar{H}(q,\theta)^{-1}[w_{\mathcal{Y}}(t) - \bar{G}(q,\theta)w_{\mathcal{D}}(t) - \bar{J}(q,\theta)u_{\mathcal{K}}(t) - \bar{S}(q)u_{\mathcal{P}}(t)], \quad (5.1)$$

where $\bar{G}(q,\theta), \bar{J}(q,\theta)$ and $\bar{H}(q,\theta)$ are parametrized transfer function matrices and $\bar{S}(q)$ is a selection (binary) matrix. The target module is then embedded in the model $\bar{G}(q,\theta)$, and the objective is to estimate the target module consistently and possibly with minimum variance. The direct method does not require any *post-processing* and hence $\bar{G}^0_{ji} = G^0_{ji}$.

2. An *indirect method*, that is based on selecting a particular set of external excitation signals r_k, $k \in \mathcal{D}$, and a set of predicted node signals w_ℓ, $\ell \in \mathcal{Y}$,

that are used in a predictor model, leading to

$$\varepsilon_{\mathcal{Y}}(t,\theta) \quad = \quad w_{\mathcal{Y}}(t) - \bar{T}_{\mathcal{Y}}(q,\theta) r_{\mathcal{D}}(t) \tag{5.2}$$

Since $\bar{T}_{\mathcal{Y}}$ reflects a mapping from external signals ($r_{\mathcal{D}}$) to internal signals ($w_{\mathcal{Y}}$), a processing step is necessary to recover the target module G_{ji} from an estimated $\bar{T}_{\mathcal{Y}}$. Consistency of the target module estimate is the typical objective. Different variations of indirect methods exist, including two-stage and instrumental variable (IV) methods.

3. A *generalized method*, that is based on the same predictor model and prediction error as the direct method, and the objective is to estimate the target module consistently. The generalized method requires *post-processing* using the estimated models in order to recover the target module.

In this chapter, we will primarily focus on the direct method and the generalized method. For these methods to arrive at a consistent estimate of the target module, there are two prime conditions that need to be satisfied.

1. A predictor model needs to be chosen, on the basis of which it is possible to reconstruct the target module G_{ji} from the estimated objects $\bar{G}, \bar{J}$ and $\bar{H}$. The predictor model (5.1) is determined by the selection of signals that appear in $w_{\mathcal{Y}}, w_{\mathcal{D}}, u_{\mathcal{P}}$ and $u_{\mathcal{K}}$.

2. For the chosen predictor model, the data appearing in this model should be sufficiently informative so as to guarantee that consistent estimates of the objects $\bar{G}$, $\bar{J}$, $\bar{H}$ are obtained (refer to condition (b) of Theorem 4.1 and condition (b) of Theorem 3.2).

The first aspect is covered in Chapter 3 and Chapter 4. In the sequel of this chapter, we will focus on the data-informativity aspects as mentioned in the second aspect.

5.3 Data-Informativity

5.3.1 Introduction and definition

We consider an estimation setup on the basis of the network equations

$$w_{\mathcal{Y}}(t) = \bar{G}(q) w_{\mathcal{D}}(t) + \bar{J}(q) u_{\mathcal{K}}(t) + \bar{S}(q) u_{\mathcal{P}}(t) + \bar{H}(q) \xi_{\mathcal{Y}}(t) \tag{5.3}$$

with $w_{\mathcal{Y}}$, $w_{\mathcal{D}}$, $u_{\mathcal{K}}$, $u_{\mathcal{P}}$ selected node- and excitation signals and $\xi_{\mathcal{Y}}$ a stationary white noise process.

The one-step ahead predictor for (5.3) is uniquely defined through

$$\hat{w}_{\mathcal{Y}}(t) \quad := \quad \mathbb{E}\{w_{\mathcal{Y}}(t) | w_{\mathcal{Y}}^{t-1}, w_{\mathcal{D}}^{t}, u_{\mathcal{P}}^{t}, u_{\mathcal{K}}^{t}\} = W(q) z(t) \tag{5.4}$$

with the predictor filter given by

$$W(q) := \left[(1 - \bar{H}(q)^{-1}) \quad \bar{H}(q)^{-1}\bar{G}(q) \quad \bar{H}(q)^{-1}\bar{J}(q) \quad \bar{H}(q)^{-1}\bar{S}(q)\right] \tag{5.5}$$

and

$$z(t) := \begin{bmatrix} w_{\mathcal{Y}}(t) \\ w_{\mathcal{D}}(t) \\ u_{\mathcal{K}(t)} \\ u_{\mathcal{P}(t)} \end{bmatrix}. \tag{5.6}$$

In line with the corresponding definitions in the prediction error literature ([77], Definition 8.1), we can now define the notion of data-informativity for the related network predictor model.

Definition 5.1 *Consider a set of network signals contained in z and a network predictor model*

$$\hat{w}_{\mathcal{Y}}(t, \theta) = W(q, \theta) z(t)$$

for a parametrized set of models

$$\mathcal{M} := (\bar{G}(q, \theta), \bar{J}(q, \theta), \bar{S}(q), \bar{H}(q, \theta))_{\theta \in \Theta}.$$

Then a quasi-stationary data set $Z^{\infty} := \{z(t)\}_{t=0,\cdots}$ with $z(t)$ defined in (5.6) is informative enough with respect to the model set $\mathcal{M}$ if, for any two predictor filters $W_1(q)$ and $W_2(q)$ in the parameterized model set,

$$\bar{\mathbb{E}}[(W_1(q) - W_2(q))z(t)]^2 = 0$$

implies that $W_1(e^{i\omega}) \equiv W_2(e^{i\omega})$ for almost all ω. □

In line with ([77], Definition 8.2), we formulate:

Definition 5.2 *A quasi-stationary data set Z^{∞} is informative if it is informative enough with respect to the model set $\mathcal{L}^*$, consisting of all linear time-invariant models.*

And in line with ([77], Definition 13.2):

Definition 5.3 *A quasi-stationary signal z is said to be persistently exciting if $\Phi_z(\omega) > 0$ for almost all ω.*

The essential difference with the classical definitions in [77] is in the composition of the signal vector $z(t)$, being composed according to (5.6).

5.3.2 Classical open-loop case

The classical open-loop case can be represented by the situation that in the predictor model, the predictor input is $w_{\mathcal{D}} = r$. In this case

$$z(t) := \begin{bmatrix} w_{\mathcal{Y}}(t) \\ w_{\mathcal{D}}(t) \end{bmatrix}.$$

The well known sufficient condition for data-informativity is now [77]:

$$\Phi_z(\omega) > 0 \quad \text{for almost all } \omega. \tag{5.7}$$

For estimating *finite-dimensional* models, this sufficient condition can be further relaxed[1] to be satisfied for a sufficient number of frequencies ω. The signal vector z contains both predictor inputs and predictor outputs. Since there are output disturbances on $w_{\mathcal{Y}}$ that are uncorrelated to $w_{\mathcal{D}}$, the informativity condition simplifies to the condition that $w_{\mathcal{D}}$ should be persistently exciting.

5.3.3 Classical closed-loop case: direct method

The direct method for closed-loop systems is characterized by the situation that in the predictor model

- $w_{\mathcal{Y}}$ and $w_{\mathcal{D}}$ are distinct signals;
- $u_{\mathcal{P}}$ and $u_{\mathcal{K}}$ are not included in the predictor;
- $w_{\mathcal{D}}$ may depend on the present and past samples of $w_{\mathcal{Y}}$ (feedback).

It follows that $z(t) := \begin{bmatrix} w_{\mathcal{Y}}(t) \\ w_{\mathcal{D}}(t) \end{bmatrix}$ and the "open-loop" results of [77] still apply, i.e. the informativity condition of the data is represented by the condition (5.7).

5.3.4 The network case: local direct method and generalized method

When applying the direct identification method or generalized method in the network case, a predictor model is constructed with node signals $w_{\mathcal{D}}$ and external excitation signals $(u_{\mathcal{P}}, u_{\mathcal{K}})$ as predictor inputs and $w_{\mathcal{Y}}$ as predicted outputs. According to the results in Chapter 3 and Chapter 4, we end up in a multi-output predictor model, as schematically indicated in Figure 5.1. In this setting we distinguish:

- $w_{\mathcal{Y}} = \begin{bmatrix} w_{\mathcal{O}} \\ w_{\mathcal{Q}} \end{bmatrix}$; $w_{\mathcal{D}} = \begin{bmatrix} w_{\mathcal{U}} \\ w_{\mathcal{Q}} \end{bmatrix}$;

[1] For the network case and considering MISO models this is also addressed in [53].

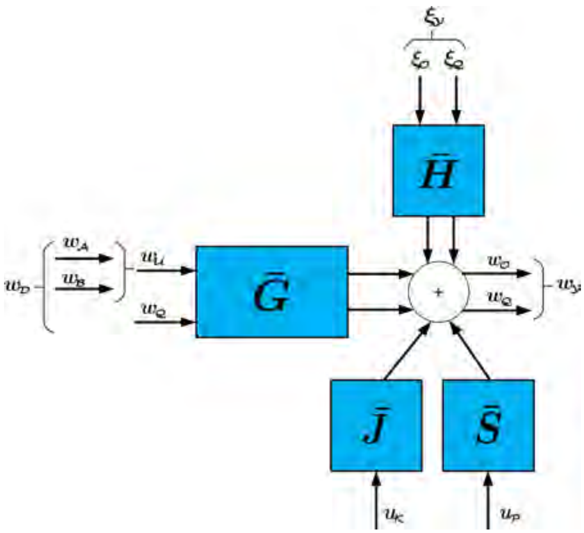


Figure 5.1: Predictor model for local direct method and generalized method; the set of node signals $w_{\mathcal{Q}}$ appears both at the input and at the output of the predictor model.

- For the local direct method in Chapter 3 : $w_{\mathcal{O}} = w_j$ or $w_{\mathcal{O}}$ is void if w_j is present in $w_{\mathcal{Q}}$;

- $u_{\mathcal{P}}$ contains those measured external excitation signals in u that add directly to measured outputs w_k, $k \in \mathcal{Y}$, i.e. for which $\bar{S}$ is a binary (selection) matrix with known elements, indicating which output signals are excited by signals $u_{\mathcal{P}}$ (see Proposition 4.3);

- $u_{\mathcal{K}}$ contains those measured external excitation signals in u that dynamically affect the measured outputs w_k, $k \in \mathcal{Y}$, i.e. for which $\bar{J}$ is a parameterized matrix that contains the dynamics indicating which output signals are excited by signals $u_{\mathcal{K}}$ (see Section 4.5.1).

Inputs and outputs are allowed to share some common signals, while all node signals are allowed to depend on each other's (present and) past. According to the consistency results for the local direct method and the generalized method in Chapter 4, the data-informativity conditions now become:

$$\Phi_{\kappa}(\omega) > 0 \quad \text{for almost all } \omega, \tag{5.8}$$

with

$$\kappa(t) := \begin{bmatrix} w_{\mathcal{D}}(t) \\ \xi_{\mathcal{Y}}(t) \\ u_{\mathcal{K}}(t) \end{bmatrix}$$

and $\xi_{\mathcal{Y}}(t)$ the white noise innovation process that relates to output $w_{\mathcal{Y}}(t)$ in (5.3). In the vector signal κ we collect all the measured node signals that appear as predictor input, all the measured external excitation signals that appear as predictor input which contribute to the output through parameterized transfer function matrix $\bar{J}$, and the (external) noise terms $\xi_{\mathcal{Y}}$. The spectrum condition on $\kappa(t)$ can then be interpreted as a condition that requires a full rank spectrum of $w_{\mathcal{D}}$, while using all external signals in the network *except* $(\xi_{\mathcal{Y}}, u_{\mathcal{K}})$. In other words, $(\xi_{\mathcal{Y}}, u_{\mathcal{K}})$ can not be used for the "excitation" of the signals $w_{\mathcal{D}}$, but this excitation has to come from other external signals in the network. This can be interpreted as follows. $u_{\mathcal{K}}$ and $\xi_{\mathcal{Y}}$ are used for estimating $\bar{J}$ and $\bar{H}$ respectively. Hence, we need other external signals in the network for estimating $\bar{G}$. This mechanism is going to be further elaborated upon in the next Section.

5.4 Path-based conditions for data-informativity

5.4.1 General results

The condition (5.8) for data-informativity in the direct method and in the generalized method is compactly formulated, but it is actually implicit and hard to ensure for the situation of a dynamic network with given topology and unknown dynamics. It would be very attractive to formulate this condition in terms of properties and locations of the external signals in the network (i.e. u (or r) and e) together with topological conditions on the interconnection structure in the network models that we consider. In order to achieve this objective, we consider the following Lemma:

Lemma 5.1 *Let $x(t) \in \mathbb{R}^m$ be a quasi-stationary signal that is persistently exciting, and let $F(z) \in \mathbb{R}(z)^{p\times m}$ be the proper rational transfer function of a stable filter. Then the signal $y(t) = F(q)x(t)$ is persistently exciting if and only if filter $F(z)$ has rank p over the field of rational functions.* □

Proof: Collected in the Appendix. ■

If we apply this Lemma with x-signals being the external signals u, e, and y-signals being selected node signals w in the network, then the row rank of the considered transfer function $(u, e) \rightarrow y$ would need to be evaluated in order to make a statement about data-informativity. In line with the idea of introducing a generic form of identifiability [65], i.e. independent of particular numerical values of coefficients, we can use the same generic type of result for data-informativity, based on the results of [131].

Proposition 5.1 *Consider the situation of Lemma 5.1. The property that $y(t)$ is persistently exciting holds generically*[a] *if in the dynamic network there are p vertex-disjoint paths between the nodes x and y. This is denoted by $b_{x\rightarrow y} =$*

p.

[a] Generically has to be considered here in terms of a Lebesgue measure 0 of the vector of coefficient values of the rational transfer functions in all modules of the network.

So, a persistently exciting "input" signal x and a sufficient number of vertex-disjoint paths between x and y, will generically provide a persistently exciting "output" signal y. This result can be used to translate persistence of excitation conditions on node signals, to persistence of excitation conditions on external network signals.

5.4.2 Path-based conditions

The result on vertex-disjoint paths, as formulated in Proposition 5.1 can now be applied to the particular situation of condition (5.8). In this step the consequence of having the white noise signal $\xi_{\mathcal{Y}}$ in the condition (5.8) needs to be translated to conditions on signals in the original network (2.2). We now formulate a path-based condition for verifying data-informativity.

Theorem 5.1 *Consider a dynamic network with external signals u and e, and let $u_{\mathcal{K}}$ be the u-signals that appear as predictor input in the setting of the local direct method and the generalized method, satisfying the conditions of Proposition 4.4. Let $\bar{\mathcal{K}} = \mathcal{L} \backslash \mathcal{K} = \{\mathcal{P} \cup \mathcal{A} \cup \bar{\mathcal{Z}}\}$, where $\bar{\mathcal{Z}}$ denotes the indices of excitation signals in u (i.e. $u_\ell, \ell \in \bar{\mathcal{Z}}$) that satisfies the following:*

- *all paths from w_ℓ to $w_k, k \in \mathcal{Y}$ pass through a node in $w_{\mathcal{Q} \cup \mathcal{U}}$;*
- *there exist no direct or unmeasured paths from w_ℓ to any $w_m, m \in \mathcal{B}$.*

Consider the signal vector

$$\eta(t) := \begin{bmatrix} u_{\bar{\mathcal{K}}} \\ e^{\mathcal{U}} \end{bmatrix}, \quad \textit{with}$$

$e^{\mathcal{U}}$: any e-signal that has a direct or unmeasured path to a node signal w_k, $k \in \mathcal{U}$.

Then the transfer function from (u, e) to κ generically has full row rank if there are $n_{\mathcal{D}}$ vertex disjoint paths between external signals η and $w_{\mathcal{D}}$.

Proof: The proof is added in Appendix 5.B. ■

It can be inferred that:

- the u signals that add directly to the outputs $w_k, k \in \mathcal{Y}$ with a selection matrix $\breve{S}(q)$ (i.e. $u_{\mathcal{P}}$) consisting of known elements contribute to the data-informativity;

- the u signals whose effect on the outputs $w_k, k \in \mathcal{Y}$ that are incorporated in the term $\bar{G}w_{\mathcal{D}}$ (i.e. $u_{\mathcal{A}}$ and $u_{\bar{\mathcal{Z}}}$) contribute to the data-informativity;
- the u signals that dynamically affect the outputs $w_k, k \in \mathcal{Y}$ with the matrix $\bar{J}(q)$ (i.e. $u_{\mathcal{K}}$) that contains unknown dynamics does not contribute to the data-informativity. These excitation signals are effectively used to estimate the dynamics in $\bar{J}$;
- $u_{\mathcal{B}} \subseteq u_{\mathcal{K}}$ and hence does not contribute to data-informativity;
- only the e signals that have direct or unmeasured path to nodes in $w_{\mathcal{U}}$ contribute to the data-informativity.

As a direct result of Proposition 5.1 we can now formulate the following Corollary:

Corollary 5.1 *The data-informativity condition (5.8) for the local direct method and the generalized method is satisfied if the path-based conditions of Theorem 5.1 are satisfied and the present excitation signals u in the network are persistently exciting.* □

We will illustrate the results of this Section in three examples.

Example 5.1 *Consider a classical closed loop system represented by a two-node network as depicted in Figure 5.2 with v_1 and v_2 being process noises that are correlated, and with $u_1 = r_1$, $u_2 = r_2$. First we consider the situation of having no external excitation signals, $r_1 = r_2 = 0$. The objective is to identify the target module G_{21}. We use the local direct method. We select w_1 as input and w_2 as output of our predictor model, but due to the correlation between v_1 and v_2, we need to include w_1 also as an output. As a result $w_{\mathcal{Y}} = \{w_1, w_2\}$, $w_{\mathcal{O}} = w_o = \{w_2\}$ and $w_{\mathcal{D}} = w_{\mathcal{Q}} = \{w_1\}$. Then $\mathcal{U}$ is void. In order to satisfy the data informativity condition according to Theorem 5.1, we need to consider vector $\eta(t)$. Since $\mathcal{U}, \mathcal{Z}$ is void and $u_{\mathcal{P}}$ is not present, η is an empty vector, indicating that there are no external signals available for exciting $w_{\mathcal{D}}$. Therefore the data-informativity condition can not be satisfied. The two noise signals e_1 and e_2 constitute the innovation process $\xi_{\mathcal{Y}}$ and according to the definition of $\kappa(t)$ in (5.8) cannot be used to excite $w_{\mathcal{D}}$. These noise signals are effectively used to estimate the 2×2 noise model.*

Adding an external excitation signal u_1 will not lead to a signal in $u_{\mathcal{P}}$ since the loop through w_1 passes only through $w_{\mathcal{O}} = w_o = w_2$, and therefore the condition for $w_\ell \in w_{\mathcal{Q}}$ in Proposition 4.3 is not satisfied. In the predictor model $w_1 \to (w_1, w_2)$, u_1 cannot effectively be used for excitation due to the fact that G_{12} is not modelled, leading to the situation that in the model the contribution of u_1 to w_1 is actually represented by $(1 - G_{12}G_{21})^{-1}u_1$, thus not satisfying the unit transfer that is required for a signal in $u_{\mathcal{P}}$.

Similarly adding an external excitation signal u_2 will not lead to a signal in

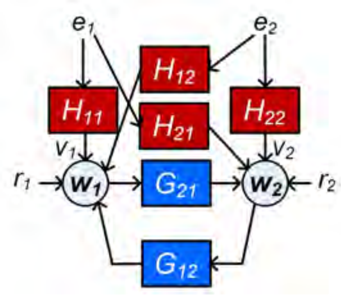


Figure 5.2: Classical closed loop example with two node signals and disturbances v_1 and v_2 being correlated.

$u_{\mathcal{P}}$ since there is a direct path from $w_{\mathcal{O}} = w_o = w_2$ to $w_{\mathcal{Q}} = w_1$, and thus the condition for $w_\ell \in w_{\mathcal{O}}$ in Proposition 4.3 is not satisfied. In the predictor model $w_1 \to (w_1, w_2)$, u_2 cannot effectively be used for excitation due to the fact that G_{12} is not modeled, leading to the situation that in the model the contribution of u_2 to w_1 is actually represented by $(1-G_{12}G_{21})^{-1}G_{12}u_2$, thus not satisfying the unit transfer that is required for a signal in $u_{\mathcal{P}}$. However, if we include w_2 also as input and model G_{12} with predictor model $(w_1, w_2) \to (w_1, w_2)$, then $w_{\mathcal{D}}$ changes to $w_{\mathcal{D}} = w_{\mathcal{Q}} = \{w_1, w_2\}$. Then we need two (vertex disjoint) paths from $\eta = u_{\mathcal{P}}$ to $w_{\mathcal{D}}$. With predictor model $(w_1, w_2) \to (w_1, w_2)$, both u_1 and u_2 lead to a signal in $u_{\mathcal{P}}$ and therefore we need two external excitation signals r_1 and r_2 for achieving data-informativity. This result is in agreement with the observations in [126].

Example 5.2 *Consider the three node network depicted in Figure 5.3 with v_1 and v_3 being disturbance signals that are correlated, and $u_1 = r_1, u_2 = r_2, u_3 = r_3$. First we consider the situation of having no external excitation signals, $r_1 = r_2 = r_3 = 0$. The objective is to identify the target module G_{12} using the local direct method. According to the local direct method, we have multiple ways to choose the predictor model. Following the full input case, we choose $w_{\mathcal{Y}} = w_o = w_1$, $w_{\mathcal{A}} = w_2$, and then we choose $w_{\mathcal{B}} = w_3$ in order to block the effect of the confounding variable e_3 for the estimation problem $w_2 \to w_1$. In this setup $w_{\mathcal{Q}}$ is void and $w_{\mathcal{D}} = w_{\mathcal{U}} = \{w_2, w_3\}$. The data-informativity condition of Theorem 5.1 now requires two vertex disjoint paths between $\{e_2, e_3\}$ and $\{w_2, w_3\}$. As this can simply be verified from the graph, the data-informativity condition is satisfied without any need for external excitation signals.*

When choosing an alternative predictor model, e.g. according to the minimum input case algorithm, we choose $w_{\mathcal{Y}} = \{w_1, w_2\}$ and $w_{\mathcal{Q}} = w_2$, i.e. we model w_2 as output also, in order to deal with the confounding variable e_3 for the estimation problem $w_2 \to w_1$. In this setup $w_{\mathcal{U}} = w_{\mathcal{A}\cup\mathcal{B}}$ is void. In order to satisfy the data informativity condition according to Theorem 5.1,

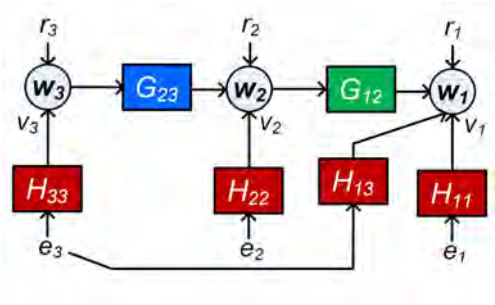


Figure 5.3: A three node network example.

we need a path from signal $\eta(t)$ to $w_{\mathcal{D}} = w_2$. Since $\mathcal{U}$ is void and $u_{\mathcal{P}}, u_{\bar{\mathcal{Z}}}$ is not present, η is an empty vector, indicating that there are no external signals available for exciting $w_{\mathcal{D}}$. Therefore the data-informativity condition can not be satisfied. Adding external signals u_1 or u_2 will lead to a signal in $\eta = u_{\mathcal{P}}$. But, when $\eta = u_1$ we do not satisfy the data informativity condition since we do not have a path from signal $\eta = u_1$ to $w_{\mathcal{D}} = w_2$. When adding $u_2 = r_2$ as external signal, we satisfy the data informativity condition since the path-based condition is satisfied. An external signal $u_{\mathcal{Z}} = u_3 = r_3$ cannot contribute to η because it does not belong to $u_{\bar{\mathcal{Z}}}$ (see Definition 4.2) since there is an unmeasured path from u_3 to w_2 which is in $w_{\mathcal{Y}}$. It cannot be used for excitation due to the non-unity transfer to w_2 which needs to be estimated, and hence does not provide data informativity for the chosen predictor model.

Example 5.3 *Consider the four-node network as depicted in Figure 5.4 with $v_1 = e_1, v_2 = e_2, v_3 = e_3, v_4 = e_4$ being process noises that are uncorrelated with each other, and non-measured node signal w_3. The other node signals are measured. First we consider the situation of having no external excitation signals, $u_1 = u_2 = u_3 = u_4 = 0$. The objective is to identify the target module G_{21} with $j = 2$ and $i = 1$. Since w_3 is non-measured, the parallel path/loop condition 3.1 required for target module invariance is not satisfied. Hence, we cannot use the local direct method and we resort to the generalized method. We select w_1 as input and w_2 as output of our predictor model. Since w_3 is non-measured, the parallel path $w_1 \to w_3 \to w_2$ violates the condition 3.1. This violation is handled in generalized method by exciting the parallel path with u_3, including u_3 as predictor input, and by including the descendant w_4 from the excited node w_3 as predicted output. As a result $w_{\mathcal{Y}} = \{w_2, w_4\}$, $w_{\mathcal{O}} = \{w_2, w_4\}$, $w_{\mathcal{Z}_r} = \{w_3\}$, $w_{\mathcal{U}} = w_{\mathcal{A}} = \{w_1\}$ and $w_{\mathcal{D}} = \{w_1\}$. Condition 4.2 requires that there are no direct or unmeasured paths from $w_k, k \in \mathcal{O}\backslash\{j\}$ to $(w_j, w_{\mathcal{B}})$ and also $i \in \mathcal{Q} \cup \mathcal{A}$. For Condition 4.2 to be satisfied, w_4 should also be included as predictor input. As a result $w_{\mathcal{Y}} = \{w_2, w_4\}$, $w_{\mathcal{Q}} = \{w_4\}$, $w_{\mathcal{O}} = \{w_2\}$, $w_{\mathcal{Z}_r} = \{w_3\}$,*

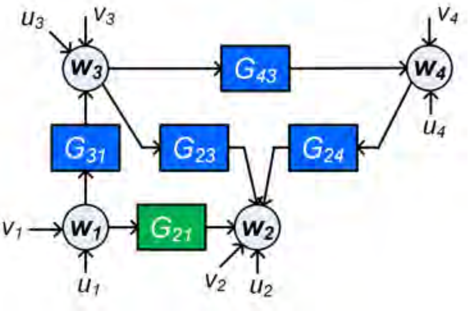


Figure 5.4: A 4-node dynamic network with w_3 non-measured and the noise signals on each node uncorrelated with each other. G_{21} is the target module that needs to be identified.

$w_{\mathcal{U}} = w_{\mathcal{A}} = \{w_1\}$ and $w_{\mathcal{D}} = \{w_1, w_4\}$. u_3 belongs to $u_{\mathcal{K}}$ and not $u_{\bar{\mathcal{Z}}}$ since there is a path from w_3 to w_4 in $w_{\mathcal{Y}}$ that does not pass through a node in $w_{\mathcal{Q}\cup\mathcal{U}}$. In order to satisfy the data informativity condition according to Theorem 5.1, we need to consider vector $\eta(t)$ and we need two vertex disjoint paths from η to $w_{\mathcal{D}}$. Since v_2, v_3 and v_4 does not have path to $w_{\mathcal{U}} = w_1$, they do not belong to $\bar{e}^{\mathcal{U}}$ and cannot contribute to η. These signals are part of $\xi_{\mathcal{Y}}$ and are used to estimate the 2×2 noise model, and hence does not contribute to η. Since $u_{\mathcal{P}}, u_{\mathcal{A}}, u_{\bar{\mathcal{Z}}}$ are not present, $\eta = \bar{e}^{\mathcal{U}} = e_1$, indicating the external signals available for exciting $w_{\mathcal{D}}$. But, when $\eta = e_1$ we do not satisfy the data informativity condition since we need two vertex disjoint paths from η to $w_{\mathcal{D}}$. When adding u_4 as external signal, it will add to $u_{\mathcal{P}}$ and hence we satisfy the data informativity condition since the path-based condition is satisfied. An external signal $u_{\mathcal{P}} = u_2$ cannot contribute to data-informativity since there does not exist a path from u_2 to w_4. Similarly, adding $u_{\mathcal{A}} = u_1$ to η will not lead to two vertex disjoint paths from η to $w_{\mathcal{D}}$ since both signals (i.e., u_1 and v_1) pass through measured node signal $w_1 \in w_{\mathcal{D}}$. u_3 cannot contribute to η because it does not belong to $u_{\bar{\mathcal{Z}}}$ since there is an unmeasured path from u_3 to w_4 which is in $w_{\mathcal{Y}}$. It cannot be used for excitation due to the non-unity transfer to w_4 which needs to be estimated using u_3, and hence does not provide data informativity for the chosen predictor model.

Remark 5.1 *For the Generalized method, the estimate of $\bar{J}(q)$ is required for post-processing and to obtain the estimate of G_{ji}^0. For the direct method, it is possible to consider the predictor model* (5.1) *without $\bar{J}(q, \theta)$. In this case, the $u_{\mathcal{K}}$-signals act as noise signals and hence the signal selection should take into effect the confounding effects of these signals. This setup has been explored in [127].*

5.5 Conclusions

For consistent identification of a single module that is embedded in a dynamic network it is necessary that the signals that constitute the chosen predictor model satisfy data-informativity conditions. We have formalized the concept of data-informativity for a generalized predictor model that is suited for dynamic network modeling, and that allows for signals to appear both as input and as output in a MIMO predictor model. It generalizes all known situations of indirect and direct methods in closed-loop systems and dynamic networks. The conditions for data-informativity have been specified for the local direct and generalized identification method, showing that the conditions can be satisfied generically by requiring persistence of excitation of external signals, together with path-based conditions on the topology of the network model set.

5.6 Related videos

Path-based data-informativity conditions for single module identification in dynamic networks

Paul Van den Hof, Karthik Ramaswamy

59th IEEE Conference on Decision and Control, December 14-18, 2020, Jeju Island, Republic of Korea

Path-based data informativity conditions

Appendices

5.A Proof of Lemma 5.1

The spectral density of the output signal is given by $\Phi_y(\omega) = F(e^{i\omega})\Phi_x(\omega)F(e^{i\omega})^\star$, with $(\cdot)^\star$ the complex conjugate. For each value of ω in $-\pi \leqslant \omega \leqslant \pi$, this is a matrix multiplication for which it holds that $\Phi_y(\omega) > 0$ only if $rank_{\mathbb{C}}(F(e^{i\omega})) = p$ and $\Phi_x(\omega) > 0$. If $rank_{\mathbb{R}(z)}(F(z)) = p$ then $rank_{\mathbb{C}}(F(e^{i\omega})) = p$ for almost all ω. Since $\Phi_x(\omega) > 0$ for almost all ω this implies that $\Phi_y(\omega) > 0$ for almost all ω. If $rank_{\mathbb{R}(z)}(F(z)) < p$ then $rank_{\mathbb{C}}(F(e^{i\omega})) < p$ for all ω and there will be no value of ω for which $\Phi_y(\omega) > 0$. □

5.B Proof of Theorem 5.1

The results of Lemma 5.1 and Proposition 5.1 indicate that the transfer function from external signals to κ generically has full row rank, if there are $n_{\mathcal{Q}} + n_{\mathcal{U}} + n_{\mathcal{Y}} + n_{\mathcal{K}}$ vertex disjoint paths between the external signals and κ, where $n_{\mathcal{U}}, n_{\mathcal{Y}}, n_{\mathcal{K}}$ are cardinality of set $\mathcal{U}, \mathcal{Y}, \mathcal{K}$ respectively.

In the system's setting with all unmeasured nodes removed as in (4.36), we first have to determine which external signals should be considered. According to the term $\bar{u}$ in system representation (4.36), the external excitation signals are u and disturbances sources ξ.

Characterizing ξ: disturbance sources in the transformed network

When premultiplying (4.29) with

$$P = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -\tilde{H}'_{\mathcal{UQ}} & -\tilde{H}'_{\mathcal{UO}} & I \end{bmatrix}$$

where $[\tilde{H}'_{\mathcal{UQ}} \quad \tilde{H}'_{\mathcal{UO}}] = [\tilde{H}_{\mathcal{UQ}} \quad \tilde{H}_{\mathcal{UO}}] \begin{bmatrix} \tilde{H}'''_{\mathcal{QQ}} & \tilde{H}'''_{\mathcal{QO}} \\ \tilde{H}''_{\mathcal{OU}} & \tilde{H}'''_{\mathcal{OO}} \end{bmatrix}^{-1}$, while only keeping the identity terms on the left hand side, we obtain an equivalent network equation

$$\begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{O}} \\ w_{\mathcal{U}} \end{bmatrix} = \begin{bmatrix} \bar{G}_{\mathcal{QQ}} & 0 & \bar{G}_{\mathcal{QU}} \\ \bar{G}_{\mathcal{OQ}} & 0 & \bar{G}_{\mathcal{OU}} \\ \breve{G}'_{\mathcal{UQ}} & \breve{G}'_{\mathcal{UO}} & \breve{G}'_{\mathcal{UU}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{O}} \\ w_{\mathcal{U}} \end{bmatrix} + \begin{bmatrix} \tilde{H}'''_{\mathcal{QQ}} & \tilde{H}'''_{\mathcal{QO}} & 0 \\ \tilde{H}''_{\mathcal{OQ}} & \tilde{H}''_{\mathcal{OO}} & 0 \\ 0 & 0 & \tilde{H}_{\mathcal{UU}} \end{bmatrix} \underbrace{\begin{bmatrix} \tilde{\xi}_{\mathcal{Q}} \\ \tilde{\xi}_{\mathcal{O}} \\ \tilde{\xi}_{\mathcal{U}} \end{bmatrix}}_{\tilde{\xi}} + P'\bar{u}\,, \quad (5.9)$$

where the third equation has been scaled to maintain a hollow matrix $\breve{G}'_{\mathcal{UU}}$. The disturbance term in this equation can, after spectral factorization and creating a monic, stable and minimum phase noise model, be rewritten into

$$\begin{bmatrix} \bar{H}_{\mathcal{QQ}} & \bar{H}_{\mathcal{QO}} & 0 \\ \bar{H}_{\mathcal{OQ}} & \bar{H}_{\mathcal{OO}} & 0 \\ 0 & 0 & \bar{H}_{\mathcal{UU}} \end{bmatrix} \begin{bmatrix} \xi_{\mathcal{Q}} \\ \xi_{\mathcal{O}} \\ \xi_{\mathcal{U}} \end{bmatrix}, \quad (5.10)$$

showing that $\xi_{\mathcal{U}}$ is a filtered version of $\tilde{\xi}_{\mathcal{U}}$.

Writing the disturbance $\tilde{\xi}_{\mathcal{U}}$ in terms of external signals

According to the proof of lemma 4.2 we have

$$\tilde{\xi} = \tilde{H}^{-1}\breve{H}e = \tilde{H}^{-1}\breve{v}. \quad (5.11)$$

where $\tilde{H}$ is a monic, stable and minimum phase rational matrix and $\breve{v}$ is the process noise on the nodes in the immersed network, i.e. the network that results after removing the unmeasured node signals. Following Lemma 3 in [104] (see Lemma 3.3), if condition 3.2 is satisfied, then $\tilde{H}$ is block diagonal and of the form

$$\tilde{H} = \begin{bmatrix} \tilde{H}_b & 0 \\ 0 & \tilde{H}_a \end{bmatrix}; \quad \breve{v} = \begin{bmatrix} \breve{v}_{\mathcal{Y}\cup\mathcal{B}} \\ \breve{v}_{\mathcal{A}} \end{bmatrix}.$$

where $\tilde{H}_b$ combines the block rows and columns related to the nodes in $\mathcal{Q} \cup \mathcal{O} \cup \mathcal{B} = \mathcal{Y} \cup \mathcal{B}$.

Since $\tilde{H}_b$ is monic, the matrix inverse definitely has nonzero diagonal terms, implying that with (5.11), $\breve{v}_{\mathcal{B}}$ is affecting $\tilde{\xi}_{\mathcal{B}}$, and with a similar reasoning $\breve{v}_{\mathcal{A}}$ is affecting $\tilde{\xi}_{\mathcal{A}}$. Consequently the disturbance terms that appear in $\tilde{\xi}_{\mathcal{U}}$ are given by $\breve{v}_{\mathcal{U}}$, which are the noise signals on $w_{\mathcal{U}}$ in the immersed network (3.11) with unmeasured nodes removed, and hence is a filtered version of all signals in e that have a direct or unmeasured path to a node in $w_{\mathcal{U}}$.

Combining the above result, and using the fact that $\xi_{\mathcal{U}}$ is a filtered version of $\tilde{\xi}_{\mathcal{U}}$ it follows that the following external signals appear in $\xi_{\mathcal{U}}$:

- $\breve{e}^{\mathcal{U}}$: all signals in e that have a direct or unmeasured path to a node in $w_{\mathcal{U}}$.

Finalizing the proof

The mapping that we need to evaluate for verifying the number of vertex disjoint paths is given by

$$(u, \xi) \rightarrow (w_{\mathcal{D}}, \xi_{\mathcal{Y}}, u_{\mathcal{K}}).$$

Since $\xi_{\mathcal{Y}}, u_{\mathcal{K}}$ appears on both sides of the mapping, the path condition can equivalently be formulated for the mapping

$$(u_{\bar{\mathcal{K}}}, \xi_{\mathcal{U}}) \rightarrow w_{\mathcal{D}}.$$

From the results of Proposition 4.4, we know that $\mathcal{K} = \mathcal{L} \backslash \{\mathcal{P} \cup \mathcal{A} \cup \bar{\mathcal{Z}}\}$. Therefore, $\bar{\mathcal{K}} = \mathcal{L} \backslash \mathcal{K} = \{\mathcal{P} \cup \mathcal{A} \cup \bar{\mathcal{Z}}\}$. Therefore, the path condition can equivalently be formulated for the mapping

$$(u_{\mathcal{P}}, u_{\mathcal{A}}, u_{\bar{\mathcal{Z}}}, \xi_{\mathcal{U}}) \rightarrow w_{\mathcal{D}}.$$

Given the external signals that affect $\xi_{\mathcal{U}}$ as analyzed above, it is sufficient to evaluate the mapping

$$(u_{\mathcal{P}}, u_{\mathcal{A}}, u_{\bar{\mathcal{Z}}}, e^{\mathcal{U}}) \rightarrow w_{\mathcal{D}}.$$

6 Chapter

A scalable multi-step least squares method

Identification methods for dynamic networks typically require prior knowledge of the network and disturbance topology, and often rely on solving poorly scalable non-convex optimization problems. While methods for estimating network topology are available in the literature, less attention has been paid to estimating the disturbance topology, i.e., the (spatial) noise correlation structure and the noise rank in a filtered white noise representation of the disturbance signal. In this work we present an identification method for dynamic networks, in which an estimation of the disturbance topology precedes the identification of the full dynamic network with known network topology. To this end we extend the multi-step Sequential Linear Regression [31] and Weighted Null Space Fitting methods [51] to deal with reduced rank noise, and use these methods to estimate the disturbance topology and the network dynamics in the full measurement situation. As a result, we provide a multi-step least squares algorithm with parallel computation capabilities that rely only on explicit analytical solutions, thereby avoiding the usual non-convex optimizations involved. Consequently we consistently estimate dynamic networks of Box-Jenkins (BJ) model structure, while keeping the computational burden low. We provide a consistency proof that includes path-based data informativity conditions for allocation of excitation signals in the experimental design. Numerical simulations performed on a dynamic network with reduced rank noise illustrate the potential of this method.

This chapter is based on: S. J. M. Fonken, K.R. Ramaswamy and P.M.J. Van den Hof, “A scalable multi-step least squares method for network identification with unknown disturbance topology", *To appear in Automatica, July 2022*, ArXiv: 2106.07548.

6.1 Introduction

While dynamic networks increase in complexity and size, and measurement data are becoming increasingly accessible, there is a strong demand for accurate and scalable data driven modeling methods. To identify all the modules in a dynamic network, the *joint direct method* [144] predicts all node signals in the network jointly and achieves consistency and minimum variance properties in the situation that the network and disturbance topology are given a priori and the noise can be of reduced rank. However it strongly relies on solving (constrained) non-convex optimization problems, which seriously limits its scalability to larger networks. There are multi-step convex identification methods available for full network identification, such as the Sequential Linear Regression (SLR) [31], Sequential Least Squares (SLS) [140] and extensions of Weighted Null Space Fitting (WNSF) [51] such as [43]. Moreover, methods such as the SLR and SLS allow for splitting the MIMO optimization into multiple linear regressions, which contributes to a lower computational burden. The available convex methods[1] are scalable to larger networks, but are limited to particular model structures of the network, and additionally, they do not allow for handling reduced rank noise. Particularly in large-scale network identification, stepping away from the typical assumption that all disturbance signals have their own independent noise source (i.e. full rank noise), is an appealing situation that should be supported by an effective estimation algorithm. Handling this situation of reduced-rank noise can substantially reduce the variance of estimated models. However it also introduces the problems of estimating the noise rank and noise correlation structure from data.

We witnessed in the previous chapters that in order to consistently identify the modules in a dynamic network, it is vital to know the *disturbance correlation structure* or *disturbance topology* for selecting the appropriate predictor model that can handle confounding variables. All available convex and non-convex methods for network identification require prior knowledge on the topology (i.e. rank and spatial correlation structure of the disturbance model). This information can be unknown. While in dynamic factor analysis [33] attention has been paid to the estimation of noise rank, in prediction error identification this does not appear to be included yet in the identification algorithms. For situations where the disturbance topology information is not readily available, it is attractive to develop methods that include estimating this information from data.

The topology estimation literature shows a variety of available methods to estimate the topology, such as Wiener filter based methods [82, 85], Bayesian model selection techniques [23, 116, 139], or methods that infer the topology from parametric estimates [11, 28, 152]. While the main focus of topology detection literature has been on estimating network topology in the situation of a diagonal disturbance spectrum $\Phi_v(\omega)$, extensions towards non-diagonal spectrum have been presented in [13, 37, 134]. In [134] network topology and the non-zero pattern in the disturbance spectrum are estimated jointly.

[1] These methods involve multiple steps, however the optimization problems involved in all the steps are convex.

In this chapter we assume that we do not know the disturbance topology a priori, but we assume that the network topology is known e.g., from its underlying physics, which is commonly the case for engineered systems. In the situation that the network topology is not known beforehand, it is possible to use any of the above cited methods to estimate it. We allow the process noise to be spatially correlated, i.e. the disturbance spectrum $\Phi_v(\omega)$ is not necessarily diagonal. Additionally the noise is allowed to be of reduced rank, i.e. $\Phi_v(\omega)$ can be singular.

6.1.1 Approach in a nutshell

The objective is to develop a multi-step convex algorithm that estimates the disturbance topology and all the dynamic modules in the network for general model structures including the Box Jenkins (BJ) structure, while adhering to computational algorithms that are scalable, while achieving favorable properties in terms of low experiment cost, consistency and reduced variance of the network estimates.

To this end we develop a multi-step algorithm to identify the network dynamics. In the first step the noise rank and the nonzero pattern in the corresponding disturbance model (noise shaping filter) are estimated. This is done through a (nonparametric) high-order ARX model, inspired by the SLR method [31]. Next, this information is used to develop a multi-step convex algorithm that can accurately identify the dynamics of the network in the situation of reduced rank noise and for a very general Box Jenkins model structure, thereby combining the recently introduced multi-step convex identification methods SLR [31] and WNSF [43, 51] and extending them to the described situation.

The chapter is organized as follows. In Section 6.2 we present a new method for estimating the disturbance topology from data, followed in Section 6.3 by a multi-step identification algorithm that exploits the prior estimated disturbance topology. Section 6.4 presents the consistency analysis of the method, including graph-based conditions for data informativity. Results of numerical simulations are provided in Section 6.5, followed by conclusions in Section 10.8. The consistency proofs are collected in the Appendix.

We assume that the data generating network satisfies the following assumptions is addition to Assumption 2.1.

Assumption 6.1

a. *All the modules in G^0 are strictly proper.*

b. *$\breve{H}^0$ is square, monic, stable and minimum phase.*

c. *The topology of G^0 and R^0, and the non-zero elements of R^0 are fixed* *and known.*

d. *The matrix R^0 has a block diagonal structure: $R^0 = diag(R^0_a, R^0_b)$ in the situation of ordered nodes as meant in* (2.4).

e. *Measurements of all node signals w and all present excitation signals r are available.*

f. *The standard regularity conditions on the data are satisfied that are required for convergence of parameter estimate of the prediction error identification method.*

The two main steps of the identification method that will be developed in this chapter are

- Estimating the disturbance topology, i.e. the noise rank and the zero pattern in the disturbance model.
- Estimating the dynamical components in the network for a given network and disturbance topology, while using a parametric BJ model structure.

In the next section we first focus on the disturbance topology estimation method, followed by the developed identification method in the section thereafter.

6.2 Disturbance topology estimation

Before we can use a unique disturbance model that is structured according to $\breve{H}^0$ in (2.4), we need to estimate the noise rank p and we need to be able to reorder the node signals in such a way that a noise representation as in (2.4) can be used. This step is necessary as the unstructured disturbance model H^0 is non-unique in the situation $p < L$. Therefore the disturbance topology estimation is performed in two main steps:

- Step 1: Estimating the noise rank, and reordering the signals to the situation of Lemma 2.1.
- Step 2: Estimating the structure of the disturbance model $\breve{H}^0$ (see Section 2.1.1).

6.2.1 Step 1: Estimating noise rank p and reordering of nodes

For estimating the noise rank p, we are going to estimate the covariance matrix $\breve{\Lambda}^0$ (2.5) of innovation signal $\breve{e}$, which through its rank p can provide us access to the correct noise rank.

An estimate of the covariance matrix is obtained by estimating a high-order (non-parametric) ARX model on the basis of measured signals w, r, and by using the residual (predictor error) of this estimated model as an estimate of the white noise term $\breve{e}$.

A parametrized ARX model is chosen according to

$$\begin{aligned} \breve{A}(q,\zeta) &= I + \breve{A}_1 q^{-1} + \cdots + \breve{A}_n q^{-n} & (6.1)\\ \breve{B}(q,\zeta) &= \breve{B}_0 + \breve{B}_1 q^{-1} + \cdots \breve{B}_n q^{-n} & (6.2) \end{aligned}$$

while all coefficients of $\breve{A}_k, \breve{B}_k$ are collected in the parameter vector ζ. The one-step-ahead predictor [77], defined as

$$\hat{w}(t|t-1;\zeta) := \bar{\mathbb{E}}\{w(t)|w^{t-1},\, r^t\} \tag{6.3}$$

is given by

$$\begin{aligned} \hat{w}(t|t-1,\zeta) &= \big(I - \breve{A}(q,\zeta)\big)w(t) + \breve{B}(q,\zeta)r(t) & (6.4)\\ &= \varphi(t)\zeta & (6.5) \end{aligned}$$

with $\varphi(t)$ composed of the appropriate terms in w and r.
Note that for an actual network with representation $G^0, \breve{H}^0, R^0$, the one-step predictor will be given by

$$\begin{aligned} \hat{w}(t|t-1) &= \big(I - (\breve{H}^0(q))^{-1}(I - G^0(q))\big)w(t) + \\ &\quad +(\breve{H}^0(q))^{-1}R^0(q)r(t). & (6.6) \end{aligned}$$

This implies that the polynomial predictor model (6.4) can only accurately approximate the rational filters that are present in (6.6) if the ARX order n is chosen very high. The ARX model is estimated according to $\hat{\zeta}_N^n = \arg\min_\zeta \frac{1}{N}\sum_{t=1}^N \varepsilon^T(t,\zeta)\varepsilon(t,\zeta)$, with $\varepsilon(t,\theta) = w(t) - \hat{w}(t|t-1;\zeta)$, leading to the analytical solution

$$\hat{\zeta}_N^n = \left[\frac{1}{N}\sum_{t=1}^N \varphi(t)\varphi^\top(t)\right]^{-1} \frac{1}{N}\sum_{t=1}^N \varphi(t)w(t). \tag{6.7}$$

Since the network identifiability conditions of [143] are satisfied for the considered model set, the sample estimate

$$\hat{\Lambda} := \frac{1}{N}\sum_{t=1}^N \varepsilon(t,\hat{\zeta}_N^n)\varepsilon^\top(t,\hat{\zeta}_N^n), \tag{6.8}$$

will then, under mild regularity conditions, be a consistent estimate of the noise covariance $\breve{\Lambda}^0$. The rank p of the noise process can then be estimated through a rank test on $\hat{\Lambda}$, e.g., through a singular value decomposition. Alternatively, other matrix factorizations or information based criteria can be applied for estimating the rank, see e.g., [17]. When $\hat{\Lambda}$ and the estimated rank $\hat{p} < L$ have been determined, the L signals can be reordered through a permutation matrix Π such that the first $\hat{p}$ components of the permuted noise vector have a rank $\hat{p}$ covariance matrix, i.e. $\begin{bmatrix} I_{\hat{p}} & 0 \end{bmatrix} \Pi^\top \hat{\Lambda} \Pi \begin{bmatrix} I_{\hat{p}} & 0 \end{bmatrix}^\top$ has rank $\hat{p}$.

Remark 6.1 *Since the polynomials $\breve{A}(\zeta)$ and $\breve{B}(\zeta)$ are fully parametrized with independent parameters on each polynomial entry, the MIMO least squares optimization that leads to the solution (6.7) can also be decomposed in L separate linear regressions that minimize the residual $\varepsilon_j(t,\zeta)$ separately for each j, which is computationally attractive since the computations can be performed in parallel or sequentially.*

Remark 6.2 *The resulting estimation scheme will generally not provide us with consistent estimates of the ARX model. This is not only due to the fact that typically the order n of the ARX model would need to go to infinity, but also to the fact that the solution for $\hat{\zeta}_N^n$ is non-unique in the situation $p < L$. However, this latter non-uniqueness does not affect the uniqueness and whiteness of the residual $\varepsilon(t, \hat{\zeta}_N^n)$ since, according to the projection theorem, every solution for $\hat{\zeta}_N^n$ determines the same predictor [32]. The estimate $\hat{\Lambda}$ is therefore consistent, i.e. $\hat{\Lambda} = cov(\breve{e})$ w.p. 1 as $n, N \to \infty$.*

Remark 6.3 *Although a correct estimation of the noise rank p cannot be guaranteed, consistency results for estimating p would be possible when applying information-based criteria for rank estimation, e.g., based on the BIC criterion [17]. In the next steps of our approach it will be assumed that a correct estimation of p has been obtained.*

After reordering the node signals as described above, we can now adhere to a network representation with a unique disturbance model according to the structure in Lemma 2.1, where $\breve{H}^0$ can be parametrized by the transfer function matrices H_a and H_b.

6.2.2 Step 2: Estimating the noise correlation structure

In the second step we are going to estimate which entries in our disturbance model are nonzero. To this end we extend the SLR method [31] to the situation of reduced rank noise and show how the noise correlation structure can be obtained.

Step 2.1: Refining the nonparametric ARX model

With the noise rank p available and the nodes being ordered, we have gained additional information on $\breve{H}^0$ (2.4), namely the last $L - p$ columns are now known. Now, we perform the same approach of identification using high order ARX modeling as in the previous step, but by utilizing the known entries in $\breve{H}^0$, leading to refined estimates of $\breve{A}(\hat{\zeta}_N^n)$ and $\breve{B}(\hat{\zeta}_N^n)$. In the analysis results of Section 6.4.1 it shown that the known entries in $\breve{H}^0$ can simply be mapped to known entries in the parametrized polynomial $\breve{B}(\zeta)$, and therefore can simply be taken into account in the least squares problem (6.7). In Section 6.4.1 it is shown that this leads to consistent estimates $\hat{\zeta}_N^n$ for $n, N \to \infty$.

Step 2.2: Predictor model with reconstructed innovation input

In this step we are going to use the estimated nonparametric ARX model to reconstruct the innovation signal. This allows us to use the reconstructed innovation signal as a measured input in the predictor model that will be used for estimating the structure of the disturbance model.

If there exists a parameter ζ^0 such that the ARX model $(\breve{A}(\zeta^0), \breve{B}(\zeta^0))$ captures the dynamics of the network, then it follows from [144] that

$$\varepsilon(t, \zeta^0) = \begin{bmatrix} I \\ \Gamma^0 \end{bmatrix} e(t). \tag{6.9}$$

We can accordingly decompose $\varepsilon(t, \zeta)$ as

$$\varepsilon(t, \zeta) = \begin{bmatrix} \varepsilon_a(t, \zeta) \\ \varepsilon_b(t, \zeta) \end{bmatrix} \tag{6.10}$$

while the consistency property of $\hat{\zeta}_N^n$ implies that

$$\begin{aligned} \varepsilon_a(t, \hat{\zeta}_N^n) &\to e(t) && \text{w.p. 1 as } N \to \infty \; \forall t, \\ \varepsilon_b(t, \hat{\zeta}_N^n) &\to \Gamma^0 e(t) && \text{w.p. 1 as } N \to \infty \; \forall t. \end{aligned} \tag{6.11}$$

We will refer to $\varepsilon(t, \hat{\zeta}_N^n)$ as the "reconstructed innovation".
For a network with ordered nodes we evaluate a new one-step-ahead predictor

$$\hat{w}(t|t-1) := \mathbb{E}\{w(t)|w^{t-1}, r^t, e^{t-1}\} \tag{6.12}$$

that includes the innovation signal $e^{t-1} := \{e(0), e(1), \cdots, e(t-1)\}$ in the expectation. Then it follows that

$$\hat{w}(t|t-1) = G^0(q)w(t) + (\breve{H}^0(q) - I)\breve{e}(t) + R^0(q)r(t), \tag{6.13}$$

where

$$(\breve{H}^0 - I)\breve{e} = \Big(\begin{bmatrix} H_a^0 & 0 \\ H_b^0 - \Gamma & I \end{bmatrix} - I\Big)\breve{e} = \begin{bmatrix} H_a^0 - I \\ H_b^0 - \Gamma^0 \end{bmatrix} e = \bar{H}^0 e. \tag{6.14}$$

This motivates the use of the following parametrized predictor model per node:

$$\hat{w}_j(t|t-1, \eta_j) = \sum_{l \in \mathcal{N}_j} G_{jl}(\eta_j) w_l + \sum_{s \in \mathcal{V}_j} \bar{H}_{js}(\eta_j) \varepsilon_{a_s}(\hat{\zeta}_N^n) + \sum_{k \in \mathcal{R}_j} R_{jk} r_k, \tag{6.15}$$

where the terms $G(\eta)$ and $\bar{H}(\eta)$ are parametrized versions of G^0 and $\bar{H}^0$ respectively, and $\varepsilon_a(\hat{\zeta}_N^n)$ is an estimate of the noise signal $e(t)$. $G_{jl}(\eta) = \sum_{k=1}^{n} g_k^{jl} q^{-k}$ and $\bar{H}_{js}(\eta) = \sum_{k=1}^{n} h_k^{js} q^{-k}$ are parametrized as strictly proper polynomials of order n, the term $\sum_{k\in\mathcal{R}_j} R_{jk} r_k(t)$ is known, the sets $\mathcal{N}_j$ and $\mathcal{R}_j$ are known from the topology of G^0 and R^0, and $\mathcal{V}_j$ defines the set of indices of noise signals for which noise dynamics is present in the disturbance model. This leads to an ARX model, like in Step 1, but now with the reconstructed innovation $\varepsilon_a(t, \hat{\zeta}_N^n)$ added as external predictor input signal, and the coefficients of the unknown polynomials collected in the parameter vector η. It is our next objective now to determine the sets $\mathcal{V}_j$ for $j = 1, \cdots L$. To this end we follow two approaches namely the structure selection approach and the Glasso approach, which will be presented next.

Structure selection

For a particular choice of $\mathcal{V}_j$ we evaluate the residual $\varepsilon_j(t, \hat{\eta}_{N_j}^n) := w_j(t) - \hat{w}_j(t|t-1, \hat{\eta}_{N_j}^n)$ where $\hat{\eta}_{N_j}^n$ is the estimated parameter that minimizes the quadratic criterion $\frac{1}{N}\sum_{t=1}^{N} \varepsilon_j^2(t, \eta_j)$, and that is obtained through an analytical solution, similar to (6.7). We test this residual with possible combinations in set $\mathcal{V}_j$ and employ model selection techniques such as AIC, BIC and Cross-validation (CV) on the obtained estimates $\hat{\eta}_{N_j}^n$ [152], of which the BIC provides a consistent estimate [71, 114]. Because we use ARX models to estimate η, model selection techniques such as AIC, BIC and CV involves convex optimization problems. Additionally, since we derive the disturbance topology per node, we have to test at most 2^L possible sets $\mathcal{V}_j$ for L nodes. This results in a lower computational burden compared to when we detect the topology in a MIMO setting, where we would have to test at most 2^{L^2-L} possible sets $\mathcal{V}_j$ simultaneously for all j [152]. However, for large networks these model selection techniques can still become computationally heavy.

Sparse estimation with Glasso

For each node j, a Glasso (Group Lasso) estimate is computed by minimizing the following cost function over η_j for a fully parametrized disturbance model with p white noise inputs:

$$\min_{\eta_j} \left\{ \frac{1}{2} \sum_{t=1}^{N} (w_j(t) - \hat{w}_j(t|t-1, \eta_j))^2 + \lambda_j \cdot \|\eta_j\|_2 \right\} \tag{6.16}$$

with the one-step-ahead predictor (6.15), and η_j being the vector of parameters related to the modules G_{ji} for $i \in \mathcal{N}_j$, and related to the modules $\bar{H}_{js}$ for $s = 1, \cdots p$; λ_j is the tuning parameter (penalization factor) of Glasso. The tuning of λ_j is described in the numerical illustrations in Section 6.5.

The right hand side of (6.16) is a mixed l_1/l_2 norm. The Glasso estimate is a convex extension to lasso that penalizes groups of estimated parameters [151], imposing sparsity at group level. Within a group, it does not yield sparsity [4]. If an appropriate penalization factor is chosen, only the dynamic modules that are actually present in the data generating network remain while the non-present terms are forced to 0, thus providing an estimate of the structure of $\bar{H}$.

With either of the methods (structure selection and sparse estimation with Glasso) the structure $\mathcal{V}_j$ of the disturbance model can be estimated entirely with convex and thus scalable methods, employing non-parametric (high order ARX-) models. This structural information can be effectively used in the actual estimation of parametric dynamic models in the next section.

> **Remark 6.4** *It is possible to add regularization when estimating the high-order ARX models presented in this section to guarantee stability of the estimates.*

6.3 Estimating parametric network models

The next step in our identification procedure is

- Step 3: Estimating a parametric network model.

While in Step 1 and 2 high order (nonparametric) models of the same model order n are used, and thus providing estimates with relatively high variance, in this step a parametric model is estimated from data where we exploit a very flexible Box-Jenkins model structure. In Step 3 we extend the WNSF method [51], and its application to dynamic networks in [43], to the reduced rank noise case such that we are able to obtain parametric models $G(\theta)$ and $H(\theta)$. The WNSF is in itself a three step method that starts with a high-order model before estimating the parametric model.

6.3.1 Step 3.1: Refining the nonparametric model

By fixing the correctly estimated disturbance topology obtained in the previous section we obtain consistent estimates of η_j using one-step-ahead predictor (6.15) defined in (6.12), leading to a high-order ARX model with structured disturbance model. The conditions for consistency of $\hat{\eta}^n_{j_N}$ are derived in Section 6.4. By employing the structured disturbance model we reduce the variance of $\hat{\eta}^n_{j_N}$, while the model order n remains the same.
Using the consistent estimate $\hat{\eta}^n_{j_N}$, we update the reconstructed innovation. Subsequently, we again update the high-order ARX model by replacing $\varepsilon_a(\hat{\zeta}^n_{j_N})$ with the updated reconstructed innovation $\varepsilon_a(\hat{\eta}^n_{j_N})$ in (6.15), and use this updated predictor to re-estimate η_j. This latter estimate can be seen as the starting high-order model for the WNSF method. At this point we still have a high variance on the estimates of η but negligible bias if model order n throughout all the steps is chosen sufficiently large. In the next step we reduce

the variance by reducing the number of parameters to estimate, where we will make the step from a high-order (nonparametric) model to a parametric model.

6.3.2 Step 3.2: Parametric model estimate

On the basis of the nonparametric model estimate characterized by $\hat{\eta}_{j_N}$ we are now going to estimate a parametric model of the dynamic network by utilizing a Box Jenkins model structure:

$$\begin{aligned} G_{jl}(q,\theta) &= \frac{l_1^{jl}q^{-1}+\cdots+l_{m_l}^{jl}q^{-m_l}}{1+f_1^{jl}q^{-1}+\cdots+f_{m_f}^{jl}q^{-m_f}}, \\ H_{jj}(q,\theta) &= \frac{1+c_1^{jj}q^{-1}+\cdots+c_{m_c}^{jj}q^{-m_c}}{1+d_1^{jj}q^{-1}+\cdots+d_{m_d}^{jj}q^{-m_d}}, \\ H_{js}(q,\theta) &= \frac{c_1^{js}q^{-1}+\cdots+c_{m_c}^{js}q^{-m_c}}{1+d_1^{js}q^{-1}+\cdots+d_{m_d}^{js}q^{-m_d}},\ s\neq j \end{aligned} \tag{6.17}$$

that can be rewritten as

$$G_{jl}(q,\theta) = \frac{L_{jl}(q,\theta)}{F_{jl}(q,\theta)}, \quad H_{js}(q,\theta) = \frac{C_{js}(q,\theta)}{D_{js}(q,\theta)}. \tag{6.18}$$

From $G_{jl}(\hat{\eta}_{j_N}^n)$ and $\bar{H}_{js}(\hat{\eta}_{j_N}^n)$ that are obtained in the previous step through the predictor (6.15), we can derive a related estimate of $H^0(q)$ according to (6.14) leading to $H(\hat{\eta}_N^n) = \bar{H}(\hat{\eta}_N^n) + \begin{bmatrix} I \\ \Gamma(\hat{\eta}_N^n) \end{bmatrix}$, with $\Gamma(\hat{\eta}_N^n)$ an estimate of the direct feedthrough term Γ^0 of H_b^0, and that based on the relation $\breve{e}_b(t) = \Gamma^0\breve{e}_a(t)$ from (2.4), can be given by

$$\Gamma(\hat{\eta}_N^n) = \Big(\frac{1}{N}\sum_{t=1}^{N}\varepsilon_b(\hat{\eta}_N^n)\varepsilon_a^\top(\hat{\eta}_N^n)\Big)\Big(\frac{1}{N}\sum_{t=1}^{N}\varepsilon_a(\hat{\eta}_N^n)\varepsilon_a^\top(\hat{\eta}_N^n)\Big)^{-1}. \tag{6.19}$$

Following the WNSF approach, we are now going to fit the parametric Box Jenkins model to the nonparametric model estimated from Step 3.1, by solving for θ in the equations

$$\begin{aligned} F_{jl}(\theta)G_{jl}(\hat{\eta}_N^n) - L_{jl}(\theta) &= 0\ , \\ D_{js}(\theta)H_{js}(\hat{\eta}_N^n) - C_{js}(\theta) &= 0. \end{aligned} \tag{6.20}$$

However, since these equations can not be solved exactly, an optimization problem is formulated [51] that comes down to minimizing the quadratic residual vector on the equations (6.20) by solving (in node-wise or MISO notation):

$$\min_{\theta_j} \|\hat{\eta}_{j_N}^n - Q_j(\hat{\eta}_{j_N}^n)\theta_j\|_2 \tag{6.21}$$

where

$$Q_j(\eta) = \begin{bmatrix} Q_j^g & 0 \\ 0 & Q_j^h \end{bmatrix}, \tag{6.22}$$

with Q_j^g and Q_j^h diagonal matrices with entries

$$\begin{aligned} Q_j^{g^{jl}}(\eta) &= \left[-\mathcal{T}_{n\times m_f}[G_{jl}(\eta)] \quad \bar{I}_{n\times m_l}\right], \\ Q_j^{h^{js}}(\eta) &= \left[-\mathcal{T}_{n\times m_d}[H_{js}(\eta)] \quad \bar{I}_{n\times m_c}\right], \end{aligned} \tag{6.23}$$

with model orders $m_i, i \in \{l, f, c, d\}$ according to (6.17), the top left corner of $\bar{I}_{n\times m}$ is $I_{m\times m}$ and has zeros otherwise, and $\mathcal{T}_{n\times m}[X_{ji}(q)]$ is a lower triangular Toeplitz matrix where the first column is $\begin{bmatrix} x_0^{ji} & \cdots & x_{n-1}^{ji} \end{bmatrix}^\top$ with $X_{ji}(q) = \sum_{k=0}^{\infty} x_k^{ji} q^{-k}$. The problem (6.21) is solved in first instance through the analytical least squares solution

$$\hat{\theta}_{jN}^{[0]} = \left(Q_j^\top(\hat{\eta}_{jN}^n) Q_j(\hat{\eta}_{jN}^n)\right)^{-1} Q_j^\top(\hat{\eta}_{jN}^n)\hat{\eta}_{jN}^n. \tag{6.24}$$

However, a parameter estimate with smaller variance can be achieved if a weighted least squares criterion is applied[2]. This is introduced in the next step.

6.3.3 Step 3.3: Re-estimation of parametric model

In this step we reduce the variance further by re-estimating the obtained parametric models $G(\theta)$ and $H(\theta)$ defined in (6.18). For a statistical optimal solution of (6.21), instead of the standard least squares problem (6.21), a weighted least squares problem should be solved, where the optimal weight is given by the inverse of the covariance matrix of the residual $\hat{\eta}_{jN}^n - Q_j(\hat{\eta}_{jN}^n)\theta_j^0$, with θ_j^0 the actual network coefficients related to node w_j. This is not directly applicable since θ_j^0 is unknown. However it can be shown [51] that

$$\hat{\eta}_{jN}^n - Q_j(\hat{\eta}_{jN}^n)\theta_j^0 = T_j(\theta_j^0)(\hat{\eta}_{jN}^n - \eta_j^{n0}), \tag{6.25}$$

with η_j^{n0} the real network coefficients related to the η-parametrized ARX model and $T_j(\theta)$ a block diagonal matrix with the denominator polynomials as entries

$$\begin{aligned} T_j^{g^{jl}}(\theta) &= \mathcal{T}_{n\times n}[F_{jl}(\theta)], \\ T_j^{h^{js}}(\theta) &= \mathcal{T}_{n\times n}[D_{js}(\theta)], \end{aligned} \tag{6.26}$$

where $\mathcal{T}_{n\times n}[X_{ji}(q)]$ is a lower triangular Toeplitz matrix where the first column is $\begin{bmatrix} 1 & x_1^{ji} & \cdots & x_m^{ji} & 0_{n-m-1} \end{bmatrix}^\top$ with $X_{ji}(q) = 1 + \sum_{k=1}^{\infty} x_k^{ji} q^{-k}$.

Result (6.25) motivates the use of a weighted least estimator with weighting matrix

$$W_j = T_j^{-1}(\theta_j^0)(P_{\hat{\eta}_{jN}^n})^{-1} T_j^{-T}(\theta_j^0)$$

with $P_{\hat{\eta}_{jN}^n}$ the covariance matrix of the nonparametric model. This can be

[2] As an alternative we can consider a weighted least squares criterion to obtain $\hat{\theta}_{jN}^{[0]}$ (6.24), with the covariance matrix of the nonparametric model as weight.

implemented in an iterative scheme according to

$$\begin{aligned}&\hat{\theta}_{j_N}^{[k+1]} = \\ &\left(Q_j^\top(\hat{\eta}_{j_N}^n)W_j(\hat{\theta}_{j_N}^{[k]})Q_j(\hat{\eta}_{j_N}^n)\right)^{-1}Q_j^\top(\hat{\eta}_{j_N}^n)W_j(\hat{\theta}_{j_N}^{[k]})\hat{\eta}_{j_N}^n.\end{aligned} \tag{6.27}$$

For consistency of the estimates of parameter vector θ we refer to the proof in the WNSF method [51], with the actual model orders m_i with $i = f, l, c, d$ (6.17) known.

Remark 6.5 *Because in this final step we correct for the variance due to the modeling error* (6.25)*, the final estimate will have a reduced variance.*

Throughout the presented steps we split the MIMO optimization into L linear regressions that rely on explicit analytical solutions, and that allows for parallel computing. The entire approach is presented in Algorithm 6.1.

Algorithm 6.1 *Algorithm for full network identification in dynamic networks, including disturbance topology detection*

Inputs: $w(t), r(t)$*,* $R^0(q)$*, model orders* $m_i, i \in \{l, f, c, d\}$*, network topology.*
Output: *Disturbance topology,* $\hat{\theta}_N$*.*

Disturbance topology detection

1. *Estimate noise rank* p *based on the reconstructed innovation* $\varepsilon(t, \hat{\zeta}_N^n)$ (6.10)*, and if* $p < L$ *order the nodes.*
2. 2.1 *Obtain consistent estimate* $\hat{\zeta}_N^n$ *with least squares solution* (6.31)*, where the nodes are ordered and by utilizing the estimated noise rank* p*.*

   2.2 *Use the reconstructed innovation* $\varepsilon_a(t, \hat{\zeta}_N^n)$ *as measured input in the one-step-ahead predictor* (6.15) *defined in* (6.12) *to estimate the noise correlation structure. We use*

   i. *Structure selection with AIC, BIC and CV,*

   ii. *Glasso,*

   applied to estimate $\hat{\eta}_{j_N}^n$ *that is obtained with least squares solution* (6.32).

Estimating parametric network models

3. 3.1 *Refine the nonparametric ARX model and obtain consistent estimate* $\hat{\eta}_N^n$ *with one-step-ahead predictor* (6.15)*, where the estimated disturbance topology is fixed and update the reconstructed innovation to* $\varepsilon_a(t, \hat{\eta}_N^n)$ *to re-estimate* $\hat{\eta}_N^n$*.*

3.2 Reduce the nonparametric ARX model to a parametric model and obtain initial estimate $\hat{\theta}_N^{[0]}$ by (6.24).

3.3 Re-estimate $\hat{\theta}_{j_N}^{[k+1]}$ with (6.27)*, where we update the weighting matrix $W_j(\hat{\theta}_{j_N}^{[k]})$ in each iteration.*

We continue to iterate the above step 3.3 until we have reached the convergence criterion $\frac{\|\hat{\theta}_N^{[k]} - \hat{\theta}_N^{[k-1]}\|}{\|\hat{\theta}_N^{[k-1]}\|} < 0.0001$. This convergence criterion is also used in the simulation results in Section 6.5. In the next section we derive the conditions required for consistency of estimates $\hat{\zeta}_{j_N}^n$ and $\hat{\eta}_{j_N}^n$.

6.4 Theoretical analyses

From here on we consider $n = n(N)$ i.e. the model order n increases as the data length N increases, while with increasing N, n/N tends to 0 with a particular rate (refer conditions D1 - D3 in [51], conditions D1 - D5 in [78]). Next we derive the conditions under which the estimates $\hat{\zeta}_N^n$ and $\hat{\eta}_N^n$, and consequently the reconstructed innovation are consistent.

6.4.1 Consistency of $\hat{\zeta}_N^n$ in Step 2.1: Refining the nonparametric model

With the noise rank p available and the nodes ordered we gained structural information on the unique noise model $\breve{H}^0(q)$ (2.4), namely we know that for the reduced noise rank case $p < L$ the last $L - p$ columns in $\breve{H}^0(q)$ are $\begin{bmatrix} 0 & I \end{bmatrix}^\top$. Moreover, taking the inverse of $\breve{H}^0(q)$ does not affect the last $L - p$ columns since

$$(\breve{H}^0)^{-1} = \begin{bmatrix} (H_a^0)^{-1} & 0 \\ -(H_b^0 - \Gamma^0)(H_a^0)^{-1} & I \end{bmatrix}. \tag{6.28}$$

As a result the term $(\breve{H}^0(q))^{-1}R^0(q)$ in the one-step predictor (6.6), has the following structure

$$(\breve{H}^0)^{-1}R^0 = \begin{bmatrix} (H_a^0)^{-1}R_a^0 & 0 \\ -(H_b^0 - \Gamma^0)(H_a^0)^{-1}R_a^0 & R_b^0 \end{bmatrix}, \tag{6.29}$$

with the second block column consisting of known terms only. This allows in the parametrization of the predictor (6.4) to replace the square polynomial $\breve{B}(\zeta)$ with a non-square polynomial $B(\zeta)$, leading to

$$\begin{aligned}\hat{w}(t|t-1,\zeta) &= \big(I - \breve{A}(\zeta)\big)w(t) + B(\zeta)r_a(t) + \begin{bmatrix}0\\R_b^0\end{bmatrix} r_b(t)\\ &= \varphi(t)\zeta + \begin{bmatrix}0\\R_b^0\end{bmatrix} r_b(t),\end{aligned} \tag{6.30}$$

with $\varphi(t)$ composed of the appropriate terms in w and r_a.

Note that for an actual network with representation $G^0, \breve{H}^0, R^0$, the one-step predictor is still given by (6.6), but now the predictor model (6.30) can use the known external excitation signals $r_b(t)$. The ARX model is estimated according to $\hat{\zeta}_N^n = \arg\min_\zeta \frac{1}{N}\sum_{t=1}^N \varepsilon^T(t,\zeta)\varepsilon(t,\zeta)$, with $\varepsilon(t,\theta) = w(t) - \hat{w}(t|t-1;\zeta)$, leading to the analytical solution:

$$\hat{\zeta}_N^n = \left[\frac{1}{N}\sum_{t=1}^N \varphi(t)\varphi^\top(t)\right]^{-1} \frac{1}{N}\sum_{t=1}^N \varphi(t)\left[w(t) - \begin{bmatrix}0\\R_b^0\end{bmatrix} r_b(t)\right]. \tag{6.31}$$

Note that Remark 1 holds and therefore predictor (6.30) can be decomposed in separate predictors for each node. The conditions for consistency are formulated in Proposition 6.1 and the proof is added in the appendix.

Proposition 6.1 *Consistency* $\hat{\zeta}_N^n$
Consider a dynamic network that satisfies Assumption 6.1. Additionally, consider the one-step-ahead predictor (6.30) *with* $\breve{A}(q,\zeta)$ *and* $B(q,\zeta)$ *are of high order. Then the transfer function matrices* $(\breve{H}^0(q))^{-1}(I - G^0(q))$ *and* $(\breve{H}^0(q))^{-1}\begin{bmatrix}R_a^0(q)^\top & 0\end{bmatrix}^\top$ *are consistently estimated with the analytical solution* (6.7)*, if the following conditions hold:*

1. *The external excitation* $r(t)$ *is uncorrelated to the noise* $e(t)$*.*
2. *The spectral density of* $\kappa(t) = \begin{bmatrix}r_a(t)^\top & w(t)^\top\end{bmatrix}^\top$*,* $\Phi_\kappa(\omega) > 0$ *for a sufficiently high number of frequencies* ω*.*

Proof: See appendix. ■

Remark 6.6 *Condition (1) and (2) of Proposition 1 are given for all signals present in the network. These conditions remain unchanged when we convert from a MIMO predictor to L linear regressions. Therefore the proof also holds for a MISO predictor assessed per node.*

6.4.2 Consistency of $\hat{\eta}_N^n$ in Step 3.1: Refining the nonparametric model

A refined nonparametric model is estimated by exploiting the information on the noise topology in the form of a structured polynomial model $B(\eta_j)$ for $\bar{H}_{js}(\eta_j)$ in the predictor (6.15), leading to the analytical solution

$$\hat{\eta}_N^n = \left[\frac{1}{N}\sum_{t=1}^{N}\varphi(t)\varphi^\top(t)\right]^{-1}\frac{1}{N}\sum_{t=1}^{N}\varphi(t)\big[w(t)-R^0r(t)\big]. \tag{6.32}$$

with $\varphi(t)$ composed of the appropriate terms in w and $\varepsilon(\hat{\eta}_N^n)$.
The conditions for consistency are formulated in Proposition 6.2.

Proposition 6.2 *Consistency* $\hat{\eta}_N^n$
Consider a dynamic network that satisfies Assumption 1 and Proposition 6.1, and assume the disturbance topology is estimated correctly. Additionally, consider the one-step-ahead predictor (6.15) *for all j. Then the transfer function matrices of $G^0(q)$ and $\breve{H}^0(q) - I$ are consistently estimated with the analytical solution $\hat{\eta}_N^n$* (6.32)*, if the following conditions hold:*

1. *For all j, the spectral density $\Phi_{\bar{\kappa}}(\omega)$ of $\bar{\kappa}(t) := \begin{bmatrix} w_{\{\mathcal{N}_j\}}(t)^\top & e_{\{\mathcal{V}_j\}}(t)^\top \end{bmatrix}^\top$, satisfies $\Phi_{\bar{\kappa}}(\omega) > 0$ for a sufficiently high number of frequencies ω.*
2. *The data generating system is in the model set, i.e. there exists a η_0 such that $G(q, \eta_0) = G^0(q)$ and $\bar{H}(q, \eta_0) = \breve{H}^0(q) - I$.*

Proof: See appendix. ■

With consistent estimate $\hat{\eta}_N^n$ we can update the reconstructed innovation $\varepsilon(t, \hat{\eta}_N^n) = \begin{bmatrix}\varepsilon_a(t, \hat{\eta}_N^n)^\top & \varepsilon_b(t, \hat{\eta}_N^n)^\top\end{bmatrix}^\top$ consistently for each time step $t = 1, \ldots, N$

$$\varepsilon(t, \hat{\eta}_N^n) \to \breve{e}(t) \qquad \text{w.p. 1 as } N \to \infty\ \forall t, \tag{6.33}$$

where the innovation is reconstructed per node according to $\varepsilon_j(t, \eta) = w_j(t) - \hat{w}_j(t|t-1, \eta)$ using one-step-ahead predictor (6.15).

> **Remark 6.7** *Note that Condition 2 of Proposition 6.2 incorporates the condition that the noise rank p is chosen correctly, and the disturbance model is flexible enough to represent the exact disturbance topology of the network.*

Following the line of reasoning in [127], the spectral conditions in Propositions 6.1 and 6.2, which are actually data informativity conditions, can generically be replaced by path-based conditions on the graph of the network model set.

6.4.3 Generic data informativity conditions

Condition (2) of Proposition 6.1 and Condition (1) of Proposition 6.2 is a spectral data informativity condition on internal node signals in w, and it is difficult to interpret it for an experimenter. In this section we replace the spectral condition with a path-based data informativity condition in a generic sense[3], i.e. independent of the numerical values of the network dynamics. By doing so we can evaluate if data informativity is satisfied based on the network and disturbance topology, and the properties of the external signals. Next we formulate the conditions in terms of properties and locations of the external signals analogous to Lemma 5.1 and Proposition 5.1 from Chapter 5, by means of vertex-disjoint paths from external signals to internal node signals, where two paths are vertex-disjoint if they have no nodes in common, including their start and end nodes [131]. The consequences are illustrated in a 6-node example.

Vertex-disjoint paths

The generic version of Condition (2) of Proposition 6.1 is given in Proposition 6.3.

Proposition 6.3 *The spectrum condition* $\Phi_\kappa(\omega) > 0$ *for* $\kappa(t) = \begin{bmatrix} r_a(t)^\top & w(t)^\top \end{bmatrix}^\top$ *in Condition (2) of Proposition 6.1 is generically satisfied if there are* L *vertex-disjoint paths from* $\begin{bmatrix} r_b(t)^\top & e(t)^\top \end{bmatrix}^\top$ *to* $w(t)$.

Proof: See appendix. ■

Proposition 6.3 gives a sufficient generic path-based condition that requires to have external excitation signals at certain locations in the network, combining data informativity conditions with identifiability [127].

The set $\mathcal{V}$ denotes the set of indices of all the disturbing noise signals, where $\mathcal{V}_j$ is a subset of $\mathcal{V}$. For the generic condition for Condition (1) of Proposition 6.2 we introduce notation $e_{\{\mathcal{X}_j\}}(t)$, where $\mathcal{X}_j$ is the set of indices of all the disturbing noise signals excluding indices that are already present in set $\mathcal{V}_j$, i.e. $\mathcal{X}_j = \mathcal{V}/\mathcal{V}_j$.

Proposition 6.4 *The spectrum condition* $\Phi_{\bar{\kappa}}(\omega) > 0$ *for* $\bar{\kappa}(t) = \begin{bmatrix} w_{\{\mathcal{N}_j\}}(t)^\top & e_{\{\mathcal{V}_j\}}(t)^\top \end{bmatrix}^\top$ *in Condition (1) of Proposition 6.2 is generically satisfied if there are Cardinal*$\{\mathcal{N}_j\}$ *vertex-disjoint paths from* $\begin{bmatrix} r(t)^\top & e_{\{\mathcal{X}_j\}}(t)^\top \end{bmatrix}^\top$ *to* $w_{\{\mathcal{N}_j\}}(t)$.

Proof: See appendix. ■

Proposition 6.4 gives a sufficient generic path based condition that requires

[3] Genericity is considered in the sense that the corresponding property holds for almost all models in the model set, possibly excluding a set of measure 0.

external excitation signals at certain locations such that $\Phi_{\bar{\kappa}}(\omega) > 0$ for a sufficiently high number of frequencies.

> **Remark 6.8** *If we want to identify only the j^{th} row of the network (or only part of the network), we can consider the predictor in Proposition 6.2 only for node j and satisfy the conditions in Proposition 6.2 and 6.4 for node j.*

Next we elaborate the vertex-disjoint path conditions by means of an example where a network is subject to reduced rank noise.

Reduced rank noise example

We consider a 6-node network that satisfies Assumption 1 and is subject to reduced rank noise of rank $p = 4$ shown in Figure 6.1. This 6-node example is additionally used in the simulations in Section 6.5, and is further defined in Appendix 6.E. The nodes are ordered such that the first p nodes are subject to full rank noise. Moreover, we assume the disturbance topology is correctly estimated. The goal of this example is to elaborate on the path-based data informativity conditions given in Proposition 6.3 and 6.4. To be more specific, we show which external excitation signals are sufficient in order to satisfy the spectral Condition (2) in Proposition 6.1 and Condition (1) in Proposition 6.2. In the example we have external noise signals $e(t) = \begin{bmatrix} e_1(t) & \dots & e_4(t) \end{bmatrix}^\top$ and external excitation signals $r_k(t)$, for simplicity we assume R^0 contains elements that are either 0 or 1.

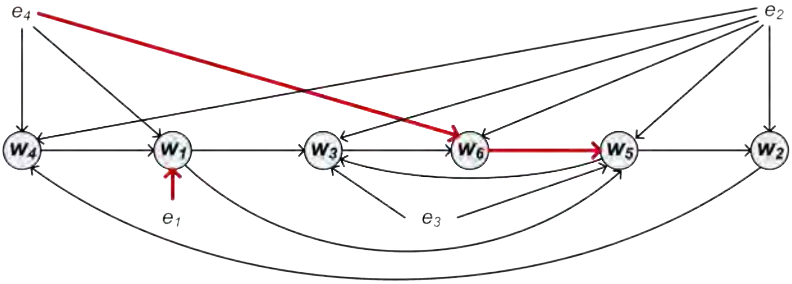


Figure 6.1: 6-node dynamic network with reduced rank noise that has rank $p = 4$, no $r(t)$ signals are shown. The arrows represent the edges for which $G^0_{ji} \neq 0$ and $H^0_{ji} \neq 0$, where the arrows indicated in red are examples of the two vertex disjoint paths needed to satisfy Proposition 6.4 for output $w_3(t)$.

In order to satisfy Proposition 6.3, we require $L = 6$ vertex-disjoint paths from $\begin{bmatrix} r_b(t)^\top & e(t)^\top \end{bmatrix}^\top$ to $w(t)$, with $r_b(t)$. The first $p = 4$ nodes, denoted by $w_a(t)$, are excited by the noise $e(t)$; we therefore require at least $L - p = 2$ external excitation signals $r_k(t)$ on the last 2 nodes $w_b(t) = \begin{bmatrix} w_5(t) & w_6(t) \end{bmatrix}^\top$, i.e $r_b(t) = \begin{bmatrix} r_5(t) & r_6(t) \end{bmatrix}^\top$ with $R_b = I \in \mathbb{R}^{2\times 2}$. Therefore we satisfy Proposition 6.3 since we have 6 vertex-disjoint paths from $\begin{bmatrix} e(t)^\top & r_b(t)^\top \end{bmatrix}^\top$ to $\begin{bmatrix} w_a(t)^\top & w_b(t)^\top \end{bmatrix}^\top$.

To show how Proposition 6.4 is satisfied, we first consider output node $w_3(t) = G_{31}(\eta)w_1(t) + G_{35}(\eta)w_5(t) + H_{32}(\eta)e_2(t) + H_{33}(\eta)e_3(t)$, that has $w_{\{\mathcal{N}_3\}}(t) = \begin{bmatrix} w_1(t) & w_5(t) \end{bmatrix}^\top$ and $e_{\{\mathcal{V}_3\}}(t) = \begin{bmatrix} e_2(t) & e_3(t) \end{bmatrix}^\top$. We need Cardinal$\{\mathcal{N}_3\} = 2$ vertex-disjoint paths from $\begin{bmatrix} r(t)^\top & e_{\{\mathcal{X}_j\}}(t)^\top \end{bmatrix}^\top$ to $w_{\{\mathcal{N}_3\}}(t)$. There already exist 2 vertex disjoint paths from $e_{\{\mathcal{X}_j\}}(t) = \begin{bmatrix} e_1(t) & e_4(t) \end{bmatrix}^\top$ to $w_{\{\mathcal{N}_3\}}(t)$. This shows that Proposition 6.4 is satisfied by the two vertex disjoint paths from $e_1(t) \rightarrow w_1(t)$ and from $e_4(t) \rightarrow w_6(t) \rightarrow w_5(t)$ as indicated in red in Figure 6.1. If we apply the same reasoning to the other nodes we see that for node

- $w_1(t)$ with $w_{\{\mathcal{N}_1\}}(t) = w_4(t)$, there exists a vertex-disjoint path from $e_2(t) \rightarrow w_4(t)$.
- $w_2(t)$ with $w_{\{\mathcal{N}_2\}}(t) = w_5(t)$, there exists a vertex-disjoint path from $e_3(t) \rightarrow w_5(t)$.
- $w_4(t)$ with $w_{\{\mathcal{N}_4\}}(t) = w_2(t)$, there exists a vertex-disjoint path from $e_3(t) \rightarrow w_5(t) \rightarrow w_2(t)$
- $w_5(t)$ with $w_{\{\mathcal{N}_5\}}(t) = \begin{bmatrix} w_1(t) & w_6(t) \end{bmatrix}^\top$, there exist 2 vertex-disjoint paths from $e_1(t) \rightarrow w_1(t)$ and from $e_4(t) \rightarrow w_6(t)$.
- $w_6(t)$ with $w_{\{\mathcal{N}_3\}}(t) = w_3(t)$, there exists a vertex-disjoint path from $e_3(t) \rightarrow w_3(t)$.

In order to satisfy Proposition 6.4 we therefore do not require additional external excitation signals $r_k(t)$. Consequently, in order to identify the full network for the given example, it is sufficient to add external signals $r_b(t) = \begin{bmatrix} r_5(t) & r_6(t) \end{bmatrix}^\top$ with $R_b = I \in \mathbb{R}^{2\times 2}$ that satisfies Proposition 6.3.

6.5 Numerical simulations

In this section we show the results of different steps in Algorithm 1. We assume $R^0 = I$, and consider the system given in Figure 6.1 and Appendix 6.E.

For the simulation study we use normally distributed zero mean white external signals, where $\{r(t)\}$ has a variance of 5 and the vector of e-signals has variances $\{0.1, 0.2, 0.3, 0.4\}$. We simulate the nodes according to $w(t) = (I - G^0)^{-1}(R^0 r(t) + H^0 e(t))$ and perform $M = 100$ Monte Carlo runs over five data lengths logarithmically spaced between 300 and 50000. For each of the data lengths N a specific value of the model order n is chosen according to $n = 10, 20, 30, 40, 40$, for increasing values of N. The actual model orders $m_i, i \in \{l, f, c, d\}$ can be derived from Appendix 6.E.

Next we describe the noise rank estimation results of step 1 of Algorithm 1.

6.5.1 Rank p and ordering of the nodes

In order to obtain the noise rank p we perform a rank test (singular value decomposition) on covariance matrix $\hat{\Lambda}$ (6.8). For data length $N = 300$, the singular values averaged over the 100 Monte Carlo runs are $svd(\hat{\Lambda}_N) = \begin{bmatrix}0.37 & 0.26 & 0.21 & 0.06 & 2.13{\cdot}10^{-8} & 1.96{\cdot}10^{-9}\end{bmatrix}$, where we see that the last two singular values are close to zero. As data length increases the last two values converge even closer to zero. For $N = 50000$ we obtain the following averaged singular values $svd(\hat{\Lambda}_N) = \begin{bmatrix}0.59 & 0.40 & 0.39 & 0.10 & 4.04{\cdot}\,10^{-13} & 1.24{\cdot}10^{-13}\end{bmatrix}$, showing that a clear gap between the fourth and fifth singular value points to a correct rank estimate of 4. Finally with the noise rank p available we can reorder the nodes such that $\begin{bmatrix}I_p & 0\end{bmatrix}\Pi^\top\hat{\Lambda}\Pi\begin{bmatrix}I_p & 0\end{bmatrix}^\top$ has rank p.

Next we show the disturbance topology detection results of step 2 of Algorithm 1.

6.5.2 Topology estimation of the disturbance model

For the topology detection we are interested in which indices belong in set $\mathcal{V}_j$ for all j, where the indices indicate where the edges are located in the disturbance model. We evaluate the performance of the topology detection by evaluating the trade-off between overestimating and underestimating the number of edges, that is typically used in receiver operating characteristic (ROC) curves [61].

If an edge is present in both the data generating disturbance and the estimated disturbance topology, we count this edge as a true positive (TP). If an edge is present in the estimated disturbance topology but does not exist in the data generating system, we count this edge as a false positive (FP). Additionally we let Pos indicate the total number of existing edges and Neg indicates the total number of non-existing edges in the disturbance model. The ROC curve plots the true positive rate (TPR) versus the false positive rate (FPR), with

$$TPR = \frac{TP}{Pos}, \quad FPR = \frac{FP}{Neg}, \tag{6.34}$$

where FPR=0 and TPR=1 represented by the point $(0, 1)$, indicates the topology is perfectly reconstructed. We evaluate the closeness to the point $(0, 1)$ by utilizing the distance function

$$dis = \sqrt{FPR^2 + (1 - TPR)^2}, \tag{6.35}$$

For the structure selection procedure we test all possible combinations in set $\mathcal{V}_j$ and employ AIC, BIC and CV. For AIC we use

$$\frac{1}{2}\log\Big(V_{j_N}(\hat{\eta}^n_{j_N})\Big) + \frac{n_{p_j}}{N}, \tag{6.36}$$

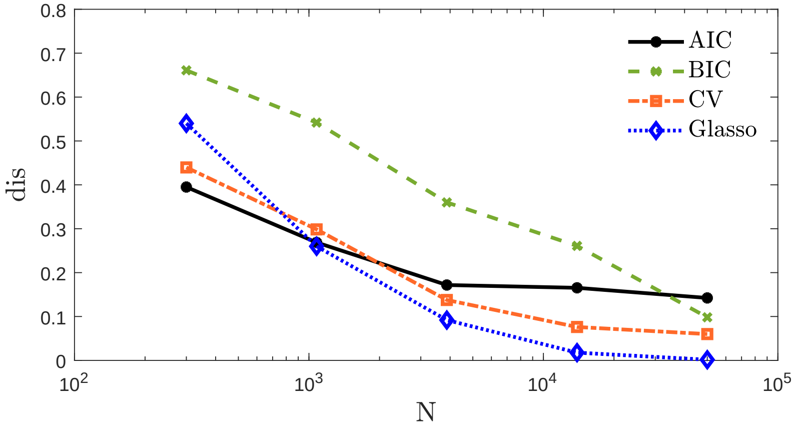


Figure 6.2: dis as a function of N, averaged over the Monte Carlo runs.

with n_{p_j} the number of estimated parameters for node j and

$$V_{j_N}(\hat{\eta}^n_{j_N}) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon_j(t,\hat{\eta}^n_{j_N})^2. \tag{6.37}$$

For BIC we use

$$N * \log\Big(V_{j_N}(\hat{\eta}^n_{j_N})\Big) + N(\log(2\pi)+1) + n_{p_j}\log(N). \tag{6.38}$$

From these simulations we select set $\mathcal{V}_j$ that gives the smallest AIC or BIC value. For the CV we split the data $Z^N = Z^{(1)}Z^{(2)}$ in a training set $Z^{(1)}$ of length $\frac{2}{3}(N+1)$ and obtain the estimates for the different combinations in set $\mathcal{V}_j$ according to

$$\hat{\eta}^{(1)}_{j_N} = \underset{\eta}{\text{argmin}} V_{j_N}(\eta_j, Z^{(1)}), \tag{6.39}$$

With the validation set $Z^{(2)}$, that contains the remaining data of length $N^{(2)} = \frac{1}{3}(N+1)$, we minimize objective function

$$V_{j_N}(\hat{\eta}^{(1)}_{j_N}, Z^{(2)}) = \frac{1}{N^{(2)}}\sum_{t=1}^{N^{(2)}} \varepsilon_j(t,\hat{\eta}^{(1)}_{j_N})^2, \tag{6.40}$$

and select the set $\mathcal{V}_j$ that gives the smallest root mean squared error (RMSE)

$$RMSE_j = \sqrt{V_{j_N}(\hat{\eta}^{(1)}_{j_N}, Z^{(2)})}. \tag{6.41}$$

For Glasso we fully parametrize the disturbance model, using the known topology of G^0 and fixed $R^0 = I$. We inspect all elements of the disturbance model matrix that is parametrized with the Glasso estimates (6.16). If element

$H_{ji}(\hat{\eta}_N)$ of the disturbance model matrix contains nonzero Glasso estimates we say this element contains dynamics, and therefore an edge is present and $i \in \mathcal{V}_j$. To prevent arbitrary small Glasso estimates are seen as dynamics we define a tolerance, where the Glasso estimates are nonzero if the l_2 norm of these estimates is larger than 10^{-3}. The choice to include the estimates of $G_{jl}(\eta)$ in the penalization is due to the implementation of Glasso [15]. For good estimates on the disturbance topology, we utilize the known topology of G^0 and deal with known $R^0 r(t)$ signals appropriately.

Tuning of λ_j is done via a grid based search similar to the CV structure selection. First we select a grid $\lambda_j^{grid} = \{0, 25, 50, \cdots, 2000\}$ containing λ_j values to test. For each grid point we estimate $\hat{\eta}_j^{grid}$ using Glasso, from where the topology is derived by inspecting the disturbance model for dynamics as mentioned before, and fix the topology H_j^{grid} per node. Next we apply CV using topology H_j^{grid} and estimate the RMSE$_j$. The grid point with the lowest RMSE$_j$ is selected as the λ_j value. Repeating the tuning procedure over a number of runs gives the minimally required value for λ_j. The tuning procedure is applied to all nodes for the different data lengths N.

Figure 6.2 shows the topology detection results, with the distance averaged over 100 Monte Carlo runs. The BIC is a consistent information criterion [71, 114], meaning that the estimated disturbance topology will converge to the actual topology if $N \to \infty$. However, as can be seen in the results in Figure 6.2, the full convergence of the BIC procedure is not reached for the given data lengths. Until the BIC procedure converges to the actual disturbance topology, it tends to underestimate the number of edges that are actually present, therefore the mismatch in the distance function is caused by not detecting all the TP's. The AIC is not a consistent information criterion, but has a faster convergence rate compared to the BIC [153]. The AIC tends to overestimate the number of edges, meaning the mismatch is caused by detecting the FP's. The CV is comparable to AIC but has a slower convergence rate. Finally the Glasso seems to have the best of both AIC and BIC. However, these results heavily depend on the selected tuning parameter λ, where it is not guaranteed that a suitable λ exists.

Next we show the parametric estimation results of step 3 of Algorithm 1, where we fix the estimated disturbance topology. Based on the results in Figure 6.2 we have fixed the correctly estimated disturbance topology obtained with Glasso for $N = 50000$, where $TPR = 1$ and $FPR = 0$.

6.5.3 Estimating the parametric model

Next we present the results of the estimation of the parametric model. Because Algorithm 1 is consistent we have a negligible bias and the mean squared error (MSE) represents the variance. For the simulations we use the correct estimated disturbance topology from the previous step. Additionally, for Step 3.2 of Algorithm 1, we compute the $\hat{\theta}_{j_N}^{[0]}$ in (6.24) using the covariance matrix of the nonparametric model as weighting. Figures 6.3 and 6.4 present the sample MSE

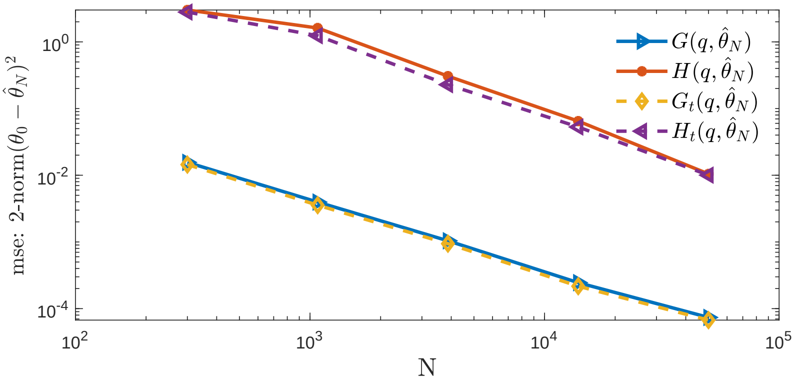


Figure 6.3: MSE between $\hat{\theta}_N$ and θ_0 as function of sample size, averaged over the Monte Carlo runs, obtained with Algorithm 1 with $R^0 = I$, where subscript $\{t\}$ indicates the use of the true (unknown) white noise as a predictor input instead of the reconstructed innovation.

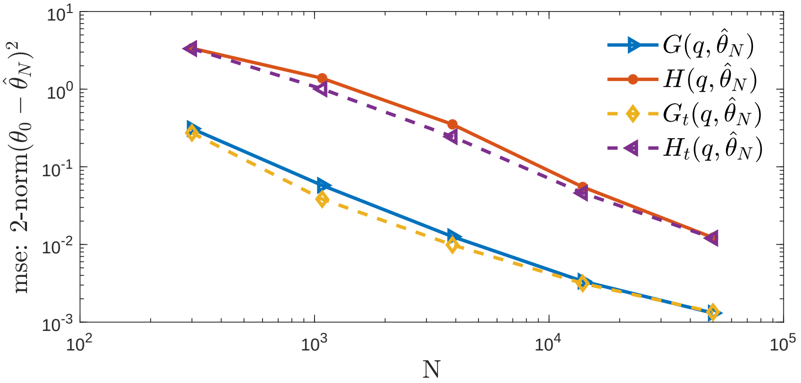


Figure 6.4: MSE between $\hat{\theta}_N$ and θ_0 as function of sample size, averaged over the Monte Carlo runs, obtained with Algorithm 1 with $R^0 = \begin{bmatrix} 0 & R_b^{0^\top} \end{bmatrix}^\top$ and $R_b = I \in \mathbb{R}^{2\times 2}$, where subscript $\{t\}$ indicates the use of the true (unknown) white noise as a predictor input instead of the reconstructed innovation.

that is computed according to $\text{MSE}(N) = \frac{1}{M}\sum_{c=1}^{M}\left\|\hat{\theta}_{N,c} - \theta_0\right\|^2$, where c indicates the Monte Carlo run and $\hat{\theta}_{N,c}$ the final estimate (6.27). In Figure 6.3 we use $R^0 = I$ in the data generating network, and in Figure 6.4 we use $R^0 = \begin{bmatrix} 0 & R_b^{0\top} \end{bmatrix}^\top$ with $R_b = I \in \mathbb{R}^{2\times 2}$ according to Section 6.4.3. The solid lines represent Algorithm 1 where the estimates are obtained using the reconstructed innovation as input. The dotted lines represent Algorithm 1 where we use the realization of the actual noise $e(t)$ as input, indicated by subscript $\{t\}$. The results for the whole network are shown, while using L MISO linear regressions. Both simulations shown in Figures 6.3 and 6.4, typically perform $k = 6$ iterations for data length $N = 300$ in (6.27). As the data length N increases the number of iterations performed decreases, where for $N = 50000$ the simulations typically perform $k = 2$ iterations. The $\text{MSE}(N)$ improvement after the iterations is shown in Table 6.1. From Table 6.1 we can derive that we benefit most from iterating k in the final step of Algorithm 1 if we do not have full excitation on the network with $R^0 = I$.

N	300	1078	3873	13916	50000
$R^0 = I$	$1.6\cdot 10^{-3}$	$5.1\cdot 10^{-5}$	$-1.2\cdot 10^{-6}$	$-1.9\cdot 10^{-7}$	$3.7\cdot 10^{-8}$
$R_b^0 = I$	0.43	0.26	0.15	0.07	0.01

Table 6.1: MSE improvement:
$\frac{1}{M}\sum_{c=1}^{M}\left\|\hat{\theta}_{N,c} - \theta_0\right\|^2 - \frac{1}{M}\sum_{c=1}^{M}\left\|\hat{\theta}^{(1)}_{N,c} - \theta_0\right\|^2$ over k iterations

In Figures 6.3 and 6.4 we see convergence between the solid and dotted lines as the data length N increases. This indicates that as data length N increases the reconstructed innovation converges to the actual noise. Furthermore all MSE results continue to converge towards zero which is in line with the consistency proof.

The results of this simulation study support the consistency proof and we consistently estimate the BJ model structure, while employing a row-wise optimization.

6.6 Conclusions

In this chapter we present a multi-step least squares method for network identification without prior information on disturbance topology, that can handle reduced rank noise with low computational burden. We follow a step wise procedure where we first extend the SLR identification method to detect the disturbance topology, and thereafter extend the WNSF method to consistently identify networks of general model structure, including a BJ model structure. For a BJ network, usually a non-convex MIMO identification method is needed. In this chapter, we show that we identify the BJ network using analytical solutions. Simulation results indicate that we can identify the disturbance topology of the given network with low error if the data length N is sufficiently large. We show that the presented method is consistent, and provide path based data informativity conditions, that guides where to allocate external excitation signals for the experimental design. Considering large networks subject to correlated and/or reduced rank noise, the presented method is promising due to its scalability and low variance results.

The presented method is modular. That is, if the disturbance topology is needed for any identification method, the disturbance topology detection step can be separately used. Similarly, if the topology information are known apriori, the estimation of parametric model step can be used. In this chapter we have estimated the full network, however extensions can be easily made to estimate a single module or a set of modules in a network.

Appendices

6.A Proof of Proposition 6.1

Consider the prediction error for the predictor $\hat{w}(t|t-1,\zeta)$ from (6.30):

$$\begin{aligned} \varepsilon(t,\zeta) = w(t) - \hat{w}(t|t-1,\zeta) &= \breve{A}(\zeta)w(t) - \breve{B}(\zeta)r(t), \\ &= \breve{A}(\zeta)w(t) - B(\zeta)r_a(t) - \begin{bmatrix}0 & R_b^\top\end{bmatrix}^\top r_b(t). \end{aligned} \tag{6.42}$$

With the data generating system (2.2) given as

$$\begin{aligned} w(t) = (\breve{A}^0)^{-1}\breve{B}^0 r(t) + (\breve{A}^0)^{-1}\breve{e}(t), \\ \text{with } \breve{A}^0 = (\breve{H}^0)^{-1}(I - G^0),\ \ \breve{B}^0 = (\breve{H}^0)^{-1}R^0 \end{aligned} \tag{6.43}$$

we can rewrite the prediction error as

$$\varepsilon(t,\zeta) = \big(\breve{A}^0 - \Delta\breve{A}(\zeta)\big)w - \big(\breve{B}^0 - \Delta\breve{B}(\zeta)\big)r \tag{6.44}$$

with $\Delta\breve{A}(\zeta) = \breve{A}^0 - \breve{A}(\zeta)$ and $\Delta\breve{B}(\zeta) = \breve{B}^0 - \breve{B}(\zeta)$. Then with (6.43) it follows that

$$\varepsilon(t,\zeta) \quad = \quad \Delta\breve{B}(\zeta)r - \Delta\breve{A}(\zeta)w + \breve{e}, \tag{6.45}$$

and since the second block column of $\breve{B}(\zeta)$ is fixed and known, it follows that $\Delta\breve{B}(\zeta)r = \Delta B(\zeta)r_a$. We now proceed by evaluating the j-th component

$$\varepsilon_j(t,\zeta) = \quad \Delta B_j(\zeta)r_a - \Delta\breve{A}_j(\zeta)w + \breve{e}_j, \tag{6.46}$$

where $\Delta\breve{A}_j(\zeta)$ and $\Delta B_j(\zeta)$ are the rows of matrices $\Delta\breve{A}(\zeta)$ and $\Delta B(\zeta)$ belonging to node j.
The consistency proof consists of two steps:

1. Show that the objective function is bounded from below by the noise variance $\bar{V}_j(\zeta) := \bar{\mathbb{E}}\varepsilon_j^2(t,\zeta) \geqslant \sigma_{\breve{e}_j}^2$, where the minimum is achieved for $\Delta\breve{A}_j(\zeta) = 0$ and $\Delta\breve{B}_j(\zeta) = 0$.

2. Show that the global minimum is unique.

6.A.1 Consistency proof step (1)

With (6.43) substituted into (6.46), the expression for $\varepsilon_j(t,\zeta)$ becomes

$$\Delta B_j(\zeta) r_a - \Delta \breve{A}_j(\zeta)\Big((\breve{A}^0)^{-1}\breve{B}^0 r + (\breve{A}^0)^{-1}\breve{e}\Big) + \breve{e}_j \tag{6.47}$$

from which, due to the fact that $\Delta\breve{A}_j(\zeta)$ is strictly proper and r and e are uncorrelated, it follows that $\breve{e}_j$ is uncorrelated with the remaining terms in the expression. As a result, the objective function is given by

$$\bar{V}_j(\zeta) = \bar{\mathbb{E}}\Big[\Big(\Delta B_j(\zeta) r_a - \Delta\breve{A}_j(\zeta) w\Big)^2\Big] + \sigma^2_{\breve{e}_j}, \tag{6.48}$$

from which we can infer that $\bar{V}_j(\zeta) \geqslant \sigma^2_{\breve{e}_j}$ with equality for $\Delta\breve{A}_j(\zeta) = 0$ and $\Delta B_j(\zeta) = 0$.

6.A.2 Consistency proof step (2)

For the second step we show that the minimum is unique, by showing that $\bar{V}_j(\zeta) = \sigma^2_{\breve{e}_j}$ implies $\Delta\breve{A}_j(\zeta) = 0$ and $\Delta B_j(\zeta) = 0$. With (6.48) and by applying Parseval's theorem, $\bar{V}_j(\zeta) = \sigma^2_{\breve{e}_j}$ implies

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \Delta x^\top(e^{j\omega},\zeta)^\top \Phi_\kappa(\omega)\Delta x(e^{-j\omega},\zeta)\mathrm{d}\omega = 0, \tag{6.49}$$

with $\Delta x^\top = \begin{bmatrix}\Delta B_j(\zeta) & -\Delta\breve{A}_j(\zeta)\end{bmatrix}$ and $\kappa = \begin{bmatrix} r_a^\top & w^\top\end{bmatrix}^\top$.
By Condition (2) the spectral density $\Phi_\kappa(\omega)$ is positive definite. Therefore equation (6.49) holds only for $\Delta x^\top = 0$ which is satisfied by Condition (3). The global minimum of $\bar{V}_j(\zeta)$ is thus unique for $\breve{A}_j(\zeta) = \breve{A}^0_j$ and $\begin{bmatrix} B_j(\zeta) & \bar{R}_j \end{bmatrix} = \breve{B}^0_j$, with $\bar{R}_j = 0$ for $j = 1,\dots,p$ and $\bar{R}_j$ is a row of R_b for $j = p+1,\dots,L$. ■

6.B Proof of Proposition 6.2

For ease of notation we start with the MIMO notation of the one-step-ahead predictor (6.15)

$$\hat{w}(t|t-1,\eta) = G(\eta)w + Rr + \bar{H}(\eta)\varepsilon_a(\hat{\zeta}^n_N), \tag{6.50}$$

From Proposition 1 we know $\hat{\zeta}^n_N$ is consistent, therefore

$$\varepsilon(\hat{\zeta}^n_N) \to \breve{e} \quad \text{w.p. 1 as } N \to \infty\, \forall t, \tag{6.51}$$

and we can rewrite the one-step-ahead predictor as

$$\hat{w}(t|t-1,\eta) = G(\eta)w + Rr + \bar{H}(\eta)e \tag{6.52}$$

Considering the data generating system in (2.2) the residual becomes

$$\begin{aligned}\varepsilon(t,\eta) &= w(t) - \hat{w}(t|t-1,\eta)\\ &= \Delta G(\eta)w + H^0 e - \bar{H}(\eta)e\\ &= \Delta G(\eta)w + \Delta\bar{H}(\eta)e + \begin{bmatrix} I \\ \Gamma^0 \end{bmatrix} e,\end{aligned} \tag{6.53}$$

where $\Delta G(\eta) = G^0 - G(\eta)$, and $\Delta\bar{H}(\eta) = \begin{bmatrix}\Delta\bar{H}_a(\eta)\\ \Delta\bar{H}_b(\eta)\end{bmatrix}$, with $\Delta\bar{H}_a(\eta) = \bar{H}_a^0 - \bar{H}_a(\eta)$, with $\bar{H}_a = H_a - I$ and $\Delta\bar{H}_b(\eta) = \bar{H}_b^0 - \bar{H}_b(\eta)$, with $\bar{H}_b = H_b - \Gamma$.
The residual per node is written as

$$\varepsilon_j(t,\eta) = \sum_{l\in\mathcal{N}_j} \Delta G_{jl}(\eta)w_l + \sum_{s\in\mathcal{V}_j} \Delta\bar{H}_{js}(\eta)e_s + \breve{e}_j, \tag{6.54}$$

where $\Delta G_{jl}(\eta) = G_{jl}^0 - G_{jl}(\eta)$ is an element of matrix $\Delta G(\eta)$, and $\Delta\bar{H}_{js}(\eta)$ is an element of matrix $\Delta\bar{H}(\eta)$.
The consistency proof consists of two steps

1. Show that the objective function is bounded from below by the noise variance $\bar{V}_j(\theta) := \bar{\mathbb{E}}\varepsilon_j^2(t,\theta) \geqslant \sigma_{\breve{e}_j}^2$, where the minimum is achieved for $\Delta G_{jl} = 0$ and $\Delta\bar{H}_{js} = 0$.

2. Show that the global minimum is unique.

Step 1 By using the property that all ΔG- and ΔH-terms are strictly proper, it follows from (6.54) that

$$\bar{V}_j(\eta) = \bar{\mathbb{E}}\Big[\Big(\sum_{l\in\mathcal{N}_j} \Delta G_{jl}(\eta)w_l + \sum_{s\in\mathcal{V}_j} \Delta\bar{H}_{js}(\eta)e_s\Big)^2\Big] + \sigma_{\breve{e}_j}^2 \tag{6.55}$$

and $\bar{V}_j(\eta) \geqslant \sigma_{\breve{e}_j}^2$ with equality for $\Delta G_{jl} = 0$ and $\Delta\bar{H}_{js} = 0$ for all $l \in \mathcal{N}_j$ and $s \in \mathcal{V}_j$.

Step 2 Showing that the minimum is unique is done by showing that $\bar{V}_j(\eta) = \sigma_{\breve{e}_j}^2$ implies $\Delta G_{jl} = 0$ and $\Delta\bar{H}_{js} = 0$ for all $l \in \mathcal{N}_j$ and $s \in \mathcal{V}_j$. With (6.55) and by applying Parseval's theorem, $\bar{V}_j(\zeta) = \sigma_{\breve{e}_j}^2$ implies

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \Delta x^\top(e^{j\omega},\eta)^\top \Phi_{\bar{\kappa}}(\omega)\Delta x(e^{-j\omega},\eta)\mathrm{d}\omega = 0, \tag{6.56}$$

with $\Delta x^\top = \begin{bmatrix}\Delta G_{jl\in\mathcal{N}_j} & \Delta\bar{H}_{js\in\mathcal{V}_j}\end{bmatrix}$ and $\bar{\kappa} = \begin{bmatrix} w_{\{\mathcal{N}_j\}}^\top & e_{\{\mathcal{V}_j\}}^\top\end{bmatrix}^\top$.
By Condition (1) the spectral density $\Phi_{\bar{\kappa}}$ is positive definite. Therefore equation (6.56) holds only for $\Delta x^\top = 0$. The Parseval's theorem shows the the global minimum of $\bar{V}_j(\eta)$ is unique for $G_{jl}(\eta) = G_{jl}^0$ and $\bar{H}_{js}(\eta) = \breve{H}_{js}^0 - I_{js}$ by Condition (2). ■

6.C Proof of Proposition 6.3

The vector signal κ is written as

$$\kappa = \begin{bmatrix} r_a \\ w \end{bmatrix} = \underbrace{\begin{bmatrix} I & 0 & 0 \\ J_{wa} & J_{wb} & J_{we} \end{bmatrix}}_{J} \begin{bmatrix} r_a \\ r_b \\ e \end{bmatrix} \tag{6.57}$$

with J_{wa}, J_{wb}, J_{we} appropriate transfer function matrices. Since $\rho = \begin{bmatrix} r_a^\top & r_b^\top & e^\top \end{bmatrix}^\top$ is persistently exciting, i.e. $\Phi_\rho(\omega) \succcurlyeq 0$ for all ω, it follows from Lemma 1 in [127] that κ is persistently exciting if and only if matrix J has full row rank. Since full row rank of J is equivalent to a full row rank of $[J_{wb}\ J_{we}]$, the result of Proposition 1 in [127] then shows the equivalence with the condition that there are L vertex disjoint paths from the inputs of $[J_{wb}\ J_{we}]$, i.e. r_b and e, to its outputs, i.e. w. ■

6.D Proof of Proposition 6.4

Similar to the line of reasoning in the proof of Proposition 6.3, the vector signal $\bar{\kappa}$ is written as

$$\bar{\kappa} = \begin{bmatrix} w_{\{\mathcal{N}_j\}} \\ e_{\{\mathcal{V}_j\}} \end{bmatrix} = \underbrace{\begin{bmatrix} J_{wr} & J_{wx} & J_{wv} \\ 0 & 0 & I \end{bmatrix}}_{\bar{J}} \begin{bmatrix} r \\ e_{\{\mathcal{X}_j\}} \\ e_{\{\mathcal{V}_j\}} \end{bmatrix} \tag{6.58}$$

with J_{wr}, J_{wx}, J_{wv} appropriate transfer function matrices. Since $\bar{\rho} = \begin{bmatrix} r^\top & e_{\{\mathcal{X}_j\}}^\top & e_{\{\mathcal{V}_j\}}^\top \end{bmatrix}^\top$ is persistently exciting, i.e. $\Phi_{\bar{\rho}}(\omega) \succcurlyeq 0$ for all ω, it follows from Lemma 1 in [127] that $\bar{\kappa}$ is persistently exciting if and only if matrix $\bar{J}$ has full row rank. Since full row rank of $\bar{J}$ is equivalent to a full row rank of $[J_{wr}\ J_{wx}]$, the result of Proposition 1 in [127] then shows the equivalence with the condition that there are $Cardinal\{\mathcal{N}_j\}$ vertex disjoint paths from the inputs of $[J_{wr}\ J_{wx}]$, i.e. r and $e_{\{\mathcal{X}_j\}}$, to its outputs, i.e. $w_{\{\mathcal{N}_j\}}$. ■

6.E System used in simulations

In the simulation results in Section 6.5 we use the data generating network of which the graph is represented in Figure 6.1. The data generating transfer functions G and H are given by

$$G = \begin{bmatrix} 0 & 0 & 0 & G_{14} & 0 & 0 \\ 0 & 0 & 0 & 0 & G_{25} & 0 \\ G_{31} & 0 & 0 & 0 & G_{35} & 0 \\ 0 & G_{42} & 0 & 0 & 0 & 0 \\ G_{51} & 0 & 0 & 0 & 0 & G_{56} \\ 0 & 0 & G_{63} & 0 & 0 & 0 \end{bmatrix}, \tag{6.59}$$

with the elements of G_{jl}

$$\begin{aligned} G_{14} &= \tfrac{0.38q^{-1}+0.24q^{-2}}{1-1.35q^{-1}+0.54q^{-2}}, & G_{25} &= \tfrac{0.20q^{-1}}{1-1.30q^{-1}+0.60q^{-2}}, \\ G_{31} &= \tfrac{0.39q^{-1}}{1-0.80q^{-1}+0.20q^{-2}}, & G_{35} &= \tfrac{0.16q^{-1}}{1-1.23q^{-1}+0.51q^{-2}}, \\ G_{42} &= \tfrac{-0.30q^{-1}}{1-0.60q^{-1}+0.20q^{-2}}, & G_{51} &= \tfrac{-0.60q^{-1}}{1+0.45q^{-1}+0.12q^{-2}}, \\ G_{56} &= \tfrac{-0.22q^{-1}}{1-1.22q^{-1}+0.46q^{-2}}, & G_{63} &= \tfrac{-0.11q^{-1}}{1-1.49q^{-1}+0.62q^{-2}}, \end{aligned} \tag{6.60}$$

and

$$H = \begin{bmatrix} H_{11} & 0 & 0 & H_{14} \\ 0 & H_{22} & 0 & 0 \\ 0 & H_{32} & H_{33} & 0 \\ 0 & H_{42} & 0 & H_{44} \\ 0 & H_{52} & H_{53} & 0 \\ 0 & H_{62} & 0 & H_{64} \end{bmatrix}, \tag{6.61}$$

with noise rank $p = 4$ and elements

$$\begin{aligned} H_{11} &= \tfrac{1+0.52q^{-1}}{1+0.41q^{-1}}, & H_{14} &= \tfrac{0.41q^{-1}}{1-0.56q^{-1}}, \\ H_{22} &= \tfrac{1+0.44q^{-1}}{1+0.35q^{-1}}, & H_{32} &= \tfrac{-0.56q^{-1}}{1-0.40q^{-1}}, \\ H_{33} &= \tfrac{1-0.20q^{-1}}{1+0.43q^{-1}}, & H_{42} &= \tfrac{0.26q^{-1}}{1-0.62q^{-1}}, \\ H_{44} &= \tfrac{1+0.52q^{-1}}{1+0.45q^{-1}}, & H_{52} &= \tfrac{0.49q^{-1}}{1-0.49q^{-1}}, \\ H_{53} &= \tfrac{1+0.66q^{-1}}{1+0.51q^{-1}}, & H_{62} &= \tfrac{1+0.24q^{-1}}{1+0.53q^{-1}}, \\ H_{64} &= \tfrac{-0.56q^{-1}}{1-0.56q^{-1}+0.21q^{-2}}, \end{aligned} \tag{6.62}$$

where $\Gamma^0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$.

7 CHAPTER

Empirical Bayes Direct Method

In order to identify one system (module) in an interconnected dynamic network, one typically has to solve a Multi-Input-Single-Output (MISO) identification problem that requires identification of all modules in the MISO setup. For application of a parametric identification method this would require estimating a large number of parameters, as well as an appropriate model order selection step for a possibly large scale MISO problem, thereby increasing the computational complexity of the identification algorithm to levels that are beyond feasibility. An alternative identification approach is presented employing regularized kernel-based methods. Keeping a parametric model for the module of interest, we model the impulse response of the remaining modules in the MISO structure as zero mean Gaussian processes (GP) with a covariance matrix (kernel) given by the first-order stable spline kernel, accounting for the noise model affecting the output of the target module and also for possible instability of systems in the MISO setup. Using an Empirical Bayes (EB) approach the target module parameters are estimated through an Expectation-Maximization (EM) algorithm with a substantially reduced computational complexity, while avoiding extensive model structure selection. Numerical simulations illustrate the potentials of the introduced method in comparison with the state-of-the-art techniques for local module identification.

7.1 Introduction

In this chapter we aim at improving the performance of the direct method for dynamic networks, since the direct method exploits both the external excitation

This chapter is based on the publication: K.R. Ramaswamy, G. Bottegal and P.M.J. Van den Hof, “Learning linear models in a dynamic network using regularized kernel-based methods", *Automatica*, Vol. 129, Article 109591, July 2021.

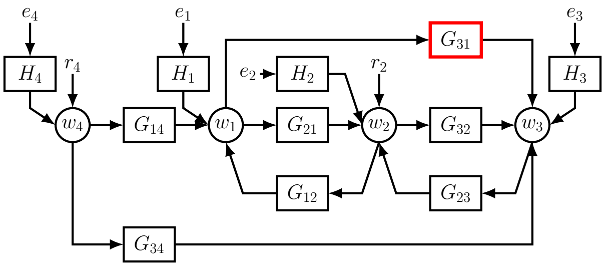


Figure 7.1: Network example with 4 internal nodes, 2 excitation signals $u_2 = r_2, u_4 = r_4$ and a noise sources at each node.

signals and noise signals for data informativity. Assuming a known topology of the network, in [124] it was shown that, in order to identify a given module of interest using the direct method, we have to formulate a multi-input single-output (MISO) identification problem where the inputs of the MISO setup correspond to the inputs of all modules of the network sharing the same output with the module of interest (see Sec. 7.3 for details). A relaxed setup has been provided in [29], where the MISO setup contains only a subset of the above mentioned inputs. This implies that, in both the approaches, to avoid possible bias in the parameter estimates, one has to identify all the modules constituting the MISO structure, bringing in the problem a possibly high number of parameters to be estimated that are of no primal interest to the experimenter. For example, considering the network in Figure 7.1 with the target module of interest for identification being G_{31}, one has to identify G_{31}, G_{32} and G_{34}. Adding to this, a model order selection step needs to be performed to select the number of parameters for each module using complexity criteria like AIC, BIC, or cross validation [77]. For this, it is required to test a number of combination of candidate model orders that increases exponentially with the number of models in the MISO structure, making the model order selection step computationally infeasible (e.g., for 5 modules with FIR model structure and orders from 1 to 5, one has to test 5^5 possible combinations). More importantly, if any of the modules constituting the MISO structure is unstable, the prediction error identification approaches available from the literature cannot be used, since the predictors are unstable. We stress the presence of unstable modules is compatible with stable input-output dynamics in a network. For example, in the network of Figure 7.1 the effect of unstable modules in G_{31} and/or G_{32} could be canceled by suitable controllers G_{23} and/or G_{12}.

In this chapter, we address the aforementioned problems developing an identification method based on non-parametric regularized kernel-based methods that

- identifies a local module through a direct approach, exploiting both the external excitation signals and the disturbance signals for data informativity,
- avoids the complexity of model order selection for large-scale problems,

- reduces the number of nuisance parameters that need to be estimated in local module identification, and
- can be used irrespective of the stability of the modules in the MISO structure, with no need of prior information on possible unstable modules.

In order to develop this method, we build on the following approach. We keep a parametric model for the target module of interest in order to have an accurate description of its dynamics. The impulse responses of the remaining modules in the MISO structure are modeled as zero mean Gaussian Processes (GP), with covariance (or kernel) given by the first-order stable spline kernel [21], [93], which encodes stability and smoothness of the processes. However, we need to handle the prior inclusion of stability property using kernel-based methods under the presence of unstable modules and also incorporate process noise modeling in our framework to avoid increased bias in the estimated target module. We do this by appropriately rewriting the network dynamics.

Using the aforementioned approach, we obtain a Gaussian probabilistic description that depends on a vector of parameters η containing the parameters of the module of interest, the variance of the output noise, and the hyperparamaters characterizing the stable spline kernel. Therefore, estimating η provides the parameters of the target module. This is accomplished by using an Empirical Bayes (EB) approach [80], where η is estimated by maximizing the marginal likelihood of the data, which requires solving a nonlinear non-convex optimization problem. To this end, we use the Expectation-Maximization (EM) method [35], which provides a solution by iterating over simple sub-problems which either admit analytical solutions or require solving scalar optimization problems. Numerical experiments performed on simulated dynamic networks show the potentials of the developed method in comparison with available classical methods.

This chapter is organized as follows. In Section 10.2, the setup of the dynamic network is defined. Section 7.3 provides a summary about the direct method and the extension of this framework using regularized kernel-based methods to end up in a marginal likelihood estimation problem is provided in Section 7.4. Next, we provide the approach and solution to the marginal likelihood problem using EM method. Section 10.7 provides the results of numerical simulations performed on simple dynamic networks, which is followed by the Conclusions. The technical proofs of all results are collected in the Appendix.

7.2 Problem statement

We consider the dynamic network setup in (2.2). We assume that we have collected N measurements of the internal variables $\{w_k(t)\}_{t=1}^{N}$, $k = 1, \dots, L$, and that we are interested in building a model of the module directly linking node i to node j, that is $G_{ji}^0(q)$, using the measurements of the internal variables, and possibly r. To this end, we choose a parameterization of $G_{ji}^0(q)$, denoted as

$G_{ji}(q, \theta)$, that describes the dynamics of the module of interest for a certain parameter vector $\theta_0 \in \mathbb{R}^{n_\theta}$. We assume e to be a Gaussian white noise process. We additionally consider the following assumption on the network:

Assumption 7.1 *The process noise $v_j(t)$ entering the node $w_j(t)$ is uncorrelated with the process noise entering any other node of the network.*

We define $G^0_{jk}, k \in \mathcal{N}_j$ and H^0_j as rational transfer function such that $G^0_{jk}(q) = \frac{B^0_{jk}(q)}{F^0_{jk}(q)}$ and $H^0_j(q) = \frac{C^0_j(q)}{D^0_j(q)}$ where

$$\begin{aligned} B^0_{jk}(q) &= b^0_{jk_1} q^{-1} + \cdots + b^0_{jk_{n_b}} q^{-n_{b_{jk}}}, \\ F^0_{jk}(q) &= 1 + f^0_{jk_1} q^{-1} + \cdots + f^0_{jk_{n_f}} q^{-n_{f_{jk}}}, \\ C^0_j(q) &= 1 + c^0_{j_1} q^{-1} + \cdots + c^0_{j_{n_c}} q^{-n_{c_j}}, \\ D^0_j(q) &= 1 + d^0_{j_1} q^{-1} + \cdots + d^0_{j_{n_d}} q^{-n_{d_j}}, \end{aligned} \tag{7.1}$$

are polynomials, and $n_{b_{jk}}, n_{f_{jk}}, n_{c_{jk}}, n_{d_{jk}}$ are positive integers, and $\mathcal{N}_j$ is the set of node indices k such that $G_{jk} \not\equiv 0$. We now expand the parameterization of $G^0_{ji}(q)$ as $G_{ji}(q, \theta) = \frac{B_{ji}(q,\theta_B)}{F_{ji}(q,\theta_F)} = \frac{B_{ji}(q,\theta_B)}{1+\bar{F}_{ji}(q,\theta_F)}$ with $\theta = \begin{bmatrix}\theta_B^\top & \theta_F^\top\end{bmatrix}^\top$, where θ_B and θ_F are the parameterized coefficients of polynomials $B^0_{ji}(q)$ and $F^0_{ji}(q)$ respectively as in Eq. (7.1) (i.e. $\theta_B = [b_{ji_1} \quad \cdots \quad b_{ji_{n_b}}]^\top$ and $\theta_F = [f_{ji_1} \quad \cdots \quad f_{ji_{n_f}}]^\top$).

7.3 The standard MISO direct method

Following the definition of a dynamic network, each scalar internal variable can be described as:

$$w_j(t) = \sum_{k \in \mathcal{N}_j} G^0_{jk}(q) w_k(t) + u_j(t) + v_j(t) \tag{7.2}$$

The above equation represents a MISO structure and is the starting point of the methodology presented in this chapter, which is based on extending the MISO direct method [124] (see Chapter 2). In the standard direct method for dynamic networks [124], we consider the one-step-ahead predictor [77] of $w_j(t)$:

$$\begin{aligned} \hat{w}_j(t|t-1;\theta) =& \big(1 - (H_j)^{-1}(q,\theta)\big) w_j(t) + (H_j)^{-1}(q,\theta) G_{ji}(q,\theta) w_i(t) \\ &+ (H_j)^{-1}(q,\theta)\Big(\sum_{k \in \mathcal{N}_j \backslash \{i\}} G_{jk}(q,\theta) w_k(t) + u_j(t)\Big) \end{aligned}$$

which is a function of the parameter vector θ. Not only the target module, but also the modules $G^0_{jk}(q)$, $k \in \mathcal{N}_j \backslash \{i\}$, and the noise model $H^0_j(q)$, are suitably parameterized with additional parameters. The parameter vector of interest θ is identified by minimizing the sum of the squared prediction error $\varepsilon_j(t) = w_j(t) - \hat{w}_j(t|t-1;\theta)$. We note that in this formulation, the prediction error depends also on the additional parameters entering the remaining modules and the noise model, which need to be identified to guarantee consistent estimates of θ. Therefore, the total number of parameters may grow large if the cardinality of $\mathcal{N}_j$ is large, with a detrimental effect on the variance of the estimate of θ in the case where N is not very large.

7.4 The developed Empirical Bayes identification technique

We now discuss how to use regularized kernel-based methods to avoid parameterization of the additional modules (all modules except the target module) in the MISO structure. We define the following quantities:

$$S_j^0(q) := 1 - (H_j^0)^{-1}(q) \quad , \quad S_{jk}^0(q) := (H_j^0)^{-1} G_{jk}^0(q) \, .$$

Considering the above definitions, Eq. (7.2) can be re-written as

$$\begin{aligned} w_j(t) &= \hat{w}_j(t|t-1) + e_j(t), \\ &= S_j^0(q) w_j(t) + (1 - S_j^0(q))(G_{ji}^0(q) w_i(t) + u_j(t)) + \sum_{k \in \mathcal{N}_j \backslash \{i\}} S_{jk}^0(q) w_k(t) + e_j(t), \end{aligned} \tag{7.3}$$

where we isolate the target module $G_{ji}^0(q)$. A main challenge when using kernel methods for LTI system identification is that typically a prior knowledge on the stability of the predictor filters in (7.3) is imposed to reduce the MSE of the estimated impulse response of the system (see [93, 101]). When all systems (i.e. $G_{jk}, k \in \mathcal{N}_j$) are stable, as assumed in [101], the predictor filters in (7.3) are stable and the setup in (7.3) lends itself for kernel-based estimation of the predictor filters. However, when some or all systems in the MISO structure are not stable, the imposition of prior knowledge on stability is not possible unless we suitably rewrite the network dynamics in (7.2).

Proposition 7.1 *Consider the network equation of the output node signal* $w_j(t)$ *in* (7.2). *The network equation can be represented in an alternative way as*[a]*,*

$$\begin{aligned} w_j(t) &= M_j(q) w_j(t) - (1 - M_j(q)) \bar{F}_{ji}(q) w_j(t) \\ &+ (1 - M_j(q)) B_{ji}(q) w_i(t) + \sum_{k \in \mathcal{N}_j \backslash \{i\}} M_{jk}(q) w_k(t) + \bar{e}_j(t), \end{aligned} \tag{7.4}$$

where $M_\star(q)$ *are strictly proper predictor filters,* $B_{ji}(q)$ *and* $\bar{F}_{ji}(q) = -(1 - F_{ji}(q))$ *are stable polynomials representing* $G_{ji}(q)$*, and* $\bar{e}_j(t)$ *is a Gaussian white noise with variance* $\bar{\sigma}_j^2$*.*

Proof: *Collected in the appendix. The expressions for $M_\star(q)$ are provided in the appendix.* □

[a] from now on superscript ^{0} is dropped for convenience.

Since all the predictor filters in the rewritten network dynamics are stable, this formulation lends itself to the Bayesian approach [101], as described in the subsequent sections.

7.4.1 Vector description of the dynamics

In order to apply a kernel-based method to (7.4), we are going to formulate a vector description of the network dynamics for the available N measurements. For notation purposes, we consider N-dimensional vectors b_{ji} and f_{ji} (which will also depend on θ, although we will keep this dependence tacit) which are the parameterized coefficients of $B_{ji}(q,\theta_B)$ and $\bar{F}_{ji}(q,\theta_F)$ respectively stacked with zeros (i.e. $b_{ji} = [\theta_B^\top \quad \mathbf{0}^\top]^\top$ and $f_{ji} = [\theta_F^\top \quad \mathbf{0}^\top]^\top$). Similarly, we define the vector m_k, $k \in \mathcal{N}_j \backslash \{i\}$, and m_j as the vectors containing the first l coefficients of the impulse responses of $M_{jk}(q)$, $k \in \mathcal{N}_j \backslash \{i\}$, and $M_j(q)$, respectively. The integer l is chosen large enough to ensure $m_k(l+1), m_j(l+1) \simeq 0$.

Lemma 7.1 *Let the vector notation for the node $w_j(t)$ be $w_j := \begin{bmatrix} w_j(1) & \dots & w_j(N)\end{bmatrix}^T$. Considering the parameterization of G_{ji}^0, the network dynamics in (7.4) can be represented in the vector form as:*

$$w_j = \tilde{W} m_j + \mathbf{W}_{ji} g_{ji} + \sum_{k \in \mathcal{N}_j \backslash \{i\}} W_k m_k + \bar{e}_j, \tag{7.5}$$

where $\mathbf{g}_{ji} = [b_{ji}^\top \quad f_{ji}^\top]^\top$ and $\bar{e}_j$ is the vectorized noise. $\tilde{W}$, $\mathbf{W}_{ji}$ and W_k are Toeplitz matrices constructed from measurements of the nodes in the MISO structure.

Proof: *We denote by $W_k \in \mathbb{R}^{N \times l}$ the lower triangular Toeplitz matrix of the vector $\overrightarrow{w}_k := \begin{bmatrix} 0 & w_k(1) & \dots & w_k(N-1)\end{bmatrix}^T$, $k \in \{\mathcal{N}_j \cup j\} \backslash \{i\}$ and $W_\ell^N \in \mathbb{R}^{N \times N}$ the lower triangular Toeplitz matrix of the vector $\overrightarrow{w}_\ell := \begin{bmatrix} 0 & w_\ell(1) & \dots & w_\ell(N-1)\end{bmatrix}^T$ where $\ell \in \{i, j\}$. Similarly, we denote by $\overleftrightarrow{W}_\ell \in \mathbb{R}^{N \times l}$ the lower triangular Toeplitz matrix of the vector $\overleftrightarrow{w}_\ell := \begin{bmatrix} 0 & 0 & -w_\ell(1) & \dots & -w_\ell(N-2)\end{bmatrix}^T$, $\ell \in \{i, j\}$. Also G_b and G_f are the lower triangular Toeplitz matrix of b_{ji} and f_{ji} respectively. Considering the parameterization of G_{ji}^0 and the above established notations, we can rewrite the network dynamics in (7.4) as (7.5) where $\tilde{W} := W_j + G_b \overleftrightarrow{W}_i - G_f \overleftrightarrow{W}_j$, $\mathbf{W}_{ji} = [W_i^N \quad -W_j^N]$, $g_{ji} = [b_{ji}^\top \quad f_{ji}^\top]^\top$ and $\bar{e}_j$ is the vectorized noise.* □

7.4.2 Modeling strategy for the additional modules

We now have a vector description of the module dynamics where we have isolated the objective of the identification method, namely g_{ji}, from the non-interesting nuisance terms, namely m_k and m_j. As the next step, we discuss our modeling strategy with the use of regularized kernel-based methods. Our goal is to limit the number of parameters necessary to describe w_j in (7.5), in order to increase the accuracy of the estimated parameter vector of interest θ. In order to achieve this, we keep a parametric model for g_{ji} (accounting for the zeros in g_{ji}), while the remaining impulse responses in (7.5) are modeled with non-parametric models as zero mean Gaussian processes. The choice of Gaussian processes is motivated by the fact that, with a suitable choice of the prior covariance matrix (usually referred to as kernel), we can get a significant reduction in the variance of the estimated impulse responses [93]. Therefore, we model m_j and m_k, $k \in \mathcal{N}_j \backslash \{i\}$, as independent[1] zero mean Gaussian processes (vectors in this case). The choice of the covariance matrix (kernel) of these vectors are given by the *First-order Stable Spline kernel* whose general structure is given as,

$$[K_\beta]_{x,y} = \lambda \beta^{\max(x,y)} \,, \tag{7.6}$$

where $\beta_j \in [0,1)$ is a *hyperparameter* that regulates the decay velocity of the realizations of the corresponding Gaussian vector, while $\lambda \geqslant 0$ tunes their amplitude. x, y represent the element of the matrix. The choice of this kernel is motivated by the fact that it enforces favorable properties such as stability and smoothness in the estimated impulse responses [91], [92]. Therefore, we have that

$$m_j \sim \mathcal{N}(0, \lambda_j K_{\beta_j}) \tag{7.7}$$

$$m_k \sim \mathcal{N}(0, \lambda_k K_{\beta_k}) \quad , k \in \mathcal{N}_j \backslash \{i\}, \tag{7.8}$$

where we have assigned different hyperparameters to the impulse response priors to guarantee flexible enough models.

7.4.3 Incorporating Empirical Bayes approach

We define

$$\mathbf{m} := \begin{bmatrix} m_j^\top & m_{k1}^\top & m_{k2}^\top & \dots & m_{kp}^\top \end{bmatrix}^\top \,, \tag{7.9}$$

where $k_1, \dots, k_p$ are the elements of the set $\mathcal{N}_j \backslash \{i\}$, and

$$\mathbf{W} := \begin{bmatrix} \tilde{W} & W_{k1} & W_{k2} & \dots & W_{kp} \end{bmatrix} \,, \tag{7.10}$$

$$\mathbf{K} := \operatorname{diag}\{\lambda_j K_{\beta_j}, \lambda_{k1} K_{\beta_{k1}}, \dots, \lambda_{kp} K_{\beta_{kp}}\}. \tag{7.11}$$

[1] It is clear that these impulse responses share some common dynamics given by the pre-multiplication with the inverse of the noise model $H_j(q)$. However, for computational purposes it is convenient to treat the impulse responses as independent. Furthermore, incorporating the mutual dependence through a suitable choice of prior distribution seems a non-trivial problem that deserves a thorough analysis that is outside the scope of the work in this chapter.

It is important to note that $\tilde{W}$ depends on b_{ji} and f_{ji}. Using the above, we can rewrite (7.5) in compact form as

$$w_j = \mathbf{W}\mathbf{m} + \mathbf{W}_{ji}\mathbf{g}_{ji} + \bar{e}_j\,. \tag{7.12}$$

Having assumed a Gaussian distribution of the noise, we can write the joint probabilistic description of m and w_j [91], which is jointly Gaussian, as:

$$p\left(\begin{bmatrix}\mathbf{m}\\ w_j\end{bmatrix};\eta\right) \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{0}\\ \mathbf{W}_{ji}\mathbf{g}_{ji}\end{bmatrix},\begin{bmatrix}\mathbf{K} & \mathbf{K}\mathbf{W}^\top\\ \mathbf{W}\mathbf{K} & \mathbf{P}\end{bmatrix}\right), \tag{7.13}$$

where

$$\mathbf{P} := \bar{\sigma}_j^2 I_N + \tilde{W}\lambda_j K_{\beta_j}\tilde{W} + \sum_{k\in\mathcal{N}_j\setminus\{i\}} W_k\lambda_k K_{\beta_k}{W_k}^\top, \tag{7.14}$$

and this pdf depends upon the vector of parameters

$$\eta := \begin{bmatrix}\theta^\top & \lambda_j & \lambda_{k1} & \dots & \lambda_{kp} & \beta_j & \beta_{k1} & \dots & \beta_{kp} & \bar{\sigma}_j^2\end{bmatrix},$$

which contains the parameter vector of the target module, the hyperparameters of the kernels of the impulse response models of the other modules, and the variance of the “dummy" noise corrupting $w_j(t)$. Therefore, we focus on the estimation of η, since it contains the parameter of interest θ. To this end, we apply an Empirical Bayes (EB) approach. We consider the marginal pdf of w_j, which is obtained by integrating out the dependence on $\mathbf{m}$ [91] and corresponds to

$$p(w_j;\eta) \sim \mathcal{N}(\mathbf{W}_{ji}\mathbf{g}_{ji},\mathbf{P}). \tag{7.15}$$

Then, the estimate of η is obtained by maximizing the marginal likelihood of w_j, namely

$$\begin{aligned}\hat{\eta} &= \arg\max_{\eta} p(w_j;\eta)\\ &= \arg\min_{\eta} \log\det\mathbf{P} + \big(w_j - \mathbf{W}_{ji}\mathbf{g}_{ji}\big)^\top\mathbf{P}^{-1}\big(w_j - \mathbf{W}_{ji}\mathbf{g}_{ji}\big).\end{aligned} \tag{7.16}$$

Solving this optimization problem can be a cumbersome task, because it is a nonlinear one and involves a large number of decision variables. In the next section, we study how to solve the marginal likelihood problem through a dedicated iterative scheme.

7.5 Solution to the marginal likelihood problem

In this section, we focus on solving the problem in (10.27) by deriving an iterative solution scheme through the EM algorithm [35]. For this, we need to first define a *latent variable* whose estimation simplifies the computation of the marginal likelihood. In our case, a natural choice is m. Then, the solution to (10.27) using the EM algorithm is obtained by iterating among the following two steps:

- *E-Step:* Given an estimate $\hat{\eta}^{(n)}$ computed at the n^{th} iteration, compute

  $$Q^{(n)}(\eta) = \mathbb{E}[\log p(w_j, \mathbf{m}; \eta)]\,, \tag{7.17}$$

  where the expectation of the joint log-likelihood of w_j and is taken with respect to the posterior $p(|w_j; \hat{\eta}^{(n)})$;

- *M-Step:* Update $\hat{\eta}$ by solving

  $$\hat{\eta}^{(n+1)} = \arg\max_{\eta} Q^{(n)}(\eta)\,. \tag{7.18}$$

When iterating among the above steps, convergence to a stationary point of the marginal likelihood is ensured [16]. This stationary point can be a local or global maximum of the objective function. In the next section, we show that we clearly get an advantage in solving the original marginal likelihood problem (10.27) by repetitively solving (10.29) using the EM algorithm. We show that, when we use the EM method, the nonlinear optimization problem becomes a problem of iteratively constructing analytical solutions and solving scalar optimization problems, which significantly simplifies solving (10.27).

7.5.1 Computation of E-step

First we focus on the E-step. The posterior distribution of $\mathbf{m}$ given w_j and an estimate of η is Gaussian and corresponds to (see also [2]),

$$p(\mathbf{m}|w_j; \eta) \sim \mathcal{N}\big(\mathbf{C}(w_j - \mathbf{W}_{ji}\mathbf{g}_{ji}), \mathbf{P}_m\big) \tag{7.19}$$

where

$$\mathbf{P}_m = \left(\frac{\mathbf{W}^\top \mathbf{W}}{\bar{\sigma}_j^2} + \mathbf{K}^{-1}\right)^{-1} \quad ; \quad \mathbf{C} = \frac{\mathbf{P}_m \mathbf{W}^\top}{\bar{\sigma}_j^2}.$$

Let $\hat{\mathbf{m}}^{(n)}$ and $\hat{\mathbf{P}}_m^{(n)}$ be the posterior mean and covariance of $\mathbf{m}$ obtained from (7.19) using $\hat{\eta}^{(n)}$. We define

$$\hat{\mathbf{M}}^{(n)} := \hat{\mathbf{P}}_m^{(n)} + \hat{\mathbf{m}}^{(n)}\hat{\mathbf{m}}^{(n)\top},$$

and consider its $l \times l$ diagonal blocks, which we denote by $\hat{\mathbf{M}}_j^{(n)}$, $\hat{\mathbf{M}}_{k_1}^{(n)}$, $\ldots$, $\hat{\mathbf{M}}_{k_p}^{(n)}$, respectively. These sub-matrices correspond to the posterior second moments of the estimated impulse responses $\hat{m}_j^{(n)}$,$\hat{m}_{k_1}^{(n)}$, $\ldots$, $\hat{m}_{k_p}^{(n)}$.

The following lemma provides the structure of the function $Q^{(n)}(\eta)$ for the particular situation of our setup in (10.27).

Lemma 7.2 *Let* $\hat{\eta}^{(n)}$ *be the estimate of* η *at the* n^{th} *iteration of the EM algorithm according to* (10.29)*. Then*

$$Q^{(n)}(\eta) = Q_0^{(n)}(\bar{\sigma}_j^2, \theta) + \sum_{k \in \{\mathcal{N}_j \cup j\} \backslash \{i\}} {Q_m}_k^{(n)}(\lambda_k, \beta_k) \tag{7.20}$$

where

$$\begin{aligned} Q_o^{(n)}(\bar{\sigma}_j^2, \theta) = &- N \log(\bar{\sigma}_j^2) - \frac{1}{\sigma_j^2} \Big[w_j^\top w_j - 2 w_j^\top \mathbf{W}_{ji} \mathbf{g}_{ji} + \\ & \mathbf{g}_{ji}^\top \mathbf{W}_{ji}^\top \mathbf{W}_{ji} \mathbf{g}_{ji} - 2 w_j^\top \mathbf{W} \hat{\mathbf{m}}^{(n)} \\ & + 2 \mathbf{g}_{ji}^\top \mathbf{W}_{ji}^\top \mathbf{W} \hat{\mathbf{m}}^{(n)} + \operatorname{tr}\big(\mathbf{W}^\top \mathbf{W} \hat{\mathbf{M}}^{(n)}\big) \Big], \end{aligned} \tag{7.21}$$

$${Q_m}_k^{(n)}(\lambda_k, \beta_k) = - \log \det(\lambda_k K_{\beta_k}) - \operatorname{tr}\big((\lambda_k K_{\beta_k})^{-1} \hat{\mathbf{M}}_k^{(n)}\big). \tag{7.22}$$

□

Proof: See the appendix.

The function $Q^{(n)}(\eta)$ is the summation of several terms that depend on different components of the vector η. In particular, we have a term of the type ${Q_m}_k^{(n)}(\lambda_k, \beta_k)$ for each module in the MISO structure, and a term $Q_0^{(n)}(\bar{\sigma}_j^2, \theta)$ for the module of interest and the noise variance. Therefore, the update of η according to (10.29) splits into a number of independent and smaller optimization problems.

7.5.2 Computation of M-step

We now focus on the M-step according to (10.29). From (7.20), it is evident that each kernel hyperparameters can be updated independently of the rest of the parameters. The following theorem, inspired by [14] and [38], shows how to update the kernel hyperparameters.

Theorem 7.1 *For the update of each kernel's hyperparameters that requires maximizing* (10.67)*, we define*

$${Q_\beta}_k^{(n)}(\beta_k) = \log \det(K_{\beta_k}) + l \log \left(\operatorname{tr}\big((K_{\beta_k})^{-1} \hat{\mathbf{M}}_k^{(n)}\big) \right) \tag{7.23}$$

for $k \in \{\mathcal{N}_j \cup j\} \backslash i$*. Then the updates are obtained as,*

$$\hat{\beta_k}^{(n+1)} = \arg\min_{\beta_k \in [0,1)} {Q_\beta}_k^{(n)}(\beta_k); \tag{7.24}$$

$$\hat{\lambda_k}^{(n+1)} = \frac{1}{l} \operatorname{tr}\big((K_{\hat{\beta}_k^{(n+1)}})^{-1} \hat{\mathbf{M}}_k^{(n)}\big) \tag{7.25}$$

□

Proof: See the appendix.

The optimization problem in (10.45) can be difficult to perform in practice when the determinant of the kernel has a very low value or when the inversion of the kernel becomes difficult. To tackle this, we exploit the factorization of the *first order stable spline kernel* as in [14] by writing $K_{\beta_k} = LD(\beta)L^T$, where L is lower-triangular with known entries (essentially, an "integrator") and $D(\beta)$ is diagonal with entries essentially being an exponential functions of β. Using the above technique also increases the computation speed of the algorithm.

We note that from (10.46) that we get closed-form solutions for all λ_k, $k \in \{\mathcal{N}_j \cup j\}\backslash\{i\}$, while the β_k, $k \in \{\mathcal{N}_j \cup j\}\backslash\{i\}$, can be updated by solving scalar optimization problems in the domain $[0, 1)$, as detailed in (10.45). Therefore, the hyperparameters update turns out to be a computationally fast operation.

We now turn our attention to the update of θ and $\bar{\sigma}_j^2$ for which we need to maximize (10.69). We notice that the optimum with respect to θ does not depend on the optimal value of $\bar{\sigma}_j^2$. Then, we can first update θ and then use its optimal value to update $\bar{\sigma}_j^2$. How to update θ is explained in the following theorem.

Theorem 7.2 *The estimate of the parameter vector θ is updated by solving the quadratic problem*

$$\hat{\theta}^{(n+1)} = \arg\min_{\theta} \left[\mathbf{g}_{ji}^{\top} \hat{\mathbf{A}}^{(n)} \mathbf{g}_{ji} - 2\hat{\mathbf{b}}^{(n)\top} \mathbf{g}_{ji} \right] \tag{7.26}$$

that has a closed form solution given by

$$\hat{\theta}^{(n+1)} = \left(M^{\top} \hat{\mathbf{A}}^{(n)} M \right)^{-1} M^{\top} \hat{\mathbf{b}}^{(n)}, \tag{7.27}$$

where $\hat{\mathbf{A}}^{(n)}$ and $\hat{\mathbf{b}}^{(n)}$ are computed using the current estimates $\hat{\mathbf{m}}^{(n)}$ and $\hat{\eta}^{(n)}$, and $\mathbf{g}_{ji} = M\theta$ where $M \in \mathbb{R}^{2N \times n_\theta}$ is a matrix with 1 or 0 as its elements. □

Proof: See the appendix.

Therefore, the parameter vector of the target module is updated by solving the analytical expression (7.27).

Remark 7.1 *An additional advantage of the method developed in this chapter is that it relies on iteratively solving a quadratic least squares problem to find the solution for the parameters of the target module θ rather than solving a non-linear least squares problem as in [101], making the method computationally more efficient.*

We are left with updating $\bar{\sigma}_j^2$, which is given in the next theorem.

Theorem 7.3 *Let $\hat{\mathbf{g}}_{ji}^{(n+1)}$, $\hat{\mathbf{W}}^{(n+1)}$ be constructed by inserting $\hat{\theta}^{(n+1)}$ in the general expression of $\mathbf{g}_{ji}$ and $\mathbf{W}$. Then*

$$(\hat{\bar{\sigma}}_j^2)^{(n+1)} = \frac{1}{N}\Big[\left\|w_j - \mathbf{W}_{ji}\hat{\mathbf{g}}_{ji}^{(n+1)}\right\|_2^2 - 2w_j^\top \hat{\mathbf{W}}^{(n+1)}\hat{\mathbf{m}}^{(n)} + 2\hat{\mathbf{g}}_{ji}^{(n+1)\top}\mathbf{W}_{ji}^\top \hat{\mathbf{W}}^{(n+1)}\hat{\mathbf{m}}^{(n)} + \mathrm{tr}\big(\hat{\mathbf{W}}^{(n+1)\top}\hat{\mathbf{W}}^{(n+1)}\hat{\mathbf{M}}^{(n)}\big)\Big]$$

□

Proof: See the appendix.

Thus, a closed-form solution for the estimate of the noise variance is also obtained.

> **Remark 7.2** *We estimate the "dummy" noise variance $\bar{\sigma}_j^2 = |f_{a_{nf}}|^2\sigma_j^2$, that is a scaled version of the original output noise power in the network. If there are no unstable systems in the MISO setup, then $\bar{\sigma}_j^2$ will be σ_j^2. This will be verified with numerical simulations in section 10.7.*

All-in-all, we have obtained a fast iterative procedure that provides a local solution to the marginal likelihood problem (10.27). All the updates follow simple rules that allow for fast iterative computation. Algorithm 1 summarizes the steps to follow to obtain $\hat{\eta}$ and therefore $\hat{\theta}$.

Algorithm 1 Algorithm for local module identification in dynamic networks

Input: $\{w_k(t)\}_{t=1}^N$, $k = 1, \dots, p$
Output: $\hat{\theta}$

```
1. Set $n = 0$, Initialize $\hat{\eta}^{(0)}$.
2. Compute $\hat{\mathbf{P}}_m^{(n)}$, $\hat{\mathbf{C}}^{(n)}$, $\hat{\mathbf{M}}^{(n)}$ and $\hat{\mathbf{m}}^{(n)}$.
3. Update hyperparameters $\hat{\beta_k}^{(n+1)}$ and $\hat{\lambda_k}^{(n+1)}$ using (10.45) and (10.46) respectively for all $k \in \{\mathcal{N}_j \cup \{j\}\}\backslash\{i\}$.
4. Update $\hat{\theta}^{(n+1)}$ by solving (7.27).
5. Update $\hat{\bar{\sigma}}_j^{2(n+1)}$ as in Theorem 7.3.
6. Set $\hat{\eta}^{(n+1)} = \big[\hat{\theta}^{\top(n+1)}\ \hat{\lambda_j}^{(n+1)}\ \hat{\lambda_{k_1}}^{(n+1)}\ \dots\ \hat{\lambda_{k_p}}^{(n+1)}\ \hat{\beta_j}^{(n+1)}\ \hat{\beta_{k_1}}^{(n+1)}\ \dots\ \hat{\beta_{k_p}}^{(n+1)}\ \hat{\bar{\sigma}}_j^{2(n+1)}\big]^\top$.
7. Set $n = n + 1$.
8. Repeat from steps (2) to (7) until convergence.
```

The initialization can be done by randomly choosing η considering the constraints of hyperparameters. The convergence criterion for the algorithm depend on the value of $\frac{\|\hat{\eta}^{(n)}-\hat{\eta}^{(n-1)}\|}{\|\hat{\eta}^{(n-1)}\|}$. This value should be small for convergence so that the algorithm can be terminated. A value of 10^{-2} is considered for the numerical simulations in Section 10.7. The other convergence criterion is the maximum number of iterations. It is taken as 50.

Remark 7.3 *Being applicable to a MISO identification setup, the introduced method can also be inherently used for parametric SISO identification, where the process noise modeling is now simplified by avoiding the model order selection and reducing the number of parameters of the noise model to two (which are the hyperparamters λ_j, β_j).*

Remark 7.4 *We notice that:*

- *The method does not require prior information about the stability of the systems $G_{jk}, k \in \mathcal{N}_j$ and the number of unstable poles in the systems.*
- *According to [29], in view of consistency of the target module estimate, it is not necessary to take all nodes $w_k, k \in \mathcal{N}_j$ as the inputs in the MISO structure, but it is sufficient to take a subset of nodes in $\mathcal{N}_j$ as inputs such that every parallel path[a] from w_i to w_j and every loop around w_j passes through a selected input. This may lead to confounding variables which can be handled using additional inputs[30]. At the same time, in view of an appropriate bias-variance trade off, especially under limited data circumstances, it could be attractive to include more predictor inputs than the ones that are strictly necessary for achieving consistency. While the algorithm presented in this chapter can be applied to any choice of such MISO structure, we have formulated the results for the situation where all nodes $w_k, k \in \mathcal{N}_j$ are taken as inputs.*

[a] a path from w_i to w_j that does not pass through G_{ji}.

7.5.3 Non-parametric identification of modules in the MISO structure

In this section we slightly adapt the developed method to obtain a non-parametric estimate of the target module. For this, we rewrite the network equation (7.2) as,

$$w_j(t) = M_j(q)w_j(t) + \sum_{k\in\mathcal{N}_j} M_{jk}(q)w_k(t) + \bar{e}_j(t) \tag{7.28}$$

with

$$M_j(q) \quad := \quad 1 - \left((H_j)^{-1}(q) \frac{F_a(q)}{F_a^\star(q)} \right) \quad , \tag{7.29}$$

$$M_{jk}(q) \quad := \quad (H_j)^{-1} \frac{\prod_{\ell \in \mathcal{N}_j \setminus \{k\}} F_{j\ell}^{(a)}(q)}{F_a^\star(q)} \frac{B_{jk}(q)}{F_{jk}^{(s)}(q)} , \tag{7.30}$$

where $M_{jk}(q)$ and $M_j(q)$ are stable. Following the similar approach as introduced before, but modeling the impulse response of all the modules (including m_i of M_{ji} that represents the target module) as zero mean Gaussian processes with the prior covariance matrix represented by the First-order stable spline kernel, we end up in an iterative algorithm to estimate the parameter vector η which contains the hyperparameters λ_k, β_k where $k \in \mathcal{N}_j$ and the noise variance $\bar{\sigma}_j^2$. Since we are not paramterizing any modules, we do not have θ in the parameter vector η. The solutions for the β's and λ's at each iteration are given by (10.45) and (10.46) respectively. The solution to $\bar{\sigma}_j^2$ at each iteration is given by,

$$(\hat{\bar{\sigma}}_j^2)^{(n+1)} = \frac{1}{N} \left[\|w_j\|_2^2 - 2w_j^\top \mathbf{W}\hat{\mathbf{m}}^{(n)} + \operatorname{tr}\left(\mathbf{W}^\top \mathbf{W} \hat{\mathbf{M}}^{(n)}\right) \right]$$

where

$$\mathbf{W} := \begin{bmatrix} W_j & W_{k1} & W_{k2} & \dots & W_{kp} \end{bmatrix} .$$

The above solution is equivalent to the solution of $\hat{\bar{\sigma}}_j^2$ in Theorem 7.3, however without the terms that are function of θ (i.e. $\mathbf{g}_{ji}, G_b, G_f, \mathbf{W}_{ji}\mathbf{g}_{ji}$). Thus we will end up in the same Algorithm 1, however with steps related to θ (step 4) being not applicable. The posterior mean of $m_k, k \in \mathcal{N}_j$ and m_j obtained using (7.19) (neglecting the effect of $W_{ji}g_{ji}$) for the converged η provides us the impulse response of M_{jk} and M_j respectively. From these, the impulse response estimates of the modules $G_{jk}, k \in \mathcal{N}_j$ can be obtained. Thus we obtain a non-parametric identification method to identify all the modules in the MISO structure as a derived result of the earlier developed identification technique.

7.6 Numerical simulations

Numerical simulations are performed to evaluate the performance of the developed method, which we abbreviate as Empirical Bayes Direct Method (EBDM). The simulations are performed on the dynamic network depicted in Figure 7.1. The goal is to identify G_{31}^0. To show the effectiveness of the introduced method and its flexibility to handle stable and unstable modules with a single unified identification framework, we perform the simulations for two different cases:

1. Case 1: All modules in the MISO setup are stable.
2. Case 2: The modules in the MISO setup including the target module can be stable or unstable.

The results of the numerical simulations are presented below.

7.6.1 Case study 1

The EBDM is compared with the standard direct method and the two-stage method (see [124] for details). The network modules of network in Figure 7.1 are given by

$$\begin{aligned}
G_{31}^0 &= \frac{q^{-1}+0.05q^{-2}}{1+q^{-1}+0.6q^{-2}} = \frac{b_1^0 q^{-1}+b_2^0 q^{-2}}{1+a_1^0 q^{-1}+a_2^0 q^{-2}}\\
G_{32}^0 &= \frac{0.09q^{-1}}{1+0.5q^{-1}};\\
G_{34}^0 &= \frac{1.184q^{-1}-0.647q^{-2}+0.151q^{-3}-0.082q^{-4}}{1-0.8q^{-1}+0.279q^{-2}-0.048q^{-3}+0.01q^{-4}};\\
G_{14}^0 &= G_{21}^0 = \frac{0.4q^{-1}-0.5q^{-2}}{1+0.3q^{-1}}; H_1^0 = \frac{1}{1+0.2q^{-1}};\\
G_{12}^0 &= G_{23}^0 = \frac{0.4q^{-1}+0.5q^{-2}}{1+0.3q^{-1}}; H_2^0 = \frac{1}{1+0.3q^{-1}}\\
H_3^0 &= \frac{1-0.505q^{-1}+0.155q^{-2}-0.01q^{-3}}{1-0.729q^{-1}+0.236q^{-2}-0.019q^{-3}}; H_4^0 = 1.
\end{aligned}$$

We run 50 independent Monte Carlo experiments where the data is generated using known reference signals $r_2(t)$ and $r_4(t)$ that are realizations of white noise with unit variance. The number of data samples is N = 500. The noise sources $e_1(t)$, $e_2(t)$, $e_3(t)$ and $e_4(t)$ have variance 0.05, 0.08, 0.5, 0.1, respectively. We assume that we know the model order of $G_{31}^0(q)$. In the case of direct method, we solve a 3-input/1-output MISO identification problem with $w_1(t)$, $w_2(t)$ and $w_4(t)$ as inputs. In the two-stage method, the projections of the three inputs on external signals $r_2(t)$ and $r_4(t)$ are used as inputs to the MISO identification problem. For both these methods, we consider the case where a model order selection of all the modules in the MISO structure (except for the target module) is required, and the case where the model orders are known. Moreover, in order to improve the accuracy of the identified module in the two-stage method, we identify a noise model even though it is not necessary for consistency.

Figure 7.4 shows the estimated impulse response at the end of each MC simulation using the EBDM. It can be verified that, in line with our framework, the estimates provide the description of the dynamics of M_j, $M_{jk}, k \in \mathcal{N}_j$ and G_{ji}. To evaluate the performance of the methods, we use the standard goodness-of-fit metric,

$$\text{Fit} = 1 - \frac{\left\|g_{ji}^0 - \hat{g}_{ji}\right\|_2}{\left\|g_{ji}^0 - \bar{g}_{ji}\right\|_2},$$

where g_{ji}^0 is the true value of the impulse response of G_{ji}^0, $\hat{g}_{ji}$ is the impulse response of the estimated target module and $\bar{g}_{ji}$ is the sample mean of g_{ji}^0. The box plots of the fits of the impulse response of $G_{31}(q)$ are shown in Figure 7.2,

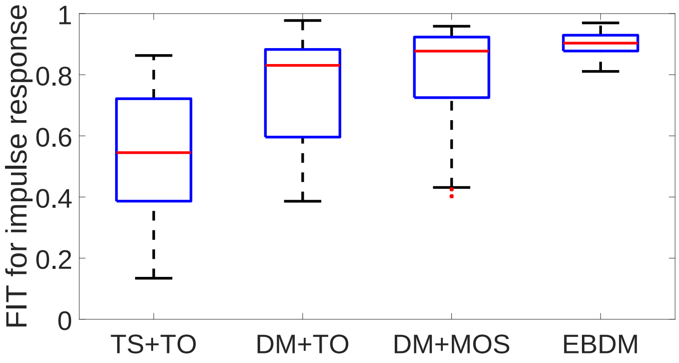


Figure 7.2: Box plot of the fit of the impulse response of $\hat{G}_{31}$ obtained by the Two-stage method, Direct method and EBDM. Number of data samples used for estimation is $N = 500$.

where we have compared the two-stage method with true model orders ('TS+TO'), the direct method with true model orders and model orders selected via BIC ('DM+TO' and 'DM+MOS', respectively), and the Empirical Bayes Direct Method ('EBDM'). As for the latter, we choose $l = 100$. It can be noted that in this setup the EBDM achieves a fit on par with the Direct method and significantly better than the two-stage method. Figure 7.3 shows the mean and standard deviation of the parameter estimates of G_{31}. It is evident that the EBDM gives a smaller bias and a greatly reduced variance compared to the other considered identification methods. The reduction in variance is attributed to the regularization approach used in this method. The fit is calculated using the estimated impulse response from the estimated parameters of the target module. Even though, the variability is high in estimated parameters using the Direct Method, it did not affect the fit of the impulse response, that produces an on par result in figure 7.2 when compared with EBDM. However, Figure 7.3 clearly shows that EBDM performs better than the other considered approaches. Considering a relatively small sized network with 3 modules in the MISO structure, the developed method proves effective. When the size of the network grows, the results of the direct method may deteriorate further due to increase in variance; furthermore, it is expected that in large networks the model order selection step contributes to inaccurate results. Thus the EBDM, by offering reduced variance and circumventing the problem of model order selection, can stand out as an effective local module identification method in large dynamic networks.

7.6.2 Case study 2

Now we look into the case where the modules in the MISO structure may not be stable. In this case, we consider the same network as in Figure 7.1, however with unstable module G^0_{31} (target module) and G^0_{32}. The network modules of network

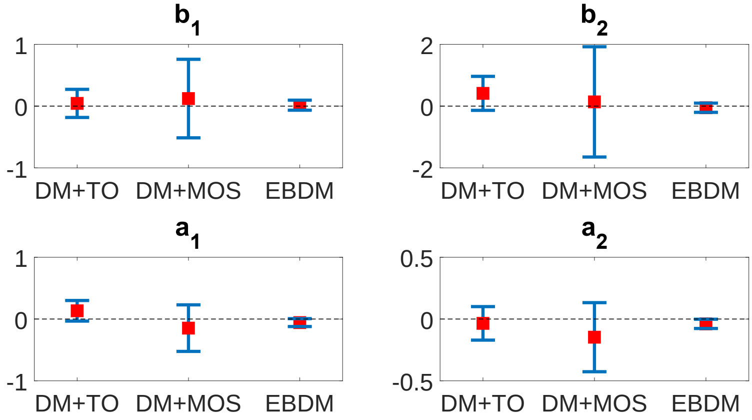


Figure 7.3: Bias and standard deviation of each parameter obtained from 50 MC simulations using different identification methods.

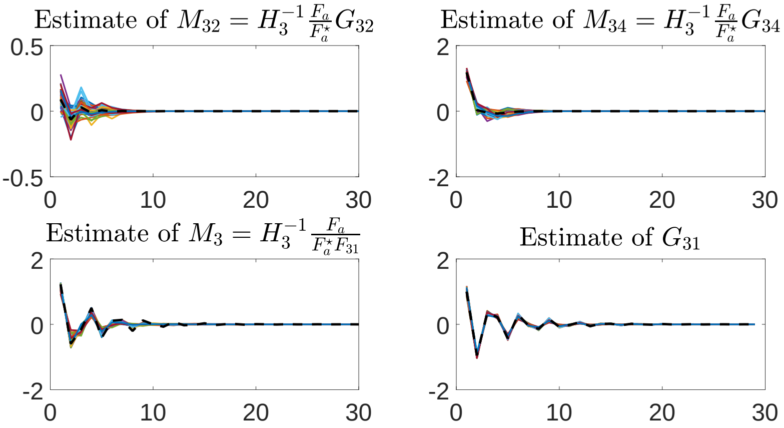


Figure 7.4: Bottom right plot provides the impulse response estimate of the target module at the end of each MC simulation, which is obtained from the estimated parameter θ. The other plots show the impulse response estimates of the filters that are modeled as GP's, which is obtained by calculating the posterior (7.19) from the estimated hyperparameters. The black dashed line provides the true impulse response of the modules.

in Figure 7.1 are the same as in previous section but with unstable G_{31}^0 and G_{32}^0 given by

$$G_{31}^0 = \frac{q^{-1} + 0.05q^{-2}}{1 + 1.7q^{-1} + 1.073q^{-2}} = \frac{b_1^0 q^{-1} + b_2^0 q^{-2}}{1 + a_1^0 q^{-1} + a_2^0 q^{-2}}$$
$$G_{32}^0 = \frac{-0.7339q^{-1} - 0.1256q^{-2} + 0.04023q^{-3} + 0.011q^{-4}}{1 - 1.089q^{-1} - 0.104q^{-2} + 0.052q^{-3} + 0.011q^{-4}}.$$

G_{31}^0 has two complex poles that are not stable and G_{32}^0 has four poles of which one is a real unstable pole. The noise source $e_3(t)$ has variance of 0.1. The experiment setup is similar to the previous case and we run 50 MC experiments with the introduced method in this chapter.

To evaluate the performance of the EBDM, we use the standard goodness-of-fit metric,

$$\text{Fit} = 1 - \frac{\left\|\theta^0 - \hat{\theta}\right\|_2}{\left\|\theta^0 - \bar{\theta}\right\|_2},$$

where θ^0 are the true parameters of the target module, $\hat{\theta}$ are the estimated parameters and $\bar{\theta}$ is the sample mean of θ^0. Due to the instability of the target module, we choose fit on parameters and not on the impulse response. The box plot of the fit of the parameters of $G_{31}(q)$ is shown in Figure 7.5, where the Empirical Bayes Direct Method ('EBDM') is used to identify the unstable target module. We choose $l = 200$. It can be noted that the box plot is above 0.9, which indicates a better fit. Figure 7.6 shows the mean and standard deviation of the parameter estimates of G_{31}. It is evident that the bias and variance is small. The reduction in variance is attributed to the regularization approach used in this method.

It is noteworthy to compare the introduced EBDM with other available approaches that can identify unstable modules. In [47], a method to identify unstable SISO systems with Box-Jenkins (BJ) structure using high order ARX modeling has been introduced. This method proves effective in estimating the unstable poles of the system with high accuracy (less variance) [47], but the estimated model will have high variance due to high order modeling. Also, the estimated model will be of high order unless there is sufficiently large data. Figure 7.7 shows the bode magnitude plot of the estimates after 50 MC simulations with the experimental setup in case study 2 using EBDM and the method of ARX modeling in [47]. ARX models of 15^{th} order are used for the latter method. Even though the estimate of unstable poles are with high accuracy for the latter method, the EBDM performs significantly better in terms of accuracy with less variance in the identified frequency response. Since we have limited data ($N = 500$), the estimated model with the method in [47] is of high order, which can be verified from figure 7.7.

A three step parametric identification method to identify unstable SISO system is introduced in [50]. The first step involves identifying the unstable poles of the parameterized model using the result that the unstable poles can be identified with high accuracy using the method in [47]. In the next step, from the obtained

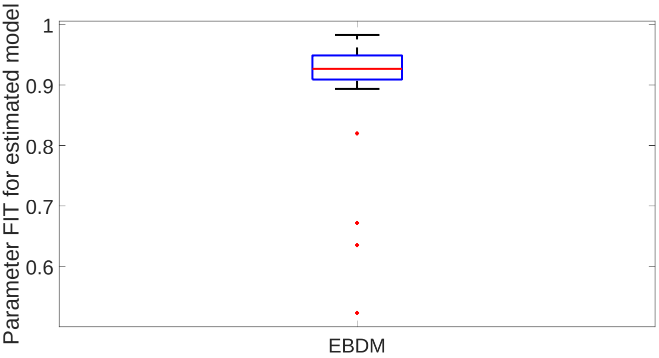


Figure 7.5: Box plot of the fit of the parameters of $\hat{G}_{31}$ obtained by the proposed method. Number of data samples used for estimation is $N = 500$.

estimates, the parameters of the anti-stable part is fixed, and a weighted null space fitting (WNSF) method is used to identify the rest of the parameters of the parameterized model of interest. However, for the MISO identification setup in a dynamic network framework, we might end up in estimating 'false' unstable poles for the target module in the first step where ARX modeling is used. Due to high order ARX modeling, these 'false' unstable poles can be the unstable poles of the modules in the MISO setup other than the target module and it becomes difficult to distinguish the unstable poles between each modules, so that the estimate of unstable roots of the target module can be fixed for the second step. For example, the simulations depicted in Figure 7.7 using the ARX modeling method, we estimate the target module of order 15 with 3 unstable poles, where 2 unstable poles are the poles of G_{31}^{0} and the extra unstable pole is the unstable pole of G_{32}^{0}. Therefore, it becomes difficult to use the WNSF method in this setup without prior knowledge about the unstable poles. An alternative BJ model has been proposed in [45] that can be used with prediction error framework. However, implementation of this is significantly more complex than the introduced EBDM.

7.6.3 Estimated noise variance

Using the experimental setup of case study 1 and 2 but with different noise power (variance) of e_3 (σ_3) acting on the output node w_3, we performed simulations using the EBDM for the network in Figure 7.1. For the case study 1, since all modules are stable (i.e. $\frac{F_a}{F_a^\star} = 1$), the estimated noise variance $\hat{\bar{\sigma}}_3$ should be approximately equal to the actual noise variance σ_3 (see remark 7.2). This can be verified from the Table 7.1 (upper) where the estimated noise variance approximates well the actual noise variance in the network. Considering the case study 2, the estimated noise variance $\hat{\bar{\sigma}}_3$ should be approximately equal to the scaled version of the actual noise variance σ_3 given by $\bar{\sigma}_3^2 = |\frac{F_a}{F_a^\star}|^2\sigma_3^2 = |f_{a_{n_f}}|^2\sigma_3^2$ i.e. the “dummy"noise variance. This can be verified from the Table 7.1 (lower).

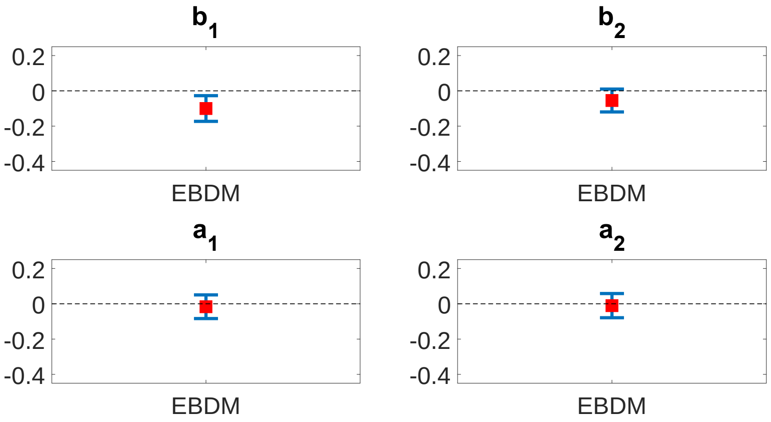


Figure 7.6: Bias and standard deviation of each parameter obtained from 50 MC simulations using different identification methods.

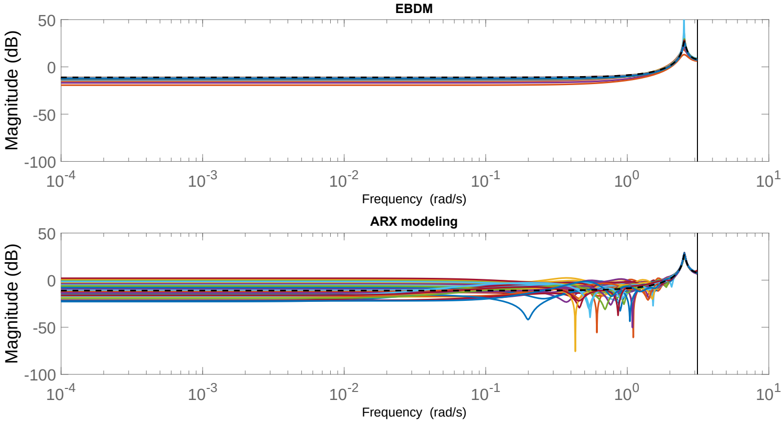


Figure 7.7: Bode magnitude plot to compare the estimates of the introduced approach(upper) and the approach in [47](lower).

Actual value ($\bar{\sigma}_3 = \sigma_3$)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	1	2
Estimated value ($\hat{\bar{\sigma}}_3$)	0.0971	0.1908	0.2804	0.4093	0.4710	0.6314	0.7620	0.8207	0.9449	1.9398

Actual value ($\bar{\sigma}_3^2 = \lvert\frac{F_a}{F_a^\star}\rvert^2 \sigma_3^2$)	0.1475	0.2950	0.4425	0.5901	0.7376	0.8851	1.0326	1.1801	1.4752	2.9503
Estimated value ($\hat{\bar{\sigma}}_3$)	0.1520	0.3005	0.4579	0.5946	0.7338	0.8642	0.9145	1.1851	1.6030	2.7349

Table 7.1: Results of the simulations that were performed using the setup of case study 1 (upper) and 2 (lower) with different noise variance of e_3 acting on the output node w_3. Table 1 shows the actual "dummy" noise variance to be estimated and the estimated noise variance using EBDM for the experimental setup in Case 1 (upper) and Case 2 (lower).

7.6.4 Additional remarks

The method described in this chapter can be developed using any of the kernels available in the literature of regularized system identification. The choice of kernel adopted in this chapter is the result of a balance between its empirical effectiveness (see [93]) and its computational efficiency (due to its factorization and the low number of hyperparameter). Other choices of kernel (e.g. the DC kernel proposed in [21]) may result in a final higher accuracy, requiring to estimate an additional hyperparameter, which might bring an additional cost in complexity. On the other hand, it is well known (see [21]) that the optimal kernel is constructed from the true impulse response, which is unknown (it is the actual object of interest). The question which is the best choice of kernel for dynamic networks is open and requires a thorough theoretical analysis which is outside the scope of the work in this chapter.

7.7 Conclusions

An effective regularized kernel-based approach for local module identification in dynamic networks has been introduced in this chapter. The introduced method (EBDM) circumvents the model order selection step for all the modules that are not of primary interest to the experimenter, but still need to be identified in order to get a consistent estimate of the target module. Furthermore, by using regularized non-parametric methods, the number of parameters to be estimated is greatly reduced, with a clear benefit in terms of mean square error of the estimated target module. Therefore, the method is computationally less complex and scales favorably to large size networks. The method developed in this chapter is capable of performing identification in networks composed by unstable modules, without any prior information about the stability of the modules. Numerical experiments performed with a dynamic network example illustrate the potentials of the developed method on comparison with the already available methods on networks of stable modules. The developed method provides better estimates and a reduced variance is observed in the identified model due to the integration of the regularization approach in the method.

Appendices

7.A Proof of Proposition 7.1

Analogous to the factorization technique used in [45] and [47], we factorize each F_{jk} (from now on superscript ^{0} is dropped for convenience) as,

$$F_{jk}(q) = F_{jk}^{(s)}(q)F_{jk}^{(a)}(q) \tag{7.31}$$

where $F_{jk}^{(s)}(q)$ contains the stable roots of $F_{jk}(z)$ and $F_{jk}^{(a)}(q)$ contains the anti-stable roots of $F_{jk}(z)$, which are given by

$$\begin{aligned} F_{jk}^{(s)}(q) &= 1 + f_{jk}{}_1^{(s)}q^{-1} + \cdots + f_{jk}{}_{n_f}^{(s)}q^{-n_{f_{jk}}^{(s)}} && (7.32) \\ F_{jk}^{(a)}(q) &= 1 + f_{jk}{}_1^{(a)}q^{-1} + \cdots + f_{jk}{}_{n_f}^{(a)}q^{-n_{f_{jk}}^{(a)}}. && (7.33) \end{aligned}$$

We introduce $F_{jk}^{*(a)}(q)$ as the monic polynomial whose roots are the mirrored (and stable) roots of $F_{jk}^{(a)}(q)$. We can write $F_{jk}^{*(a)}(q)$ as,

$$F_{jk}^{*(a)}(q) = 1 + \frac{f_{jk}{}_{n_f-1}^{(a)}}{f_{jk}{}_{n_f}^{(a)}}q^{-1} + \cdots + \frac{1}{f_{jk}{}_{n_f}^{(a)}}q^{-n_{f_{jk}}}, \tag{7.34}$$

assuming without loss of generality that $f_{jk}{}_{n_f}^{(a)} \neq 0$. Then, we define $F_a(q)$ as the product of all polynomials with anti-stable roots i.e. $F_a(q) = \prod_{k\in\mathcal{N}_j} F_{jk}^{(a)}(q) = 1 + f_{a1}q^{-1} + \cdots + f_{a n_f}q^{-n_{f_a}}$, and $F_a^*(q)$ as the polynomial with mirrored roots of $F_a(q)$ inside the unit circle i.e. $F_a^*(q) = \prod_{k\in\mathcal{N}_j} F_{jk}^{*(a)}(q) = 1 + \frac{f_{a n_f-1}}{f_{a n_f}}q^{-1} + \cdots + \frac{1}{f_{a n_f}}q^{-n_{f_a}}$.

As the next step, we re-write the noise term $v_j(t)$ in (7.2) using a input white noise process $\bar{e}_j(t)$ instead of $e_j(t)$. Using the fact that $\frac{F_a^\star(q)}{F_a(q)}$ is an all pass filter (linear) with a magnitude of $|\frac{1}{f_{a n_f}}|$[45], we can write $v_j(t) = H_j(q)\frac{F_a^\star(q)}{F_a(q)}\bar{e}_j(t)$ whose noise spectrum Φ_{v_j} equals $|H(e^{i\omega})|^2|\frac{1}{f_{a n_f}}|^2\bar{\sigma}_j^2$, where $\bar{\sigma}_j^2 = |f_{a n_f}|^2\sigma_j^2$ is the variance of $\bar{e}_j(t)$.

With the above expression of the noise term and using $G_{ji}(q) = \frac{B_{ji}(q)}{F_{ji}(q)} = \frac{B_{ji}(q)}{F_{ji}^{(s)}(q)F_{ji}^{(a)}(q)}$, and assuming $r_j(t) = 0$ for the sake of brevity, Eq. (7.2) is rewritten as,

$$\begin{aligned} w_j(t) &= M_j(q)w_j(t) - (1 - M_j(q))\bar{F}_{ji}(q)w_j(t) \\ &+ (1 - M_j(q))B_{ji}(q)w_i(t) + \sum_{k \in \mathcal{N}_j \setminus \{i\}} M_{jk}(q)w_k(t) + \bar{e}_j(t) \end{aligned} \tag{7.35}$$

with

$$M_j(q) \quad := \quad 1 - \left((H_j)^{-1}(q) \frac{\prod_{k \in \mathcal{N}_j \setminus \{i\}} F_{jk}^{(a)}(q)}{F_a^\star(q) F_{ji}^{(s)}(q)} \right) \quad , \tag{7.36}$$

$$M_{jk}(q) \quad := \quad (H_j)^{-1} \frac{\prod_{\ell \in \mathcal{N}_j \setminus \{k\}} F_{j\ell}^{(a)}(q)}{F_a^\star(q)} \frac{B_{jk}(q)}{F_{jk}^{(s)}(q)} \, , \tag{7.37}$$

where $\bar{F}_{ji}(q) = -(1 - F_{ji}(q))$, and $M_j(q)$ is a strictly proper stable filters with only stable poles which are the roots of $F_a^*(z)$, $F_{ji}^{(s)}(z)$ and poles of $(H_j)^{-1}$, while $M_{jk}(q), k \in \mathcal{N}_j \setminus \{i\}$ are also strictly proper stable filters with only stable poles which are the roots of $F_a^*(z)$, $F_{jk}^{(s)}(z)$ and poles of $(H_j)^{-1}$.

7.A.1 Proof of Lemma 7.2

Using the Bayes' rule the expression in Eq. (10.28) can be written as,

$$\begin{aligned} Q^{(n)}(\eta) &= \mathbb{E}[\log p(w_j | m_j, m_{k1}, m_{k2}, \dots, m_{kp}; \eta)] \\ &+ \mathbb{E}[\log p(m_j; \eta) + \log p(m_{k1}; \eta) + \dots + \log p(m_{kp}; \eta)] \end{aligned} \tag{7.38}$$

$$Q^{(n)}(\eta) = \mathbb{E}[\mathcal{A}] + \mathbb{E}[\mathcal{B}] \tag{7.39}$$

$$\mathcal{A} := -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\bar{\sigma}_j^2) - \frac{1}{2\bar{\sigma}_j^2}(w_j - \mathbf{W}_{ji}\mathbf{g}_{ji} - \mathbf{W}\mathbf{m})^\top (w_j - \mathbf{W}_{ji}g_{ji} - \mathbf{W}\mathbf{m}) \tag{7.40}$$

$$\begin{aligned} \mathcal{B} :=& -\frac{l}{2}\log(2\pi) - \frac{1}{2}\log[\det(\lambda_j K_{\beta_j})] - \frac{1}{2} {m_j}^\top (\lambda_j K_{\beta_j})^{-1} m_j \\ &+ \sum_{k \in \mathcal{N}_j \setminus \{i\}} \left[-\frac{l}{2}\log(2\pi) - \frac{1}{2}\log[\det(\lambda_k K_{\beta_k})] - \frac{1}{2} {m_k}^\top (\lambda_k K_{\beta_k})^{-1} m_k \right] \end{aligned} \tag{7.41}$$

Taking Expectation of each element in $\mathcal{A}$ and $\mathcal{B}$ with respect to $p(\mathbf{m}|w_j;\hat{\eta}^{(n)})$ (i.e. $\mathbb{E}_{p(\mathbf{m}|w_j;\hat{\eta}^{(n)})}$) we get,

$$\begin{aligned}\mathbb{E}[\mathcal{A}] = &-\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\bar{\sigma}_j^2) - \frac{1}{2\bar{\sigma}_j^2}\Big[w_j^\top w_j - \mathbf{g}_{ji}^\top \mathbf{W}_{ji}^\top w_j \\ &- \mathbb{E}[\mathbf{m}^\top]\mathbf{W}^\top w_j - w_j^\top \mathbf{W}_{ji}\mathbf{g}_{ji} + \mathbf{g}_{ji}^\top \mathbf{W}_{ji}^\top \mathbf{W}_{ji}\mathbf{g}_{ji} + \\ &\mathbb{E}[\mathbf{m}^\top]\mathbf{W}^\top \mathbf{W}_{ji}\mathbf{g}_{ji} - w_j^\top \mathbf{W}\mathbb{E}[\mathbf{m}] + \mathbf{g}_{ji}^\top \mathbf{W}_{ji}^\top \mathbf{W}\mathbb{E}[\mathbf{m}] + \mathrm{tr}(\mathbf{W}^\top \mathbf{W}\mathbb{E}[\mathbf{m}\mathbf{m}^\top])\Big]\end{aligned} \tag{7.42}$$

$$\begin{aligned}\mathbb{E}[\mathcal{B}] = &-\frac{l}{2}\log(2\pi) - \frac{1}{2}\log[\det(\lambda_j K_{\beta_j})] - \frac{1}{2}\mathrm{tr}\big((\lambda_j K_{\beta_j})^{-1}\mathbb{E}[m_j {m_j}^\top]\big) \\ &+ \sum_{k\in\mathcal{N}_j\setminus\{i\}}\Big[-\frac{l}{2}\log(2\pi) - \frac{1}{2}\log[\det(\lambda_k K_{\beta_k})] - \frac{1}{2}\mathrm{tr}\big((\lambda_k K_{\beta_k})^{-1}\mathbb{E}[m_k {m_k}^\top]\big)\Big]\end{aligned} \tag{7.43}$$

The constants can be removed from the objective functions and multiplication with scalar value 2 can be done to simplify the objective function. On substituting the expected values $\mathbb{E}[\mathbf{m}\mathbf{m}^\top] = \hat{\mathbf{M}}^{(n)}$, $\mathbb{E}[m_k {m_k}^\top] = \hat{\mathbf{M}}_k^{(n)}$, $\mathbb{E}[m_j {m_j}^\top] = \hat{\mathbf{M}}_j^{(n)}$ and $\mathbb{E}[\mathbf{m}] = \hat{\mathbf{m}}^{(n)}$ we get the statement of the Lemma.

7.B Proof of Theorem 7.1

The proof follows the procedure used in [14]. We partially differentiate (10.67) with respect to λ_k and equate to zero to get the λ_k^* expression. Substituting this λ_k^* in (10.67) we get the expression for (10.44) using which we obtain $\hat{\beta}_k^{(n+1)}$. Equation (10.46) is the expression of λ_k^* after substituting $\hat{\beta}_k^{(n+1)}$.

7.C Proof of Theorem 7.2

In order to find $\hat{\theta}^{(n)}$, $\bar{\sigma}_j^2$ is fixed to $\hat{\bar{\sigma}}_j^{2(n)}$ and substituted in Eq. (10.69). After substitution the terms that are independent of θ can be removed from the objective function since it becomes a constant. Then we get,

$$\begin{aligned}Q_o^{(n)}(\theta, \hat{\bar{\sigma}}_j^{2(n)}) = \text{constant} - \frac{1}{\hat{\bar{\sigma}}_j^{2(n)}}\Big[&-2w_j^\top \mathbf{W}\hat{\mathbf{m}}^{(n)} + \mathrm{tr}\big(\mathbf{W}^\top \mathbf{W}\hat{\mathbf{M}}^{(n)}\big) \\ &-2w_j^\top \mathbf{W}_{ji}\mathbf{g}_{ji} + \mathbf{g}_{ji}^\top \mathbf{W}_{ji}^\top \mathbf{W}_{ji}\mathbf{g}_{ji} + 2\mathbf{g}_{ji}^\top \mathbf{W}_{ji}^\top \mathbf{W}\hat{\mathbf{m}}^{(n)}\Big].\end{aligned} \tag{7.44}$$

We now introduce the following notation. Let $D_1 \in \mathbb{R}^{N^2\times N}$ and $D_2 \in \mathbb{R}^{N^2\times N}$ are two matrices such that, for any vector $\mathbf{w} \in \mathbb{R}^N$, $D_1\mathbf{w} = \mathrm{vec}(W)$, where W is the Toeplitz matrix of $\mathbf{w}$, and $D_2\mathbf{w} = \mathrm{vec}(W^\top)$. Let us define $\breve{m}^{(n)} \in \mathbb{R}^N$ be a vector such that, if $N \leqslant l$, $\breve{m}^{(n)}$ is the vector of first N elements of $\hat{m}^{(n)}$ and if $N > l$, $\breve{m}^{(n)}$ is a vector with the first l elements equal to $\hat{m}^{(n)}$ and the remaining ones equal to 0. Let $\breve{M}^{(n)}$, $\overleftrightarrow{W}_{\ell}^{N} \in \mathbb{R}^{N \times N}$ where $\ell \in \{i, j\}$ be the Toeplitz matrix of $\breve{m}^{(n)}$ and $\overleftrightarrow{w}_{\ell}$ respectively. Then

$$\mathbf{X} = \begin{bmatrix} W_j & W_{k1} & \dots & W_{kp} \end{bmatrix}, \quad \hat{\mathbf{Y}}^{(n)} = \breve{M}^{(n)}[\overleftrightarrow{W}_{i}^{N} \quad -\overleftrightarrow{W}_{j}^{N}]$$

and

$$\mathbf{Z}_i = \begin{bmatrix} \overleftrightarrow{W}_i & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{N \times (p+1)l} ,$$

$$\mathbf{Z}_j = \begin{bmatrix} -\overleftrightarrow{W}_j & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{N \times (p+1)l} .$$

We can re-write the following terms, $\mathbf{W}\hat{\mathbf{m}}^{(n)} = \mathbf{X}\hat{\mathbf{m}}^{(n)} + G_b \overleftrightarrow{W}_i \hat{\mathbf{m}}_j^{(n)} - G_f \overleftrightarrow{W}_j \hat{\mathbf{m}}_j^{(n)} = \mathbf{X}\hat{\mathbf{m}}^{(n)} + \hat{\mathbf{Y}}^{(n)}\mathbf{g}_{ji}$ and $\mathbf{W} = \mathbf{X} + G_b\mathbf{Z}_i + G_f\mathbf{Z}_j$. Therefore,

$$\begin{aligned}
\hat{\theta}^{(n+1)} &= \arg\max_{\theta} \Big[2w_j^{\top}\mathbf{W}\hat{\mathbf{m}}^{(n)} - \operatorname{tr}\big(\mathbf{W}^{\top}\mathbf{W}\hat{\mathbf{M}}^{(n)}\big) + 2w_j^{\top}\mathbf{W}_{ji}\mathbf{g}_{ji} \\
&\quad - \mathbf{g}_{ji}^{\top}\mathbf{W}_{ji}^{\top}\mathbf{W}_{ji}\mathbf{g}_{ji} - 2\mathbf{g}_{ji}^{\top}\mathbf{W}_{ji}^{\top}\mathbf{W}\hat{\mathbf{m}}^{(n)} \Big] \\
&= \arg\max_{\theta} \Big[2w_j^{\top}\mathbf{X}\hat{\mathbf{m}}^{(n)} + 2w_j^{\top}\hat{\mathbf{Y}}^{(n)}\mathbf{g}_{ji} - \operatorname{tr}\big(\mathbf{X}\mathbf{X}^{\top}\hat{\mathbf{M}}^{(n)}\big) - \operatorname{tr}\big(\mathbf{X}\hat{\mathbf{M}}^{(n)}\mathbf{Z}_j^{\top}G_f^{\top}\big) \\
&\quad - \operatorname{tr}\big(\mathbf{Z}_i\hat{\mathbf{M}}^{(n)}\mathbf{X}^{\top}G_b\big) - \operatorname{tr}\big(\mathbf{Z}_j\hat{\mathbf{M}}^{(n)}\mathbf{X}^{\top}G_f\big) - \operatorname{tr}\big(G_b\mathbf{Z}_i\hat{\mathbf{M}}^{(n)}\mathbf{Z}_i^{\top}G_b^{\top}\big) \\
&\quad - \operatorname{tr}\big(G_f\mathbf{Z}_j\hat{\mathbf{M}}^{(n)}\mathbf{Z}_j^{\top}G_f^{\top}\big) - \operatorname{tr}\big(G_b\mathbf{Z}_i\hat{\mathbf{M}}^{(n)}\mathbf{Z}_j^{\top}G_f^{\top}\big) - \operatorname{tr}\big(G_f\mathbf{Z}_j\hat{\mathbf{M}}^{(n)}\mathbf{Z}_i^{\top}G_b^{\top}\big) \\
&\quad - \operatorname{tr}\big(\mathbf{X}\hat{\mathbf{M}}^{(n)}\mathbf{Z}_i^{\top}G_b^{\top}\big) + 2w_j^{\top}\mathbf{W}_{ji}\mathbf{g}_{ji} - \mathbf{g}_{ji}^{\top}\mathbf{W}_{ji}^{\top}\mathbf{W}_{ji}\mathbf{g}_{ji} \\
&\quad - 2\mathbf{g}_{ji}^{\top}\mathbf{W}_{ji}^{\top}\mathbf{X}\hat{\mathbf{m}}^{(n)} - 2\mathbf{g}_{ji}^{\top}\mathbf{W}_{ji}^{\top}\hat{\mathbf{Y}}^{(n)}\mathbf{g}_{ji} \Big]
\end{aligned}$$

Neglecting constant terms we get,

$$\begin{aligned}
&\hat{\theta}^{(n+1)} \\
&= \arg\max_{\theta} \Big[2w_j^{\top}\hat{\mathbf{Y}}^{(n)}\mathbf{g}_{ji} - \operatorname{tr}\big(\mathbf{X}\hat{\mathbf{M}}^{(n)}\mathbf{Z}_i^{\top}G_b^{\top}\big) - \operatorname{tr}\big(\mathbf{X}\hat{\mathbf{M}}^{(n)}\mathbf{Z}_j^{\top}G_f^{\top}\big) - \operatorname{tr}\big(\mathbf{Z}_i\hat{\mathbf{M}}^{(n)}\mathbf{X}^{\top}G_b\big) \\
&- \operatorname{tr}\big(\mathbf{Z}_j\hat{\mathbf{M}}^{(n)}\mathbf{X}^{\top}G_f\big) - \operatorname{tr}\big(G_b\mathbf{Z}_i\hat{\mathbf{M}}^{(n)}\mathbf{Z}_i^{\top}G_b^{\top}\big) - \operatorname{tr}\big(G_f\mathbf{Z}_j\hat{\mathbf{M}}^{(n)}\mathbf{Z}_j^{\top}G_f^{\top}\big) \\
&- \operatorname{tr}\big(G_b\mathbf{Z}_i\hat{\mathbf{M}}^{(n)}\mathbf{Z}_j^{\top}G_f^{\top}\big) - \operatorname{tr}\big(G_f\mathbf{Z}_j\hat{\mathbf{M}}^{(n)}\mathbf{Z}_i^{\top}G_b^{\top}\big) + 2w_j^{\top}\mathbf{W}_{ji}\mathbf{g}_{ji} \\
&- \mathbf{g}_{ji}^{\top}\mathbf{W}_{ji}^{\top}\mathbf{W}_{ji}\mathbf{g}_{ji} - 2\hat{\mathbf{m}}^{(n)\top}\mathbf{X}^{\top}\mathbf{W}_{ji}\mathbf{g}_{ji} - 2\mathbf{g}_{ji}^{\top}\mathbf{W}_{ji}^{\top}\hat{\mathbf{Y}}^{(n)}\mathbf{g}_{ji} \Big]
\end{aligned}$$

$$
\begin{aligned}
&= \arg\max_{\theta} \Bigg[2w_j^\top \hat{\mathbf{Y}}^{(n)} \mathbf{g}_{ji} - \text{vec}(\mathbf{Z}_i \hat{\mathbf{M}}^{(n)\top} \mathbf{X}^\top)^\top D_2 b_{ji} - \text{vec}(\mathbf{Z}_j \hat{\mathbf{M}}^{(n)\top} \mathbf{X}^\top)^\top D_2 f_{ji} \\
&- \text{vec}(\mathbf{X} \hat{\mathbf{M}}^{(n)\top} \mathbf{Z}_i^\top)^\top D_1 b_{ji} - \text{vec}(\mathbf{X} \hat{\mathbf{M}}^{(n)\top} \mathbf{Z}_j^\top)^\top D_1 f_{ji} + 2w_j^\top \mathbf{W}_{ji} \mathbf{g}_{ji} \\
&- b_{ji}^\top D_1^\top (\mathbf{Z}_i \hat{\mathbf{M}}^{(n)} \mathbf{Z}_i^\top \otimes I_N) D_1 b_{ji} - f_{ji}^\top D_1^\top (\mathbf{Z}_j \hat{\mathbf{M}}^{(n)} \mathbf{Z}_j^\top \otimes I_N) D_1 f_{ji} \\
&- b_{ji}^\top D_1^\top (\mathbf{Z}_i \hat{\mathbf{M}}^{(n)} \mathbf{Z}_j^\top \otimes I_N) D_1 f_{ji} - f_{ji}^\top D_1^\top (\mathbf{Z}_j \hat{\mathbf{M}}^{(n)} \mathbf{Z}_i^\top \otimes I_N) D_1 b_{ji} \\
&- \mathbf{g}_{ji}^\top \mathbf{W}_{ji}^\top \mathbf{W}_{ji} \mathbf{g}_{ji} - 2\hat{\mathbf{m}}^{(n)\top} \mathbf{X}^\top \mathbf{W}_{ji} \mathbf{g}_{ji} - 2\mathbf{g}_{ji}^\top \mathbf{W}_{ji}^\top \hat{\mathbf{Y}}^{(n)} \mathbf{g}_{ji} \Bigg].
\end{aligned}
$$

Defining

$$
\begin{aligned}
\hat{A}_{11}^{(n)} &= [D_1^\top (\mathbf{Z}_i \hat{\mathbf{M}}^{(n)} \mathbf{Z}_i^\top \otimes I_N) D_1] \\
\hat{A}_{12}^{(n)} &= [D_1^\top (\mathbf{Z}_i \hat{\mathbf{M}}^{(n)} \mathbf{Z}_j^\top \otimes I_N) D_1] \\
\hat{A}_{21}^{(n)} &= [D_1^\top (\mathbf{Z}_j \hat{\mathbf{M}}^{(n)} \mathbf{Z}_i^\top \otimes I_N) D_1] \\
\hat{A}_{22}^{(n)} &= [D_1^\top (\mathbf{Z}_j \hat{\mathbf{M}}^{(n)} \mathbf{Z}_j^\top \otimes I_N) D_1]
\end{aligned}
$$

$$
\begin{aligned}
\hat{b}_{11}^{(n)} &= \Big[-\frac{1}{2}\text{vec}(\mathbf{Z}_i \hat{\mathbf{M}}^{(n)\top} \mathbf{X}^\top)^\top D_2 - \frac{1}{2}\text{vec}(\mathbf{X} \hat{\mathbf{M}}^{(n)\top} \mathbf{Z}_i^\top)^\top D_1 \Big]^\top, \\
\hat{b}_{12}^{(n)} &= \Big[-\frac{1}{2}\text{vec}(\mathbf{Z}_j \hat{\mathbf{M}}^{(n)\top} \mathbf{X}^\top)^\top D_2 - \frac{1}{2}\text{vec}(\mathbf{X} \hat{\mathbf{M}}^{(n)\top} \mathbf{Z}_j^\top)^\top D_1 \Big]^\top
\end{aligned}
$$

and

$$
\begin{aligned}
\hat{\mathbf{A}}^{(n)} &= \begin{bmatrix} \hat{A}_{11}^{(n)} & \hat{A}_{12}^{(n)} \\ \hat{A}_{21}^{(n)} & \hat{A}_{22}^{(n)} \end{bmatrix} + \mathbf{W}_{ji}^\top \mathbf{W}_{ji} + 2\mathbf{W}_{ji}^\top \hat{\mathbf{Y}}^{(n)}\ , \\
\hat{\mathbf{b}}^{(n)} &= \begin{bmatrix} \hat{b}_{11}^{(n)} \\ \hat{b}_{12}^{(n)} \end{bmatrix} + [w_j^\top \mathbf{W}_{ji} + w_j^\top \hat{\mathbf{Y}}^{(n)} - \hat{\mathbf{m}}^{(n)\top} \mathbf{X}^\top \mathbf{W}_{ji}]^\top
\end{aligned}
$$

we get that the parameter vector θ are updated by solving the problem

$$
\hat{\theta}^{(n+1)} = \arg\min_{\theta} \Big[\mathbf{g}_{ji}^\top \hat{\mathbf{A}}^{(n)} \mathbf{g}_{ji} - 2\hat{\mathbf{b}}^{(n)\top} \mathbf{g}_{ji} \Big]. \tag{7.45}
$$

We have $\mathbf{g}_{ji}$ to be linearly parameterized with θ, that is $\mathbf{g}_{ji} = M\theta$ where $M \in \mathbb{R}^{2N \times n_\theta}$. Therefore, the above problem becomes quadratic and a closed-form solution is achieved. Thus we get the statement of Theorem 7.2.

7.D Proof of Theorem 7.3

In order to find $\hat{\bar{\sigma}}_j^{2(n)}$, θ is fixed to $\hat{\theta}^{(n+1)}$ and substituted in Eq. (10.69). After substitution, $Q_o^{(n)}(\bar{\sigma}_j^2, \hat{\theta}^{(n+1)})$ is differentiated w.r.t. $\bar{\sigma}_j^2$ and equated to zero to get the statement of the Theorem.

8 CHAPTER

Empirical Bayes Local Direct Method

This chapter considers the same problem in the previous chapter, i.e., identifying one module embedded in a dynamic network. However, now the noise sources can be correlated. To achieve this using the direct method for single module identification, we need to formulate a Multi-Input-Multi-Output (MIMO) estimation problem, as described in Chapter 3 of this thesis. Solving the MIMO estimation problem requires a model order selection step for each module in the setup and estimation of a large number of parameters. This results in a larger variance in the estimates and increased computation complexity. Therefore, we extend the Empirical Bayes Direct Method in Chapter 7, which handles the above mentioned problems for a Multi-Input-Single-Output (MISO) setup to a MIMO setting by suitably modifying the framework. We keep a parametric model for the desired target module and model the impulse response of all the other modules as independent zero-mean Gaussian process governed by a first-order stable spline kernel. The parameters of the target module are obtained by maximizing the marginal likelihood of the output using the Empirical Bayes (EB) approach. To solve this, we use the Expectation-Maximization (EM) algorithm, which offers computational advantages. Numerical simulations illustrate the advantages of the developed method over existing classical methods.

8.1 Introduction

The situation of correlation in process noise can be handled using PEM's like the *indirect method* [54] and its variants like the *two stage method* [29, 124] and

This chapter is based on the publication: V. C. Rajagopal, K. R. Ramaswamy, and P. M. J. Van den Hof, "A regularized kernel-based method for learning a module in a dynamic network with correlated noise," in *Proc. 59th IEEE Conf. on Decision and Control (CDC)*, Jeju Island, Korea, 2020, pp. 4348–4353.

instrumental variable methods [27, 120]. However, these methods require a strong presence of measured external excitation signals to serve as predictor inputs, and might increase the cost of experiments. On the contrary, the direct approaches use the entire information of the node signal (both excitation and noise signal), but suffer from handling correlated noise when using a MISO setup. A solution to this problem has been provided in [104] (Chapter 3) as the *local direct method*. In this method, we handle the effect of noise correlation in dynamic networks by moving from a MISO to Multi-Input-Multi-Output (MIMO) identification setup, where the single module identification problem becomes embedded in a network MIMO identification problem, resulting in the problem of estimating high number of parameters (even more than the MISO setup in previous chapter) that are of no prime interest to the experimenter. In addition, all these additional modules need to be suitably parameterized based on complexity criteria like AIC, BIC, or Cross Validation (CV) [77]. This step involves permuting candidate model orders for all modules which increases exponentially with the number of modules or their orders.

To eliminate the model order selection step and reduce the number of estimated parameters, we build on the work of the previous chapter and develop a regularized kernel based method (see [93] for a survey) that extends the semi-parametric approach of the previous chapter from a MISO setting to a MIMO setting. Preserving the approach of the previous chapter, we maintain a parametric model for the target module to accurately capture the dynamics, while using independent Gaussian processes to model the impulse responses of other modules. The covariance matrix of these processes are given by the *first-order stable spline kernel* [21] which enforces stability and smoothness of the impulse response coefficients. The parameters of the target module, hyperparameters of the kernel and the covariance of the process noise are estimated by maximizing the marginal likelihood of the data, achieved by an Expectation-Maximization (EM) method having attractive computational properties.

8.2 Problem statement

Assumption 8.1 *In the dynamic network represented by* (2.2)*, we consider the following assumption:*

- *The structure of the disturbance topology is known i.e. we know a priori which entries of $H^0(q)$ are nonzero.*

According to the local direct method in chapter 3, a module G_{ji} embedded in a dynamic network with correlated noise can be consistently identified with a MIMO estimation setup, $w_{\mathcal{D}} \rightarrow w_{\mathcal{Y}}$. Here, predictor inputs $w_{\mathcal{D}}$ and predicted outputs $w_{\mathcal{Y}}$ may have common signals to handle the confounding variables that arise due to correlated disturbances. Therefore, by exploiting a multivariate noise model, the effect of correlated disturbances are covered. The estimation setup results from the network equation

$$\underbrace{\begin{bmatrix} w_{\mathcal{Q}} \\ w_o \end{bmatrix}}_{w_{\mathcal{Y}}} = \underbrace{\begin{bmatrix} \bar{G}_{\mathcal{QQ}} & \bar{G}_{\mathcal{QU}} \\ \bar{G}_{\mathcal{OQ}} & \bar{G}_{\mathcal{OU}} \end{bmatrix}}_{\bar{G}} \underbrace{\begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{U}} \end{bmatrix}}_{w_{\mathcal{D}}} + \underbrace{\begin{bmatrix} \bar{H}_{\mathcal{QQ}} & \bar{H}_{\mathcal{QU}} \\ \bar{H}_{\mathcal{OQ}} & \bar{H}_{\mathcal{OU}} \end{bmatrix}}_{\bar{H}} \underbrace{\begin{bmatrix} \xi_{\mathcal{Q}} \\ \xi_o \end{bmatrix}}_{\xi_{\mathcal{Y}}}, \tag{8.1}$$

where $w_{\mathcal{Q}}$ are the set of nodes that are common to both inputs and outputs, $w_{\mathcal{U}}$ and w_o are the sets of nodes that are exclusively inputs and outputs respectively. The vector $\xi_{\mathcal{Y}}$ is a Gaussian white noise process constructed by spectral decomposition and $\bar{H}$ is square, stable, monic and minimum phase. The desired target module is represented in $\bar{G}_{ji}$ i.e. $\bar{G}_{ji} = G_{ji}$ and $\bar{G}_{\mathcal{QQ}}$ is a hollow matrix and thus does not lead to transfers between signals that are the same. Also, the non-zero entries in $\bar{G}$ can be estimated (refer to Chapter 3 for the local direct method). Without loss of generality, $u = 0$ is considered for simplicity.

We want to identify a parametric model for the module directly linking node w_i and w_j, represented as $G_{ji}(q, \theta)$ that describes the dynamics of the module of interest for a certain parameter vector $\theta \in \mathbb{R}^{n_\theta}$, from N measurements of the node signals $w_{\mathcal{D}}$ and $w_{\mathcal{Y}}$. In the local direct method, not only the target module G_{ji} but all the modules in $\bar{G}$ are parameterized, resulting in high number of parameters to estimate which causes a detrimental effect on the variance of the parameter estimates when N is not very large. Therefore, we focus on estimating a parametric model for the target module while reducing the number of parameters for the remaining modules in the MIMO identification setup.

8.3 Developing the Bayesian model

In this section, we discuss how we avoid parameterizing all but the target module using regularized kernel-based methods. As the starting point of the methodology in this chapter, we use the MIMO structure in (8.1), as opposed to a MISO structure in the *Empirical Bayes Direct Method* (EBDM). Following (8.1), while maintaining the monicity of the noise model, the equation can be re-ordered as

$$\underbrace{\begin{bmatrix} w_j \\ w_{\tilde{\mathcal{Y}}} \end{bmatrix}}_{\tilde{w}_{\mathcal{Y}}(t)} = \underbrace{\begin{bmatrix} G_{ji} & \bar{G}_{j\tilde{\mathcal{D}}} \\ \bar{G}_{\tilde{\mathcal{Y}}i} & \bar{G}_{\tilde{\mathcal{Y}}\tilde{\mathcal{D}}} \end{bmatrix}}_{\check{G}} \underbrace{\begin{bmatrix} w_i \\ w_{\tilde{\mathcal{D}}} \end{bmatrix}}_{\tilde{w}_{\mathcal{D}}(t)} + \underbrace{\begin{bmatrix} \bar{H}_{jj} & \bar{H}_{j\tilde{\mathcal{Y}}} \\ \bar{H}_{\tilde{\mathcal{Y}}j} & \bar{H}_{\tilde{\mathcal{Y}}\tilde{\mathcal{Y}}} \end{bmatrix}}_{\check{H}} \underbrace{\begin{bmatrix} \xi_j \\ \xi_{\tilde{\mathcal{Y}}} \end{bmatrix}}_{\tilde{\xi}_{\mathcal{Y}}(t)}, \tag{8.2}$$

where $\tilde{\mathcal{Y}} = \mathcal{Y} \backslash \{j\}$ and $\tilde{\mathcal{D}} = \mathcal{D} \backslash \{i\}$. The signals $\tilde{w}_{\mathcal{Y}}$, $\tilde{w}_{\mathcal{D}}$, and $\tilde{\xi}_{\mathcal{Y}}$ are suitably rearranged. To parameterize only G_{ji} in $\check{G}$, we first define the following quantities: $S(q) = I_{|\mathcal{Y}|} - \check{H}(q)^{-1}$, $\tilde{G}(q) = \begin{bmatrix} 0 & \bar{G}_{j\tilde{\mathcal{D}}} \\ \bar{G}_{\tilde{\mathcal{Y}}i} & \bar{G}_{\tilde{\mathcal{Y}}\tilde{\mathcal{D}}} \end{bmatrix}$, and $S_{\mathcal{D}}(q) = (I - S(q))\tilde{G}(q)$, where $|\mathcal{X}|$ denotes the cardinality of set $\mathcal{X}$. With these definitions, we build a predictor from (8.2) with a parameterized G_{ji} as

$$\tilde{w}_{\mathcal{Y}}(t) = (I - S(q)) \begin{bmatrix} G_{ji}(q, \theta) \\ \mathbf{0}_{(|\mathcal{Y}|-1)\times 1} \end{bmatrix} w_i(t) + S_{\mathcal{D}}(q)\tilde{w}_{\mathcal{D}}(t) + S(q)\tilde{w}_{\mathcal{Y}}(t) + \tilde{\xi}_{\mathcal{Y}}(t). \tag{8.3}$$

8.3.1 Vector description of network dynamics

Keeping a parametric model for the target module, we now need to model the other modules. First, we obtain a vector description of the network dynamics for the available N measurements using impulse response of the modules. We stack the first ℓ coefficients of the impulse response of each module in $S_{\mathcal{D}}(q)$ and $S(q)$ as $s_{\mathcal{D}} = \left[s_{Y_1 D_1}^\top, \ldots, s_{Y_{|\mathcal{Y}|} D_{|\mathcal{D}|}}^\top\right]^\top$, and $s_{\mathcal{Y}} = \left[s_{Y_1 Y_1}^\top, \ldots, s_{Y_{|\mathcal{Y}|} Y_{|\mathcal{Y}|}}^\top\right]^\top$, where $Y_1, \ldots, Y_{|\mathcal{Y}|}$ and $D_1, \ldots, D_{|\mathcal{D}|}$ are elements of set $\mathcal{Y}$ and $\mathcal{D}$ respectively. ℓ is chosen sufficiently large to capture the impulse response dynamics. We also represent the target module $G_{ji}(q, \theta)$ as an impulse response, where the first N coefficients are collected in g_{ji} (the dependence on θ is implicit and dropped).

Next we introduce a vector notation for the signal $\tilde{w}_{\mathcal{Y}}(t)$: $\tilde{w}_{\mathcal{Y}} := \left[\tilde{w}_{Y_1}(1) \ldots \tilde{w}_{Y_1}(N)\; \tilde{w}_{Y_2}(1) \ldots \tilde{w}_{Y_{|\mathcal{Y}|}}(N)\right]$. Then, we denote $G_\theta \in \mathbb{R}^{N \times N}$ as the Toeplitz matrix of g_{ji}, $\bar{W}_i \in \mathbb{R}^{N \times \ell}$ as the Toeplitz matrix of $\left[0\ 0\ w_i(1)\ \ldots\ w_i(N-2)\right]^\top$, and $W_i \in \mathbb{R}^{N \times N}$ as the Toeplitz matrix of $\left[0\ w_i(1)\ \ldots\ w_i(N-1)\right]^\top$, and $\breve{W}_k \in \mathbb{R}^{N \times \ell}$ as the Toeplitz of $\left[0\ w_k(1)\ \ldots\ w_k(N-1)\right]^\top$ where k belongs to the elements in $\mathcal{Y}$ and $\mathcal{D}$. We also define the following:

$$W_{\mathcal{Y}} = \begin{bmatrix} W_{Y_1} & \ldots & W_{Y_{|\mathcal{Y}|}} \end{bmatrix} \quad W_{\mathcal{D}} = \begin{bmatrix} W_{D_1} & \ldots & W_{D_{|\mathcal{D}|}} \end{bmatrix}$$

$$\begin{aligned} \tilde{W}_i &= [G_\theta \bar{W}_i\ \ \mathbf{0}] \in \mathbb{R}^{N \times \ell|\mathcal{Y}|}, \\ \tilde{\boldsymbol{W}}_i &= \operatorname{diag}(\tilde{W}_i, \ldots, \tilde{W}_i) \in \mathbb{R}^{N|\mathcal{Y}| \times \ell|\mathcal{Y}|^2}, \\ \boldsymbol{W}_{\mathcal{D}} &= \operatorname{diag}(W_{\mathcal{D}}, \ldots, W_{\mathcal{D}}) \in \mathbb{R}^{N|\mathcal{Y}| \times \ell|\mathcal{D}|^2}, \\ \boldsymbol{W}_{\mathcal{Y}} &= \operatorname{diag}(W_{\mathcal{Y}}, \ldots, W_{\mathcal{Y}}) \in \mathbb{R}^{N|\mathcal{Y}| \times \ell|\mathcal{Y}|^2}. \end{aligned} \tag{8.4}$$

Having defined the above terms, (8.3) can be rewritten in vector form as

$$\tilde{w}_{\mathcal{Y}} = W_{ji} g_{ji} - \tilde{\boldsymbol{W}}_i s_{\mathcal{Y}} + \boldsymbol{W}_{\mathcal{D}} s_{\mathcal{D}} + \boldsymbol{W}_{\mathcal{Y}} s_{\mathcal{Y}} + \xi, \tag{8.5}$$

where $W_{ji} = \left[W_i^\top \mathbf{0}^\top\right]^\top$, and $\xi \in \mathbb{R}^{N|\mathcal{Y}| \times 1}$ is the vectorized noise.

8.3.2 Modeling the additional modules as GP

We now discuss our modeling strategy for the additional modules. Our aim is to increase the accuracy of the desired parameter θ by limiting the number of parameters to be estimated to describe $\tilde{w}_{\mathcal{Y}}$ in (8.5). Therefore, we keep a parametric model for g_{ji} and model the remaining impulse responses in (8.5) as independent zero mean Gaussian Processes (GP) [107]. GP are effective in reducing the variance of the impulse response estimate with suitable choice of a prior covariance matrix (kernel) [93], which we chose to be the *First order Stable Spline kernel* [21]. The kernel structure is given by $K := \lambda K_\beta$ with $[K_\beta]_{x,y} = \beta^{\max(x,y)}$, where $\beta \in [0, 1)$ and $\lambda \geqslant 0$. λ and β are hyperparameters that govern the amplitude and exponential decay of the realization of the Gaussian vector respectively. Therefore, impulse response of any length ℓ can be represented using only the above two hyperparameters λ and β. In addition, the chosen kernel enforces smoothness and stability of the estimate of the impulse responses. Therefore, we have:

$$\begin{aligned} s_{Y_pD_k} &\sim \mathcal{N}(\mathbf{0}, \lambda^D_{pk} K_{\beta^D_{pk}}), p = 1, \dots, |\mathcal{Y}|, k = 1, \dots, |\mathcal{D}| \\ s_{Y_pY_k} &\sim \mathcal{N}(\mathbf{0}, \lambda^Y_{pk} K_{\beta^Y_{pk}}), p = 1, \dots, |\mathcal{Y}|, k = 1, \dots, |\mathcal{Y}|. \end{aligned} \tag{8.6}$$

Each impulse response prior is assigned with independent hyperparameters λ and β for flexibility of modeling. Let us now define, $\boldsymbol{s} = \begin{bmatrix} s_{\mathcal{Y}}^\top & s_{\mathcal{D}}^\top \end{bmatrix}^\top$, $\mathbf{W} = \begin{bmatrix} \boldsymbol{W}_{\mathcal{Y}} - \tilde{\boldsymbol{W}}_i & \boldsymbol{W}_{\mathcal{D}} \end{bmatrix}$ and let $\mathbf{K}$ be the block diagonal matrix constructed with the covariance of the impulse response priors. Using the above definitions, (8.5) can be written as,

$$\tilde{w}_{\mathcal{Y}} = W_{ji} g_{ji} + \mathbf{W}\boldsymbol{s} + \xi. \tag{8.7}$$

In (8.7), $\boldsymbol{s}$ is modeled as Gaussian process. Therefore by considering a Gaussian distribution for noise ξ and also taking into account the noise correlations

$$\xi \sim \mathcal{N}(\mathbf{0}, \bar{\Sigma} \otimes I_N), \; \bar{\Sigma} := \begin{bmatrix} \sigma^2_{11} & \sigma^2_{12} & \cdots & \sigma^2_{1|\mathcal{Y}|} \\ * & \sigma^2_{22} & \cdots & \sigma^2_{2|\mathcal{Y}|} \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & \sigma^2_{|\mathcal{Y}||\mathcal{Y}|} \end{bmatrix}$$

we can write a joint probabilistic description of $\boldsymbol{s}$ and $\tilde{w}_{\mathcal{Y}}$, which is jointly Gaussian, as:

$$p\left(\begin{bmatrix} \mathrm{s} \\ \tilde{w}_{\mathcal{Y}} \end{bmatrix}; \eta\right) \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ W_{ji} g_{ji} \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}\mathbf{W}^\top \\ \mathbf{W}\mathbf{K} & \mathbf{P} \end{bmatrix}\right) \tag{8.8}$$

where, $\mathbf{P} := \Sigma + \mathbf{W}\mathbf{K}\mathbf{W}^\top$, $\Sigma := \bar{\Sigma} \otimes I_N$, and

$$\begin{aligned} \eta = [\theta \; &\lambda^D_{11} \; \dots \; \lambda^D_{|\mathcal{Y}||\mathcal{D}|} \; \lambda^Y_{11} \; \dots \; \lambda^Y_{|\mathcal{Y}||\mathcal{Y}|} \; \beta^D_{11} \; \dots \; \beta^D_{|\mathcal{Y}||\mathcal{D}|} \\ &\beta^Y_{11} \; \dots \; \beta^Y_{|\mathcal{Y}||\mathcal{D}|} \; \sigma^2_{11} \; \dots \; \sigma^2_{1|\mathcal{Y}|} \; \dots \; \sigma^2_{2|\mathcal{Y}|} \; \dots \; \sigma^2_{|\mathcal{Y}||\mathcal{Y}|}]^\top. \end{aligned} \tag{8.9}$$

The parameter vector η governs the probability distribution function in (8.8). It consists of the parameters of $G_{ji}(\theta)$, the hyperparameters of the kernels of the impulse response models and the elements of the covariance of the noise acting on $\tilde{w}_{\mathcal{Y}}$. It is important to note that in EBDM [101] we estimate the variance of the noise corrupting only the output of the target module $w_j(t)$, in contrast to all elements of the covariance matrix of the noise corrupting the signals $w_{\mathcal{Y}}(t)$ to capture the effect of noise correlations. Therefore, to estimate the θ contained in η, we adopt an Empirical Bayes (EB) framework [80]. To this end, we consider the marginal pdf of $\tilde{w}_{\mathcal{Y}}$ by integrating out the effect of $\boldsymbol{s}$ and maximizing the marginal likelihood of $w_{\mathcal{Y}}$. The corresponding objective function is

$$\begin{aligned}\hat{\eta} &= \arg\max_{\eta} p(\tilde{w}_{\mathcal{Y}};\eta)\\ &= \arg\min_{\eta} \log|\mathbf{P}| + (\tilde{w}_{\mathcal{Y}} - W_{ji}g_{ji})^{\top}\mathbf{P}^{-1}(\tilde{w}_{\mathcal{Y}} - W_{ji}g_{ji})\,.\end{aligned} \tag{8.10}$$

This optimization problem is complex and non-convex, and solving such a problem is cumbersome. Therefore, in the next section, we introduce a method to solve the marginal likelihood maximization problem through an iterative scheme.

8.4 Maximizing Marginal Likelihood

For maximizing the marginal likelihood, we consider the iterative method of *Expectation Maximization* (EM) method [35] for obtaining the estimate of η. For this, we need to first define the latent variable whose estimation simplifies the calculation of the marginal likelihood. In this case, we choose $\boldsymbol{s}$. The EM method guarantees convergence to a local minima [16] and the optimization problem is simplified as seen in *Lemma 7.2* compared to solving the original problem in (8.10). The EM method has two steps,

- *E-step*: Given $\hat{\eta}^{(n)}$ at the n^{th} iteration, compute

$$Q^{(n)}(\eta) = \mathbb{E}_{p(\boldsymbol{s}|\tilde{w}_{\mathcal{Y}};\hat{\eta}^{(n)})}[\log p(\tilde{w}_{\mathcal{Y}}, \boldsymbol{s};\eta)], \tag{8.11}$$

- *M-step*: Compute $\hat{\eta}^{(n+1)}$ from

$$\hat{\eta}^{(n+1)} = \arg\max_{\eta} Q^{(n)}(\eta). \tag{8.12}$$

The estimate $\hat{\eta}$ is obtained by iterating between (8.11) and (8.12) until the parameters converge. Although the procedure is iterative, the EM algorithm significantly simplifies solving (8.10), reasons for which are shown in our next steps.

The posterior distribution of $\boldsymbol{s}$ given $\tilde{w}_{\mathcal{Y}}$ for an estimate of η is Gaussian, given by $p(\boldsymbol{s}|\tilde{w}_{\mathcal{Y}};\eta) \sim \mathcal{N}(\boldsymbol{s}_m, P_{\boldsymbol{s}})$ [2] where

$$\begin{aligned}P_{\boldsymbol{s}} &= \mathbf{K} - \mathbf{K}\mathbf{W}^{\top}(\mathbf{W}\mathbf{K}\mathbf{W}^{\top} + \Sigma)^{-1}\mathbf{W}\mathbf{K},\\ \boldsymbol{s}_m &= (\mathbf{K}\mathbf{W}^{\top}(\mathbf{W}\mathbf{K}\mathbf{W}^{\top} + \Sigma)^{-1})(\tilde{w}_{\mathcal{Y}} - W_{ji}g_{ji}).\end{aligned} \tag{8.13}$$

Let $\hat{\boldsymbol{s}}^{(n)}$ and $\hat{P}_{\boldsymbol{s}}^{(n)}$ be the posterior mean and covariance of $\boldsymbol{s}$ obtained from (8.13) using $\hat{\eta}^{(n)}$, we define $\hat{\boldsymbol{S}}^{(n)} := \hat{P}_{\boldsymbol{s}}^{(n)} + \hat{\boldsymbol{s}}^{(n)}\hat{\boldsymbol{s}}^{(n)\top}$ and each of its $\ell \times \ell$ diagonal block as $\hat{\boldsymbol{S}}_m^{(n)}$ which are the posterior second moment of $\hat{\boldsymbol{s}}_m^{(n)}$. Here, m corresponds to each combination of the impulse response in (8.6) and its respective hyperparameters.

The structure of $Q^{(n)}(\eta)$ in (8.11) for the setup in (8.10) is provided in the following lemma.

Lemma 8.1 *Let* $\hat{\eta}^{(n)}$ *be the estimate of* η *at* n^{th} *iteration of the EM algorithm according to* (8.12)*, then*

$$Q^{(n)}(\eta) = Q_0^{(n)}(\theta, \Sigma) + \sum_m Q_{s_m}^{(n)}(\lambda_m, \beta_m) \tag{8.14}$$

where,

$$\begin{aligned} Q_0^{(n)}(\theta, \Sigma) = & - \log\det\Sigma - \boldsymbol{tr}\Big(\Sigma^{-1}\Big(\tilde{w}_{\mathcal{Y}}\tilde{w}_{\mathcal{Y}}^\top + \mathbf{W}\hat{\boldsymbol{S}}^{(n)}\mathbf{W}^\top \\ & + W_{ji}g_{ji}g_{ji}^\top W_{ji}^\top - W_{ji}g_{ji}\tilde{w}_{\mathcal{Y}}^\top - \tilde{w}_{\mathcal{Y}}g_{ji}^\top W_{ji}^\top - \mathbf{W}\hat{\boldsymbol{s}}^{(n)}\tilde{w}_{\mathcal{Y}}^\top \\ & - \tilde{w}_{\mathcal{Y}}\hat{\boldsymbol{s}}^{(n)\top}\mathbf{W}^\top + \mathbf{W}\hat{\boldsymbol{s}}^{(n)}g_{ji}^\top W_{ji}^\top + W_{ji}g_{ji}\hat{\boldsymbol{s}}^{(n)\top}\mathbf{W}^\top\Big)\Big), \\ Q_{s_m}^{(n)}(\lambda_m, \beta_m) = & - \log\det\lambda_m K_{\beta_m} - \frac{1}{\lambda_m}\boldsymbol{tr}\Big(K_{\beta_m}^{-1}\hat{\boldsymbol{S}}_m^{(n)}\Big) . \end{aligned}$$

Proof: See the appendix. ■

It is indeed seen that (8.11) splits into a summation of simpler terms that depend on different elements of parameter vector η. Therefore, the update of η splits into many independent and simpler optimization problems, that can be computed in parallel.

Update of kernel hyperparameters

It can be seen that the kernel hyperparameters can be updated independently of the rest of the parameters. The kernel hyperparameters are updated as per the *Theorem* **??** [14, 38].

Theorem 8.1 *Define*

$$Q_{\beta_m}^{(n)}(\beta_m) = \ell \log \boldsymbol{tr}(K_{\beta_m}^{-1}\hat{\boldsymbol{S}}_m^{(n)}) + \log\det K_{\beta_m}. \tag{8.15}$$

Then,

$$\begin{aligned} \hat{\beta}_m^{(n+1)} &= \arg\min_{\beta_n \in [0,1)} Q_{\beta_m}^{(n)}(\beta_m) \\ \hat{\lambda}_m^{(n+1)} &= \frac{1}{\ell}\boldsymbol{tr}(K_{\hat{\beta}_m^{(n+1)}}^{-1}\hat{\boldsymbol{S}}_m^{(n)}). \end{aligned} \tag{8.16}$$

Proof: See the appendix. ■

The optimization problem in (8.15) is a scalar optimization in the domain [0,1) and computationally fast. The update of $\hat{\lambda}_m^{(n+1)}$ has a closed form solution, requiring no optimization. Therefore, the hyperparameters update becomes simple.

Update of θ and noise covariance

The updates of θ and the noise covariance parameters in η are independent of the kernel hyperparameters. Following a similar reasoning in [3], θ and Σ are updated as per the *Theorem 7.2*.

Theorem 8.2 *Define*

$$Q_\theta^{(n)}(\theta) = \det\left(\sum_{t=1}^{N} P_\xi(t)\right) .$$

Then

$$\begin{aligned} \hat{\theta}^{(n+1)} &= \arg\min_{\theta} Q_\theta^{(n)}(\theta) \ , \\ \hat{\Sigma}^{(n+1)} &= \frac{1}{N}\left(\sum_{t=1}^{N} P_{\hat{\xi}^{(n+1)}}(t)\right) \otimes I_N \ . \end{aligned} \tag{8.17}$$

Here, $P_\xi(t)$ is computed based on $\hat{\eta}^{(n)}$ and $\hat{s}^{(n)}$, whereas $P_{\hat{\xi}^{(n+1)}}$ is computed based on $\hat{\theta}^{(n+1)}$, $\hat{\lambda}_m^{(n)}$, $\hat{\beta}_m^{(n)}$ and $\hat{s}^{(n)}$.

The expression for computing $P_\xi(t)$ is provided in the appendix. From *Theorem* 7.2, Σ is updated using a closed form expression, requiring minimal computation. Except for θ that requires solving a non-linear optimization problem at each iteration, all other updates are simple and computationally effective, which is significantly more efficient compared to solving the non-linear optimization problem in PEM with all modules parameterized in the MIMO setup. The steps for estimating $\hat{\eta}$ is provided in Algorithm 2. Initialization can be done by randomly choosing η subject to the constraints of the hyperparameters. For terminating the algorithm, the convergence criteria is defined as $\frac{\|\hat{\eta}^{(n)}-\hat{\eta}^{(n-1)}\|}{\|\hat{\eta}^{(n-1)}\|} < 10^{-5}$.

8.5 Numerical Simulations

Numerical simulations are performed to validate and illustrate the developed method. To this end, we consider the dynamic network shown in Figure 8.1 with 3 nodes. The network is excited using known external excitation signals $r_1(t)$ and $r_3(t)$ that are realizations of white noise with unit variance. The process noises of node 2 and 3 are correlated. In this network, we intend to identify the dynamics of the module G_{21}^0 (green module). We run 50 independent Monte Carlo simulations obtaining $N = 500$ data each time. The noise sources $e_1(t)$, $e_2(t)$ and $e_3(t)$ have variances of 0.1, 0.2 and 0.3 respectively. We assume that we know the

Algorithm 2 Algorithm for identifying a local module in a dynamic network with correlated noise

```
Input:$\{w_k\}_{t=1}^{N}, k \in \mathcal{Y} \cup \mathcal{D}$
Output: $\hat{\theta}$

1. Set $n = 0$, Initialize $\hat{\eta}^{(0)}$.
2. Compute $\hat{\boldsymbol{P}}_s^{(n)}$, $\hat{\boldsymbol{s}}$, and $\hat{\boldsymbol{S}}^{(n)}$.
3. Update the kernel hyperparameters of all the impulse responses in (8.6), $\hat{\beta}_m^{(n+1)}$ and $\hat{\lambda}_m^{(n+1)}$ using (8.16).
4. Update $\hat{\theta}^{(n+1)}$ and $\hat{\Sigma}^{(n+1)}$ using (8.17).
5. Set $\hat{\eta}^{(n+1)}$ based on (8.9).
6. Set $n = n + 1$.
7. Repeat steps (2) to (6) until convergence.
```

model order of G_{21}^0. The dynamics of all the modules and the noise models are given in (8.18).

$$
\begin{aligned}
G_{21}^0 &= \tfrac{b_1q^{-1}+b_2q^{-2}}{1+a_1q^{-1}+a_2q^{-2}} & &= \tfrac{1q^{-1}+0.5q^{-2}}{1+0.8q^{-1}+0.6q^{-2}}\\
G_{31}^0 &= \tfrac{-2.1q^{-1}+2.4q^{-2}}{1-0.9q^{-1}-0.1q^{-2}} & G_{12}^0 &= \tfrac{0.03(q^{-1}+q^{-2})}{1+1.9q^{-1}+0.9q^{-2}}\\
G_{23}^0 &= \tfrac{-0.2q^{-1}+0.02q^{-2}}{1-0.2q^{-1}-0.1q^{-2}} & H_{11}^0 &= \tfrac{1+0.1q^{-1}-0.03q^{-2}}{1+0.5q^{-1}+0.1q^{-2}}\\
H_{22}^0 &= \tfrac{1+1.5q^{-1}-0.2q^{-2}}{1+0.1q^{-1}-0.01q^{-2}} & H_{33}^0 &= \tfrac{1-0.4q^{-1}+0.1q^{-2}}{1-0.4q^{-1}+0.1q^{-2}}\\
H_{23}^0 &= \tfrac{0.3q^{-1}-0.01q^{-2}}{1-0.4q^{-1}-0.6q^{-2}} & H_{32}^0 &= \tfrac{q^{-1}-q^{-2}}{1-1.9q^{-1}+0.9q^{-2}}.
\end{aligned}
\tag{8.18}
$$

According to the local direct method [104], among the inputs $\{w_1, w_3\}$ that contribute to the output of the target module w_2, the noise correlation between the input w_3 and output w_2 can be handled by adding w_3 (common signal) to the output, thereby covering the noise correlation by a (2× 2) noise modeling. Therefore, the input and output nodes of the MIMO estimation setup are given by $w_{\mathcal{D}} = \{w_1, w_3\}$ and $w_{\mathcal{Y}} = \{w_2, w_3\}$. We choose $\ell = 100$ for the length of impulse response vectors of the additional modules. To assess the performance of the developed method (named as *Empirical Bayes Local Direct Method* (EBLDM) for comparison), we compare it with the *Direct method* (DM) [124] and the *Two Stage Method* (TS) [124]. In the case of DM, we solve a 2-input/1- output MISO identification problem with $w_1(t)$ and $w_3(t)$ as inputs and $w_2(t)$ as output. In the two-stage method, the projection of the two inputs on the external signals $r_1(t)$ and $r_3(t)$ are used as inputs to the MISO identification problem. Furthermore, to improve the accuracy of the estimate obtained by the Two Stage method, we also

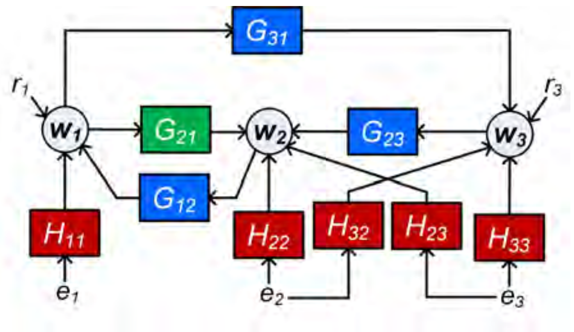


Figure 8.1: A 3 node network with process noise correlated between the nodes 2 and 3: The target module is G_{21} (green box).

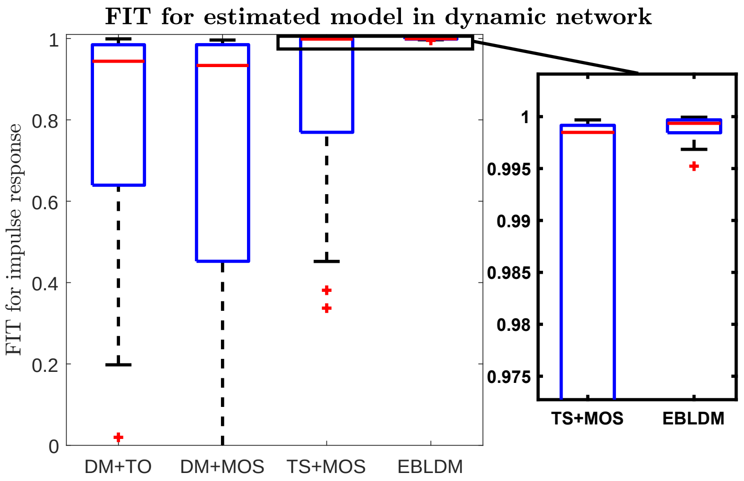


Figure 8.2: Box plot of fit of the impulse response of $\hat{G}_{21}$ obtained by the two stage method, direct method and the developed method.

identify the noise model. For both these methods, we use the *Akaike Information Criteria* (AIC) for selecting a suitable model order.

The box plot of the fit of impulse response of G^0_{21} is shown in Figure 8.2, where we have compared the performances of the direct method with true model order and the same method with model order selection step ('DM+TO' & 'DM+MOS'), the two stage method with model order selection step ('TS+MOS') and the EBLDM. The EBLDM has better overall fit of the impulse response than the classical methods. On comparing the bias and standard deviation plot of the parameters of $\hat{G}_{21}$, shown in Figure 8.3, it is evident that the EBLDM provides a smaller bias and substantially reduced variance of the estimated parameters. The reduced variance is attributed to the regularization approach of this method. Among the other methods, the two stage method achieves smaller bias and variance than the direct method. A significant bias in the estimated parameters

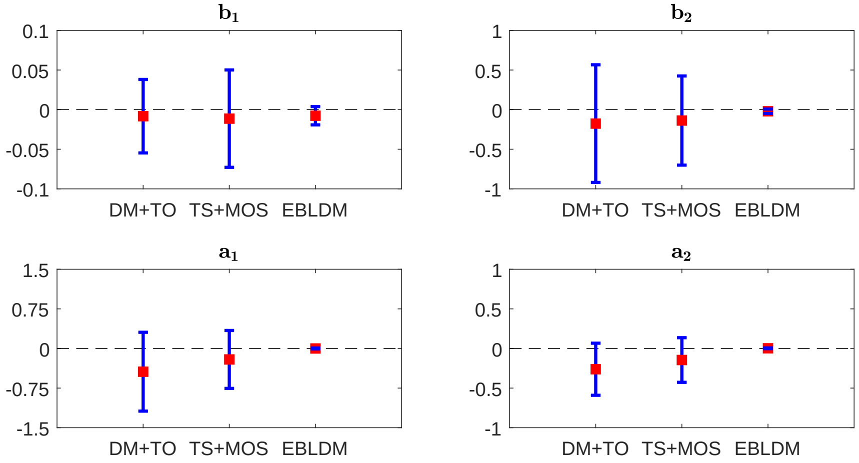


Figure 8.3: Bias and standard deviation of the estimate of target module parameters

can be witnessed in the case of 'DM+TO' from Figure 8.3. This is in accordance with the theory that the direct method with the chosen MISO identification setup provides biased estimates under the situation of correlated noise, however, a MIMO identification setup (as in EBLDM) does not (see [104]). Overall, the developed EBLDM method proves effective for the considered relatively small network. As the size of the network grows, the results of the classical methods may further deteriorate due to the increase in number of parameterized modules and model order selection step that needs to be performed for it. Concerning this situation, EBLDM can stand out as an effective method by circumventing the model order selection step and providing reduced variance for large sized networks.

8.6 Conclusion

Building on the EBDM, an effective algorithm for the network MIMO estimation problem that is required to identify a module in a dynamic network with correlated noise has been developed. The developed method circumvents the model order selection step for all the modules that are not of interest to the experimenter but needs to be identified for unbiased estimate of the target module. Furthermore, it uses the regularized non-parametric methods to reduce the number of estimated parameters, which reduces mean squared error of the estimated target module. Numerical simulation with an example network emphasize the potential of the introduced method in comparison with the available classical methods.

8.7 Related videos

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

A regularized kernel-based method for learning a module in a dynamic network with correlated noise

Venkatakrishnan Rajagopal, Karthik Ramaswamy, Paul Van den Hof

59th IEEE Conference on Decision and Control, December 14-18, 2020, Jeju Island, Republic of Korea

www.vpsignet.eu
k.r.ramaswamy@tue.nl

Regularized kernel-based method for learning a module

Appendices

8.A Proof of Lemma 8.1

Following Bayes' theorem, (8.11) can be written as follows,

$$Q^{(n)}(\eta) = \mathbb{E}[\log p(\tilde{w}_{\mathcal{Y}}|\boldsymbol{s};\eta)] + \mathbb{E}[\log p(\boldsymbol{s};\eta)] \tag{8.19}$$

Define

$$\begin{aligned}\mathcal{A} = &-\frac{N|\mathcal{Y}|}{2}\log(2\pi) - \frac{1}{2}\log\det\Sigma \\ &- \frac{1}{2}\left(\tilde{w}_{\mathcal{Y}} - \begin{bmatrix} W_i \\ \mathbf{0} \end{bmatrix} g_{ji} - \mathbf{W}\boldsymbol{s}\right)^\top \Sigma^{-1}\left(\tilde{w}_{\mathcal{Y}} - \begin{bmatrix} W_i \\ \mathbf{0} \end{bmatrix} g_{ji} - \mathbf{W}\boldsymbol{s}\right),\end{aligned} \tag{8.20}$$

$$\mathcal{B} = \sum_m \left(-\frac{\ell}{2}\log(2\pi) - \frac{1}{2}\log\det\lambda_m K_{\beta_m} - \frac{1}{2}s_m^\top(\lambda_m K_{\beta_m})^{-1}s_m \right) \tag{8.21}$$

Using properties of trace, removing constant terms, multiplying by 2 and taking the expectation with respect to posterior, (8.20) and (8.21) are written as,

$$\begin{aligned}\mathbb{E}[\mathcal{A}] = -\log\det\Sigma - \mathbf{tr}\Bigg(\Sigma^{-1}\Bigg(&\tilde{w}_{\mathcal{Y}}\tilde{w}_{\mathcal{Y}}^\top + \begin{bmatrix} W_i \\ \mathbf{0} \end{bmatrix} g_{ji}g_{ji}^\top \begin{bmatrix} W_i^\top & \mathbf{0}^\top \end{bmatrix} + \mathbf{W}\boldsymbol{s}\boldsymbol{s}^\top\mathbf{W}^\top \\ &- \begin{bmatrix} W_i \\ \mathbf{0} \end{bmatrix} g_{ji}\tilde{w}_{\mathcal{Y}}^\top - \mathbf{W}\boldsymbol{s}\tilde{w}_{\mathcal{Y}}^\top - \tilde{w}_{\mathcal{Y}}g_{ji}^\top \begin{bmatrix} W_i^\top & \mathbf{0}^\top \end{bmatrix} + \mathbf{W}\boldsymbol{s}g_{ji}^\top \begin{bmatrix} W_i^\top & \mathbf{0}^\top \end{bmatrix} \\ &- \tilde{w}_{\mathcal{Y}}\boldsymbol{s}^\top\mathbf{W}^\top + \begin{bmatrix} W_i \\ \mathbf{0} \end{bmatrix} g_{ji}\boldsymbol{s}^\top\mathbf{W}^\top\Bigg)\Bigg),\end{aligned} \tag{8.22}$$

$$\mathbb{E}[\mathcal{B}] = \sum_m -\log\det\lambda_m K_{\beta_m} - \mathbf{tr}\left((\lambda_m K_{\beta_m})^{-1}s_m s_m^\top\right) \tag{8.23}$$

By substituting, $\boldsymbol{s}$ as $\hat{\boldsymbol{s}}^{(n)}$, $\boldsymbol{s}\boldsymbol{s}^\top$ as $\hat{\boldsymbol{S}}^{(n)}$, s_m as $\hat{s}_m^{(n)}$ and $s_m s_m^\top$ as $\hat{S}_m^{(n)}$ in (8.22) and (8.23), (8.14) is obtained.

8.B Proof of Theorem 8.1

We consider $Q_{s_m}(\lambda_m, \beta_m)$ in (8.14) and differentiate it with respect to λ_m. The derivative is then equated to 0 to obtain the expression for λ_m

$$\lambda_m = \frac{1}{\ell}\text{tr}(K_{\beta_m})^{-1}\hat{S}_m. \tag{8.24}$$

(8.24) is then substituted in the $Q_{s_m}(\lambda_m, \beta_m)$ to eliminate λ_m, and with the change of sign, resulting in the following equation.

$$Q_B(\beta_m) = \ell \log \text{tr}(K_{\beta_m}^{-1}\hat{S}_m) + \log\det K_{\beta_m} \tag{8.25}$$

Once β_m is obtained, then (8.24) is used to obtain λ_m.

8.C Computation of $\hat{P}_\xi$

Let us define the matrices

$$\begin{aligned} W_1(t) &= \left[\mathbf{W}(t,*)^\top \mathbf{W}(t{+}N,*)^\top \ldots \mathbf{W}(t{+}(N_\mathcal{Y}-1)N,*)^\top\right]^\top \\ W_2(t) &= \left[W_{ji}(t,*)^\top W_{ji}(t{+}N,*)^\top \ldots W_{ji}(t{+}(N_\mathcal{Y}{-}1)N,*)^\top\right]^\top \end{aligned}$$

where, $\mathbf{W}(t,*)$ corresponds to the t^{th} row of the matrix $\mathbf{W}$. With the above definitions, we define

$$\begin{aligned} &P_\xi(t) = \tilde{w}_\mathcal{Y}(t)\tilde{w}_\mathcal{Y}^\top(t) + W_2(t)g_{ji}g_{ji}^\top W_2^\top(t) \\ &+ W_1(t)\hat{\boldsymbol{S}}^{(n)}W_1^\top(t) - W_2(t)g_{ji}\tilde{w}_\mathcal{Y}^\top(t) - W_1(t)\hat{\boldsymbol{s}}^{(n)}\tilde{w}_\mathcal{Y}^\top(t) \\ &- \tilde{w}_\mathcal{Y}(t)g_{ji}^\top W_2^\top(t) + W_1(t)\hat{\boldsymbol{s}}^{(n)}g_{ji}^\top W_2^\top(t) \\ &- \tilde{w}_\mathcal{Y}\hat{\boldsymbol{s}}^{(n)\top}W_1^\top(t) + W_2(t)g_{ji}\hat{\boldsymbol{s}}^{(n)\top}W_1^\top(t) \end{aligned} \tag{8.26}$$

$P_{\hat{\xi}^{(n+1)}}(t)$ is obtained by updating $\hat{g}_{ji}^{(n+1)}$ and recomputing (8.26).

9 CHAPTER

Learning local modules without prior topology information

Different identification methods have been developed for identifying a single module in a dynamic network, addressing aspects like the choice of signals that need to be measured, the presence of correlated disturbances, and scalability of the estimation algorithms by employing machine learning techniques. In order to select an appropriate predictor model one typically needs prior knowledge on the topology (interconnection structure) of the dynamic network, as well as on the correlation structure of the process disturbances. In this chapter we present a new approach that incorporates the estimation of this prior information into the identification, leading to a fully data-driven approach for estimating the dynamics of a local module. The developed algorithm uses non-causal Wiener filters and a series of convex optimizations with parallel computation capabilities to estimate the topology, which subsequently is used to build the appropriate input/output setting for a predictor model in the local direct method under correlated process noise. A regularized kernel-based method is then employed to estimate the dynamics of the target module, while non-parametrically handling the remaining modules that are present in the predictor model. This leads to an identification algorithm with attractive statistical properties that is scalable to handle larger-scale networks too. Numerical simulations illustrate the potential of the developed algorithm.

This chapter is based on the publication: V.C. Rajagopal, K.R. Ramaswamy, and P.M.J. Van den Hof, "Learning local modules in dynamic networks without prior topology information", *Proc. 60th IEEE Conf. Decision and Control (CDC)*, December 13-15, 2021, Austin, TX, USA, pp. 840-845.

9.1 Introduction

Local module identification focuses on identifying the dynamics of a single module embedded in a network, which includes the problem of selecting the relevant node signals to be measured. In a prediction error setting this problem was addressed in [27, 29, 54, 105, 124, 141]. We have addressed the same problem in Chapter 3 and 4 as well. The local direct identification method is presented in [104] (Chapter 3) and contains a full procedure for minimum variance estimation of a local module for the situation of correlated disturbances, and a variety of options for selecting the measured node signals. However, the current prediction error approach has two limitations:

(i) for constructing the appropriate predictor model it requires a prior knowledge of the topology of the network and the correlation structure of the disturbances; and

(ii) the computational algorithm for estimating the module typically requires estimating large number of nuisance parameters and solving a model order selection problem which is considered not to be scalable to large dimensions.

Recently solutions have become available for problem (ii), in the form of regularized kernel-based approaches (see Chapter 7 and 8) for identifying a single module ([99, 102]), building upon the signal selection procedures developed in Chapter 3.

In this chapter, we will tackle problem (i) by extending the regularized kernel-based method with a dedicated algorithm for estimating the network topology and noise topology to arrive at a full data-driven procedure for learning the dynamics of a single module. For identifying the two topologies (of the network and the noise), we exploit a procedure developed in [85] for network topology, based on the sparsity properties of non-causal Wiener filters, and extend it to networks with a non-diagonal disturbance model.

The remaining part of the chapter is structured as follows. After an introduction of the basic concepts, we sketch the main steps of the algorithm in Section 9.2. Next, we estimate the locality of the relevant node signals in Section 9.3. In Section 9.4 we present the non-causal Wiener filter approach for estimating the topologies of network and noise model, while the connection with the kernel-based method for estimating the dynamics of the target module is presented in Section 9.5. Finally, numerical simulations are provided in Section 9.6.

9.2 Problem setting and approach

We consider the dynamic network described in Chapter 2 with full rank correlated process noise. Similar to Chapter 3, without loss of generality, we consider that $u = 0$. One of the modules in the network G_{ji} is assigned to be the target module of which the dynamics needs to be identified from (a selection of) measured node signals. While there are prediction error methods that explicitly exploit the presence of measured excitation signals u (or r), the local direct method of identification [104, 128] is built on the basis of a predictor model

$$w_{\mathcal{Y}}(t) = \bar{G}(q)w_{\mathcal{D}}(t) + \breve{v}(t) \tag{9.1}$$

where $\mathcal{Y}$, $\mathcal{D}$ are selections of node signals that are chosen on the basis of the presumed topology of the network, and the noise topology. Typical elements in the selection of these sets of node signals include the conditions in Theorem 3.1 in Chapter 3. Under these conditions the local direct method can provide a consistent and minimum variance estimate of the target module. However these conditions can only be satisfied if the topology of the network, i.e. the binary structure of $G^0(q)$, and the correlation structure of the disturbances, reflected by the binary structure of $H^0(q)$ are known. For an accurate estimation of the target module, we will now exploit the following strategy consisting of three steps:

1. Step 1: Select the node signals that carry information on the estimation problem of the target module, by estimating the "locality" of the output node by generalizing the Wiener filter based approach from [85] to situations with non-diagonal noise models;
2. Step 2: Estimate the *local topology* of the network around the target module, by applying a model structure selection procedure based on the AIC selection criterion and a multi-step least squares algorithm (from Chapter 6) with analytical solutions and with parallel computation capabilities.
3. Step 3: Select an appropriate predictor model and estimate the target module dynamics by exploiting an Empirical Bayes method, in which the target module is parametrized while the remaining nuisance modules are modelled as Gaussian processes. This leads to an attractive and scalable estimation algorithm with a limited number of parameters to be estimated while achieving small mean-squared errors.

9.3 Locality detection

9.3.1 Graph aspects

Before we identify the nodal and noise interconnection structure, we need to identify the nodes that contain information about the output node. In this study, we refer to such a set of nodes as the locality of the output node. The locality of a node can be seen as the Markov blanket [89] of the node in a network with correlated process noise. The Markov blanket of a node is the set of nodes that contain all the information about the node and is defined for Bayesian networks. Similarly, we need to define *locality* for networks with correlated process noise. To this end, we define the following sets.

Definition 9.1 *Given a dynamic network denoted by* $(G, H)^a$*, the children of a node* j *are defined by* $\mathcal{C}_j := \{i | G_{ij} \neq 0\}$*; its parents are defined by* $\mathcal{P}_j := \{i | G_{ji} \neq 0\}$*, and its noise confounders by* $\mathcal{V}_j := \{i \mid \Phi_{v_j v_i} \neq 0\}$*.*

[a] Superscript ^{0} is dropped for convenience.

Usual practice in literature is to represent the topology of the network by an adjacency matrix [124]. However, since in this study, we identify both the network and the noise topology, we formally redefine them as follows.

Definition 9.2 (Topology) *For a dynamic network* (G, H)*, the topologies* $\mathcal{T}_G$ *and* $\mathcal{T}_H$ *are defined as:*

$$\mathcal{T}_G(j,i) = \begin{cases} 1, & G_{ji} \neq 0 \\ 0, & G_{ji} = 0 \end{cases} \quad \text{and } \mathcal{T}_H(j,i) = \begin{cases} 1, & H_{ji} \neq 0 \\ 0, & H_{ji} = 0 \end{cases} .$$

With the help of the defined quantities, we now define the locality of the output node j.

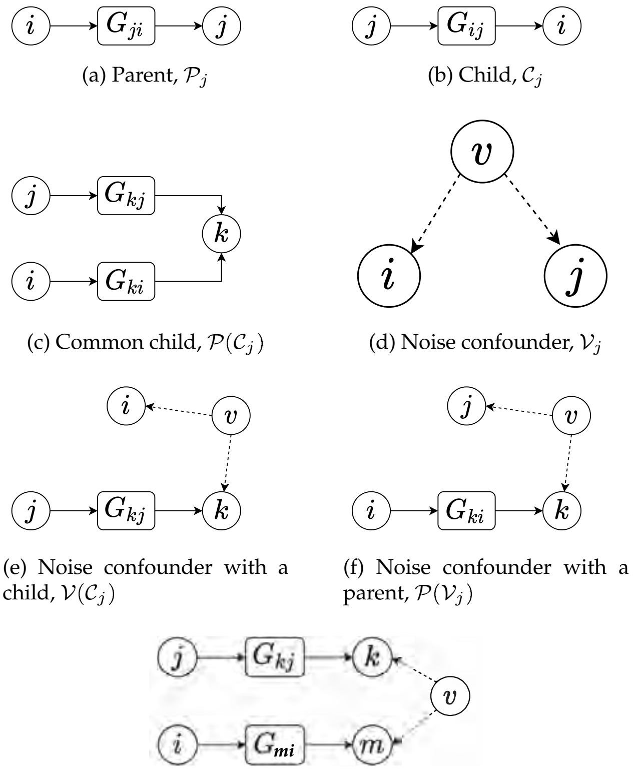
Definition 9.3 (Locality of a node) *Let* j *be a node in the considered dynamic network. The locality of the node* j *(see figure 9.1 for visual representation) is defined by*

$$\mathcal{LOC}_j := \{i \mid i = j \ \lor \ i \in \mathcal{P}_j \ \lor \ i \in \mathcal{C}_j \ \lor \ i \in \mathcal{P}(\mathcal{C}_j) \ \lor \ i \in \mathcal{V}_j$$
$$\lor \ i \in \mathcal{V}(\mathcal{C}_j) \ \lor \ i \in \mathcal{P}(\mathcal{V}_j) \ \lor \ i \in \mathcal{P}(\mathcal{V}(\mathcal{C}_j))\}.$$

The process of removing a set of node signals from the network while keeping the remaining node signals invariant and preserving the second-order properties of the remaining nodes is called immersion (refer to [29] for details). Based on the way we define the locality, immersing the remaining nodes (i.e. nodes other than the nodes corresponding to set $\mathcal{LOC}_j$) in the network does not affect the output node or the interconnection structure of the output node. As a result, a smaller network for which we need to estimate the topology is obtained, which is referred to as the *local network* and is denoted by $(\tilde{G}, \tilde{H})$.

9.3.2 Identifying the Locality

Locality of the output node is analogous to the Markov blanket of the output node defined for Bayesian Networks. In [85], it has been shown that a multivariate non-causal Wiener filter [70] computed by projecting the node j on to the remaining nodal signals can detect the Markov blanket of the corresponding node by analyzing the sparsity of the Wiener filter. However, this sparsity result was developed for networks without a noise model. In this section, we extend the results of [85] to networks with non-diagonal noise model.



(a) Parent, $\mathcal{P}_j$

(b) Child, $\mathcal{C}_j$

(c) Common child, $\mathcal{P}(\mathcal{C}_j)$

(d) Noise confounder, $\mathcal{V}_j$

(e) Noise confounder with a child, $\mathcal{V}(\mathcal{C}_j)$

(f) Noise confounder with a parent, $\mathcal{P}(\mathcal{V}_j)$

(g) Noise confounder among children, $\mathcal{P}(\mathcal{V}(\mathcal{C}_j))$

Figure 9.1: Visual representation of the relationship between nodes i and j for the node i to be in the locality of j

Theorem 9.1 (Sparsity of the non-causal Wiener filter) *Define $w := (w_1, \ldots, w_L)^T$ to be the nodal signals obtained from the dynamic network (G, H) and let the non-causal Wiener filter estimate of w_j on the basis of all other node signals in w be given by $\hat{w}_j(t) = \sum_{i \neq j} W_{ji}(q) w_i(t)$. Then $W_{ji}(z) \neq 0$ implies $w_i \in \mathcal{LOC}_j$.*

Proof: See Appendix 9.A ■

As the result of Theorem 9.1, in order to identify the locality of node w_j, we simply need to compute the multivariate non-causal Wiener filter by projecting the node w_j on the remaining nodes in the network and group the nodes that have a non-zero Wiener filter entry. Although there is no strict guarantee that in this way we detect *all* nodes in the locality, examples under which a true link between two nodes remains undetected are pathological [85]. In a practical implementation a threshold ρ needs to be defined where $\|W_{ji}\|_\infty > \rho$ is taken as

Algorithm 3 Identifying the locality ($\mathcal{LOC}_j$) of a node j

```
Input: $\{w_k(t)\}_{t=1}^{N}, k \in \{1, \dots, L\}$
Output: $\mathcal{B}_j$

1. Set $\bar{w} = \{w_k\}_{k \neq j}$.
2. Compute a multivariate, non-causal $W_j(z)$ with $j$ as output and $\bar{w}$ as input.
3. Initialize $\mathcal{B}_j = \varnothing$
4. for $i = 1 : L \backslash j$
    (a) if $\|W_{jk}\|_\infty > \rho$, add $k$ to $\mathcal{B}_j$
    (b) continue
```

evidence for the corresponding Wiener filter entry to be non-zero. The value of ρ can be chosen based on the signals w_j and w_i. For ease of computation, the Wiener filter is implemented as a non-causal FIR filter of length F, see e.g. [122]. The algorithm for identifying the locality $\mathcal{LOC}_j$ is given in Algorithm 3.

9.4 Topology estimation

In this section, we develop a method to identify the network and noise topology ($\mathcal{T}_{\tilde{G}}, \mathcal{T}_{\tilde{H}}$) of the local network ($\tilde{G}, \tilde{H}$) obtained after immersing the nodes that are absent in the locality of the output node. Numerous solutions are provided in the literature for network topology estimation considering the noise topology to be known and diagonal. In this chapter, we address the situation of having an unknown noise topology that can be non-diagonal. We follow a multi-step approach (similar to the method in Chapter 6):

1. First, we estimate the innovation process using a high-order (non-parametric) model, following an identification algorithm for networks presented in [140].

2. Next we treat this estimate as an additionally available input signal and simultaneously estimate the network and noise topology.

9.4.1 Innovation estimation

The network equation of the local network is described as,

$$w_{\mathcal{C}}(t) = \tilde{G} w_{\mathcal{C}}(t) + \tilde{H} \xi_{\mathcal{C}}(t). \tag{9.2}$$

In (9.2), $w_{\mathcal{C}}$ are the nodal signals of $\mathcal{LOC}_j$, $\tilde{G}$ is a hollow, strictly proper transfer function matrix, and $\tilde{H}$ is square, monic, stable and minimum phase, $\xi_{\mathcal{C}}(t)$ is a white noise vector (see [104] for more details).

Proposition 9.1 *For every dynamic network of the form* (9.2)*, there exists a network representation,*

$$Qw_{\mathcal{C}}(t) = Pw_{\mathcal{C}}(t) + \xi_{\mathcal{C}}(t), \tag{9.3}$$

where, Q is a diagonal and monic transfer function matrix, and P is a hollow, strictly proper transfer function matrix.

Proof: See Appendix 9.B ■

Every entry present in Q and P can be represented as an independent impulse response where its length ℓ_{ARX} is chosen sufficiently long to capture all the dynamics. By approximating Q and P with (high-order) polynomials, we get an ARX representation of the network in (9.3) where each row of Q and P can be identified independently in a parallel MISO setup. Parameterizing the coefficients of the impulse response of modules at each m^{th} row and representing the parameter vector as η^m_{ARX}, the parameter vector can be estimated using the prediction error method [77]. Considering the one-step ahead predictor [77],

$$\hat{w}_m(t|t-1;\eta^m_{ARX}) = \varphi^\top_m(t)\eta^m_{ARX} \tag{9.4}$$

with $\varphi^\top_m(t)$ representing the data matrix and the resulting prediction error

$$\varepsilon_m(t,\eta^m_{ARX}) = w_m(t) - \hat{w}_m(t|t-1;\eta^m_{ARX}) = w_m(t) - \varphi^\top_m(t)\eta^m_{ARX}, \tag{9.5}$$

the parameter vector can be consistently estimated [78] by minimizing the sum of squared prediction errors. This will result in a closed form solution,

$$\hat{\eta}^m_{ARX} = \left[\frac{1}{N}\sum_{t=1}^{N}\varphi_m(t)\varphi^\top_m(t)+R^m\right]^{-1}\cdot\frac{1}{N}\sum_{t=1}^{N}\varphi_m(t)w_m(t), \tag{9.6}$$

where $R^m = \text{diag}(R^m_1,\ldots,R^m_{|\mathcal{LOC}_j|})$ is a regularization matrix to handle the excessive variance of the estimate. The regularization term R^m_k for $k \in \mathcal{LOC}_j$ is chosen to be a modified Tuned/Correlated (TC) kernel [21] as it enforces stability. The modified TC kernel has the following structure

$$R^m_k = \text{diag}(1,\frac{1}{\alpha_k},\ldots,\frac{1}{\alpha_k^{\ell_{ARX}-1}}). \tag{9.7}$$

Here, the coefficients α_k represent the decay rate of the impulse response of the corresponding module that are estimated by cross-validation [21]. An estimate of the innovation $\xi_{\mathcal{C}}(t)$ is obtained as $\hat{\xi}_{\mathcal{C}}(t) = \text{vec}\{\varepsilon_m(t,\hat{\eta}^m_{ARX})\}_{m=1,\cdots|\mathcal{LOC}_j|}$.

9.4.2 Structure selection

For estimating the local topology $(\mathcal{T}_{\tilde{G}}, \mathcal{T}_{\tilde{H}})$, we re-write (9.2) as,

$$w_{\mathcal{C}}(t) = \tilde{G}w_{\mathcal{C}}(t) + \bar{H}\xi_{\mathcal{C}}(t) + \xi_{\mathcal{C}}(t) \tag{9.8}$$

with $\bar{H} = (\tilde{H} - \mathcal{I})$ consisting of strictly proper modules. Motivated by the fact that $\hat{\xi}_{\mathcal{C}}(t) \to \xi_{\mathcal{C}}(t)$ w.p. 1 due to consistent estimation in the previous step, we now replace the term $\bar{H}\xi_{\mathcal{C}}(t)$ by $\bar{H}\hat{\xi}_{\mathcal{C}}(t)$, so that we can use $(w_{\mathcal{C}}, \hat{\xi}_{\mathcal{C}})$ as measured inputs in our topology estimation step.

Similar to the previous step, parameterizing the coefficients of the impulse response of the modules in $\tilde{G}$ and $\bar{H}$ at each m^{th} row of (9.8) and representing the parameter vector as η^m_{FIR}, the parameter vector of modules in each row can be estimated independently in parallel using the prediction error method [77] with one-step ahead predictor $\hat{w}_m(t|t-1;\eta^m_{FIR}) = \bar{\varphi}^\top_m(t)\eta^m_{FIR}$ where $\bar{\varphi}^\top_m$ is the data matrix. The parameter vector can be consistently estimated [78] by minimizing the sum of squared prediction errors $\varepsilon_m(t,\eta^m_{ARX}) = w_m(t) - \hat{w}_m(t|t-1;\eta^m_{ARX}) = w_m(t) - \bar{\varphi}^\top_m(t)\eta^m_{FIR}$. This will result in a closed form solution,

$$\hat{\eta}^m_{FIR} = \left[\frac{1}{N}\sum_{t=1}^{N}\bar{\varphi}_m(t)\bar{\varphi}^\top_m(t)\right]^{-1} \cdot \frac{1}{N}\sum_{t=1}^{N}\bar{\varphi}_m(t)w_m(t). \tag{9.9}$$

To identify the local topology $(\mathcal{T}_{\tilde{G}}, \mathcal{T}_{\tilde{H}})$, we need to define a criterion that operates on the basis of data and provides an indirect measure of how close a certain candidate topology structure is to the true topology structure of the network in (9.8). In this regard, we consider a modified representation of the *Akaike Information Criterion (AIC)* [77]

$$J_{AIC}(\hat{\eta}^m_{FIR}) = \log\left(\frac{1}{N}\sum_{t=1}^{N}\varepsilon^2_m(t,\hat{\eta}^m_{FIR})\right) + \frac{2N_G\ell_{FIR}}{N}. \tag{9.10}$$

Here, N_G is the total number of nodes that is used as input for predicting node w_m in the candidate structure, ℓ_{FIR} is the length of the impulse response of the modules and is chosen long enough to sufficiently represent the dynamics of the network.

Having defined the selection criteria, we estimate $\hat{\eta}^m_{FIR}$ in an iterative manner for different choices of predictor inputs. The estimate $\hat{\eta}^m_{FIR}$ obtained from (9.9) is used to compute $\varepsilon_m(t,\hat{\eta}^m_{FIR})$ which in turn is used to compute the cost of the selection criterion for this candidate structure. The ideal structure is the one that minimizes the selection criterion and to find it, a total $2^{|\mathcal{LOC}_j|}$ combinations need to be tested out. Testing these combinations become computationally infeasible as $|\mathcal{LOC}_j|$ increases. Developing search algorithms to reduce the number of combinations to test is a non-trivial problem. To reduce the number of combinations to test, we develop an iterative search algorithm (referred to as the

Focus search algorithm) that is greedy in its approach to identify the ideal structure.

Consider a node $w_m, m \in \mathcal{LOC}_j$, whose interconnection structure we wish to identify. In the absence of interconnection information, it is safe to assume that $\hat{\xi}_m$ is connected to the node w_m. Therefore, in the focus search algorithm, we initialize the structure with the innovation signal $\hat{\xi}_m$ and compute the cost of the selection criterion for this structure. With this computed cost as the base cost, we search for one nodal or innovation signal that minimizes the selection criterion. This nodal/innovation signal is then added to the structure and the corresponding cost becomes the new base cost. This step is repeated with the remaining nodal and innovation signals until the cost saturates. The interconnection information is obtained from the resulting structure. The focus search algorithm along with the remaining steps for identifying $(\mathcal{T}_{\tilde{G}}, \mathcal{T}_{\tilde{H}})$ are presented in Algorithm 4.

9.5 Identification algorithm

We now identify the target module which is one of the modules in $\tilde{G}$, based on the identified local topology $(\hat{\mathcal{T}}_{\tilde{G}}, \hat{\mathcal{T}}_{\tilde{H}})$. For this we resort to the already established direct method for local module identification [104]. As indicated in Section 9.2, this requires choosing the node signals in a predictor model (9.1) such that conditions on parallel paths and loops and on the absence of confounding variables are satisfied. An appropriate choice for a predictor model can be made based on the estimated topology $(\hat{\mathcal{T}}_{\tilde{G}}, \hat{\mathcal{T}}_{\tilde{H}})$, resulting in either a MISO or a MIMO predictor model, in which the target module to be identified is embedded.

Irrespective of the structure of the predictor model, to identify one target module, all the modules in the predictor model need to be suitably estimated. This step becomes computationally expensive as the number of modules in the predictor model increases. Also, parameterizing all the modules result in an explosion of nuisance parameters that affect the variance of the target module estimate. As a result, to circumvent model order selection and to reduce the number of nuisance parameters, we consider the regularized kernel based methods, *Empirical Bayes Direct Method* (EBDM) of [101] for MISO structures and *Empirical Bayes Local Direct Method* (EBLDM) of [99] for MIMO structures.

In both these methods, the problem of model order selection step is simplified by opting for a non-parametric impulse response modeling of all modules except the target module. To reduce the number of parameters, each impulse response is modelled as a Gaussian Process (GP) governed by a *Stable spline (SS) kernel* [21]. As a result, an impulse response of any length can be captured using only two hyperparameters that govern the SS kernel. In these methods, the parameters are identified using an Empirical Bayes approach [80] by maximizing the marginal likelihood of data which inherently minimizes the mean square error of the estimation problem. This further reduces the variance of estimated parameters of the target module.

Algorithm 4 Identification of local topology $(\mathcal{T}_{\tilde{G}}, \mathcal{T}_{\tilde{H}})$

```
Input:$\{w_k(t)\}_{t=1}^{N}, k \in \mathcal{LOC}_j$
Output: $\hat{\mathcal{T}}_{\tilde{G}}, \hat{\mathcal{T}}_{\tilde{H}}$
1. for $m \in \mathcal{LOC}_j$
    (a) Estimate the $\alpha_k$, $k \in \mathcal{LOC}_j$ by cross validation (refer to [21] for details).
    (b) Identify $\hat{\eta}^m_{ARX}$ using (9.6).
2. end for
3. Generate $\hat{\xi}_c(t) = \text{vec}\{\varepsilon_m(t, \hat{\eta}^m_{ARX})\}_{m=1,\cdots|\mathcal{LOC}_j|}$ using (9.5).
4. Choose an appropriate selection criteria.
5. for $m \in \mathcal{LOC}_j$
    (a) Initialize $\mathcal{W}_m = \varnothing$, and $\mathcal{H}_m = \{\hat{\xi}_m\}$.
    (b) Estimate $\hat{\eta}^m_{FIR}$ using (9.9).
    (c) Compute the $J_{base}$ using (9.10).
    (d) for $count = 1 : 2|\mathcal{LOC}_j|^2 - |\mathcal{LOC}_j|$
        i. for $k \in \mathcal{LOC}_j$
            A. Set $\mathcal{W}_m = \{w_k\}$.
            B. Estimate $\hat{\eta}^m_{FIR}$ using (9.9).
            C. Compute $J_{iteration}$ using (9.10).
            D. if $J_{iteration} < J_{base}$
            E. Set $J_{base} = J_{iteration}$.
            F. Set $\mathcal{S}^{temp}_m = w_k$
            G. else
            H. continue
        ii. for $k \in \mathcal{LOC}_j \backslash m$
            A. Set $\mathcal{H}_m = \{\hat{\xi}_k\}$.
            B. Estimate $\hat{\eta}^m_{FIR}$ using (9.9).
            C. Compute $J_{iteration}$ using (9.10).
            D. if $J_{iteration} < J_{base}$
            E. Set $J_{base} = J_{iteration}$.
            F. Set $\mathcal{S}^{temp}_m = \hat{\xi}_k$
            G. else continue
            H. end if
        iii. end for
        iv. Add $\mathcal{S}^{temp}_m$ to $\mathcal{W}_m$ or $\mathcal{H}_m$ accordingly.
    (e) end for
    (f) Set the non-zero entries for $m^{\text{th}}$ row of $\hat{\mathcal{T}}_{\tilde{G}}, \hat{\mathcal{T}}_{\tilde{H}}$ based on $\mathcal{W}_m$ and $\mathcal{H}_m$.
6. end for
```

9.6 Numerical simulation

9.6.1 Toplogy estimation

To highlight the effectiveness of the topology estimation algorithm, we generate 50 random stable networks (modules of 2^{nd} order are randomly generated) with the network topology as shown in Figure 9.2. In this 10-node network, the process noise (v_1, v_8), (v_4, v_5), and (v_7, v_9) are pairwise correlated, while e is a Gaussian white noise process with $cov(e) = I$.

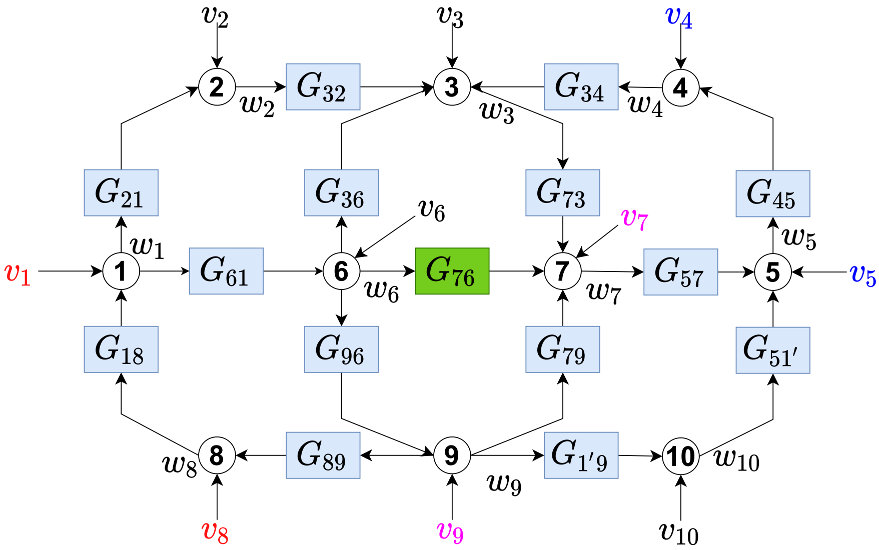


Figure 9.2: 10-Node dynamic network with target module G_{76}. The noise processes (v_1, v_8), (v_4, v_5), and (v_7, v_9) are correlated.

We first identify the locality of the node 7, and immerse the remaining nodes. The Wiener filter length F is commonly chosen to be 20 across all the different networks; the threshold value ρ is chosen after visually examining the obtained Wiener filters. Following this, we identify the local topology with $\ell_{ARX} = 50$. The behaviour of this algorithm is studied by comparing the *True Positive Rate (TPR)* and the *False Positive Rate (FPR)* over different choices of ℓ_{FIR}. The expressions for TPR and FPR are as follows

$$\text{TPR} = \frac{\text{TP}}{\text{P}}, \quad \text{FPR} = \frac{\text{FP}}{\text{A}}.$$

Here, P refers to the number of present interconnections in the network, TP refers to the number of instances of present interconnection identified by the algorithm, A refers to the number of absent interconnections in the network, and FP refers to the number of instances of absent interconnections falsely identified by the algorithm.

The obtained TPR vs FPR for $N = 1000$ over different $\ell_{FIR} = (10, 20, 40, 80)$ is shown in Figure 9.3. The quiver represent the direction of increasing ℓ_{FIR}. For a comparison with results for different selection criteria, the reader is referred to [98]. It can be seen from this Figure that as ℓ_{FIR} increases, the curves get closer to $(1, 0)$ until $\ell_{FIR} = 40$. Beyond $\ell_{FIR} = 40$, the curves do not get any closer to $(0, 1)$. This saturation is observed because the average impulse response length of the true network is approximately 40. Beyond this value, it can be observed that the performance of the algorithm is uniform in identifying the nodal and noise topology.

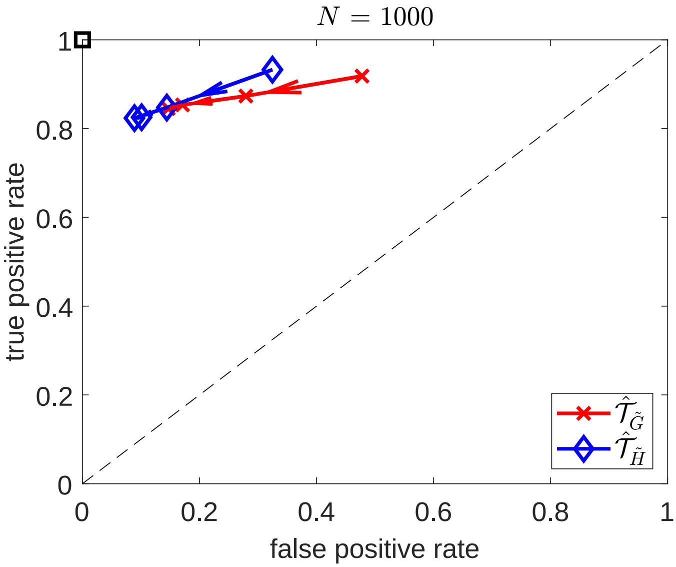


Figure 9.3: True Positive Rate and False Positive Rate for nodal topology ($\hat{\mathcal{T}}_{\tilde{G}}$) and noise topology ($\hat{\mathcal{T}}_{\tilde{H}}$) using AIC as selection criterion, over different $\ell_{FIR} = (10, 20, 40, 80)$ for $N = 1000$. The quiver represents the direction of increasing ℓ_{FIR}.

9.6.2 Target module identification

To highlight the effectiveness of the estimation algorithm, we consider one of the 50 randomly generated networks for evaluating the performance of the complete algorithm from a parameter identification perspective. To prevent issues that might arise due to lack of sufficient excitation for target module identification, we add an additional white noise source with unit power to each node in the network. We run 50 independent Monte Carlo (MC) experiments for different realizations of the noise source $e(t)$ while keeping the dynamics of the network fixed. We collect $N = 1000$ samples of all the nodes for each MC simulation. The network dynamics for the MC simulations can be found in [98]. The target module is given by,

$$G_{76} = \frac{0.1050q^{-1} - 0.3465q^{-2}}{1 + 0.0480q^{-1} - 0.2534q^{-2}} = \frac{b_1 q^{-1} - b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}}. \tag{9.11}$$

For the first step of finding the locality of w_7, the non-causal Wiener filter length is chosen to be 20 while threshold ρ for identifying the locality is chosen to be 0.18. The true locality for this network is $\mathcal{B}_7 = \{3, 4, 5, 6, 9, 10\}$ and only this set of nodes should be non-causally Wiener correlated (i.e. $\|W_{7\star}\|_\infty > \rho$) to the node 7. The mean and standard deviation plot of the non-causal Wiener filter over 50 MC simulations shown in Figure 9.4 ensures the sparsity conditions derived in Theorem 9.1.

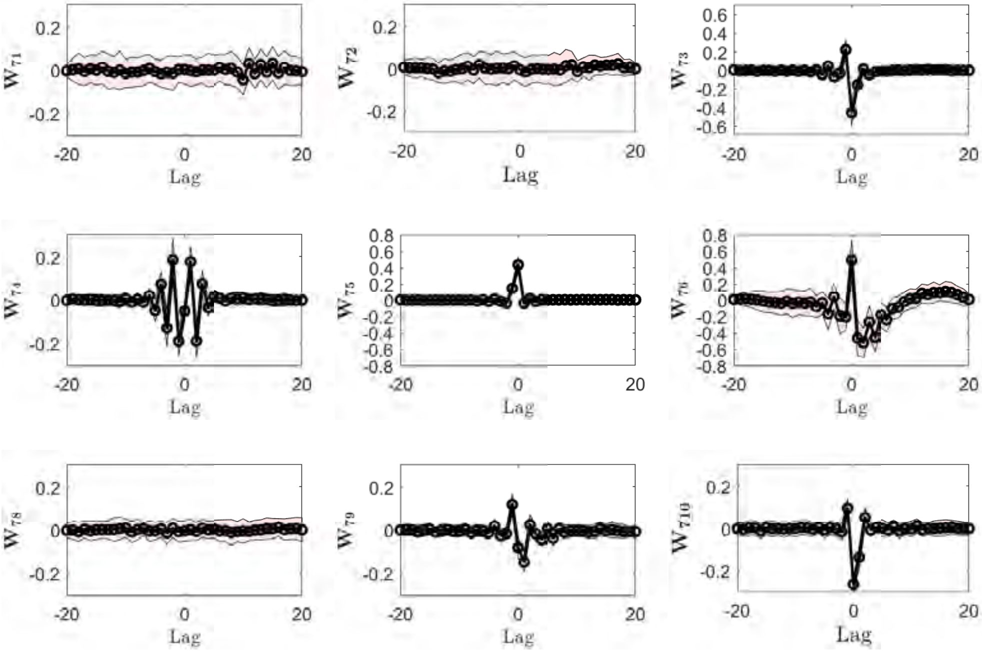


Figure 9.4: Non-causal Wiener filter estimate $\hat{W}_7$ obtained by projecting w_7 on the remaining nodes in the network. The black line represents the mean of the estimate over 50 MC simulations. The red shaded region represents the standard deviation of the estimates over 50 MC simulations.

Next, the local topology is identified by choosing $\ell_{ARX} = 50$ and $\ell_{FIR} = 40$ for the lengths of impulse response and the AIC selection criterion. The results show a TPR of 0.8417 and FPR of 0.1175.

To evaluate the performance of the identified topology in estimating the target module G_{76}, we compare the following local module identification strategies:

1. **EBLDM+TT:** This is the method developed in [99]. We use the *Full input case* algorithm of local direct method [104] for predictor model selection considering the true local topology ('TT'). This will lead to a MIMO predictor model $\{w_6, w_3, w_9\} \rightarrow \{w_7, w_9\}$. We use this estimator as an upper bound of the performance to identify the target module.;

2. **EBLDM+IT:** This is the same estimator as the previous one, with the difference that we use the identified local topology ('IT') for obtaining the MIMO predictor model;

3. **DM+TT:** This is the standard direct method introduced in [124] for systems with uncorrelated disturbances, with the output node of the target module as output and its parents as inputs. This will lead to a MISO predictor model $\{w_6, w_3, w_9\} \rightarrow \{w_7\}$ and we parameterize all modules;

4. **DM+IT:** This is the same estimator as the previous one, with the difference that the predictor model is formed with the identified topology;

5. **EBDM+TT+IN:** This is the method developed in [101, 102] for uncorrelated disturbances. However in order to deal with correlated disturbances in the network, a MISO noise model is estimated by using the innovations estimates $\hat{\xi}_{\mathcal{B}}$ as predictor input signals, and using the true topology of the network;

6. **EBDM+IT+IN:** This is same as the previous one but using the identified topology to form the predictor model.

The true model orders of all the modules in the network are assumed to be known. To evaluate the performance of the methods, we use the standard goodness-of-Fit metric,

$$\text{Fit} = 1 - \frac{\|g_{76} - \hat{g}_{76}\|_2}{\|g_{76} - \bar{g}_{76}\|_2}$$

where, g_{76} is the true value of the impulse response of G_{76}, $\hat{g}_{76}$ is the impulse response of the estimated $\hat{G}_{76}$ and $\bar{g}_{76}$ is the sample mean of g_{76}. The box plot showing the fit of the impulse response of G_{76} for the different methods is shown in Figure 9.5. It can be observed that the fit of the impulse response for the methods in which the identified topology is used, is similar to those where the true topology is used, implying that the use of the identified topology essentially preserves the performance with respect to the fit of the impulse response. In Figure 9.5, we also see that the EBLDM and the EBDM have significantly better fit compared to the DM. The reduction in variance is attributed to the regularization approach followed in both the EBLDM and EBDM. It is observed that, incorporating the innovation estimate into the estimation problem has a positive effect for EBDM whose performance is comparable to that of EBLDM except for a marginally higher estimation bias. Therefore, the developed approach for no prior topology information has an on par performance compared to the approaches that use the known true topology.

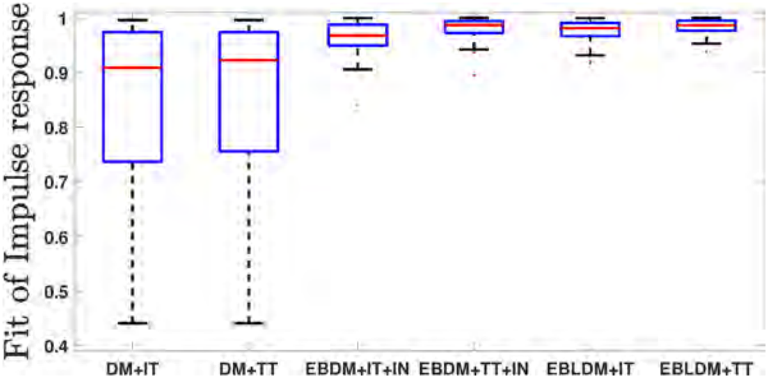


Figure 9.5: Box plot of the fit of the impulse response of $\hat{G}_{76}$ obtained over 50 MC simulations for different identification methods

Remark 9.1 *The example considered has a network of moderate size. However, when the size of the network increases, the locality of the output node need not necessarily expand. Additionally, even if the locality of the output node expands, the local topology estimation run in parallel due to its MISO formulation and the estimation steps have analytical solutions requiring no optimization. As a result, the developed algorithm will be less complex, faster and scalable to larger networks.*

Remark 9.2 *The developed approach can also be used when the disturbance topology is known or fixed. In this case, we can eliminate step 1 since we can find the locality, and fix the noise topology in step 2 and follow the same approach.*

Remark 9.3 *Step 2 acts as a stand-alone topology estimation procedure for the entire network when the set $\mathcal{B}_j$ is chosen to consist of all nodes in the network.*

9.7 Conclusions

A novel approach for identifying a module embedded in a dynamic network effectively and efficiently without any prior topology information has been developed. The approach incorporates the estimation of the required prior knowledge on the network and noise topology into the identification framework, leading to a unified and complete data-driven approach for local module identification. The local network and noise topology is estimated through non-causal Wiener filters and a multi-step least squares algorithm, thereby requiring only a series of analytical solutions with parallel computation capabilities and thus scalable to large-scale networks. Based on the estimated topology, the target module is identified using regularized kernel-based methods that reduces the complexity by circumventing the model order selection step and reduces the number of parameters to be estimated which has direct influence on the variance. Numerical simulations performed on fifty 10-node network examples with correlated process disturbances shows promising results for the network and noise topology estimation, and the target module estimated with the identified topology has on par performance compared the approaches that uses the known true topology.

9.8 Related videos

Learning local modules in dynamic networks without prior topology information

Venkatakrishnan Rajagopal, Karthik Ramaswamy, Paul Van den Hof

60th IEEE Conference on Decision and Control, December 13-17, 2021, Austin, Texas, USA

Learning local modules without prior topology information

Appendices

9.A Proof of Theorem 9.1

The proof is formulated along the same lines as the proof for the situation of uncorrelated disturbances in [85]. First we formulate a Lemma that formalizes the relationship between entries of the non-causal Wiener filter and the inverse of the spectral density matrix Φ_w.

Lemma 9.1 *In the situation of Theorem 9.1, $W_{ji}(z) = W_{ij}(z) = 0$ if and only if the entries (i, j) and (j, i) of $\Phi_w^{-1}(z)$ are zero.*

Proof: Without loss of generality, consider $j = L$ and define $\bar{w} := (w_1, \dots w_{L-1})^\top$ and $\bar{v} := (v_1, \dots v_{L-1})^\top$. Since in the considered dynamic network, the process noise may be correlated, we consider the following decomposition of the process noise v_L.

$$v_L = (v_L)^{\perp\bar{w}} + (v_L)^{\parallel\bar{w}}, \tag{9.12}$$

where projection and orthogonal complement is considered in the shift-invariant vector space equipped with the inner product $< x, y >= \mathbb{E}(xy^*)$. For the non-causal Wiener filter $W(z)$ that estimates w_L from $\bar{w}$, we can write

$$w_L = W(z)\bar{w} + (v_L)^{\perp\bar{w}} \tag{9.13}$$

where $(v_L)^{\perp\bar{w}}$ is uncorrelated to $\bar{w}$ because of the orthogonal projection properties of the Wiener filter. Also, $(v_L)^{\perp\bar{w}}$ is non-zero since the dynamic network has a full rank process noise. Define $r = (\bar{w}^\top \quad (v_L)^{\perp\bar{w}})^\top$. We observe that,

$$r = \begin{pmatrix} \mathcal{I} & 0 \\ -W(z) & 1 \end{pmatrix} w, \text{ and } \quad w = \begin{pmatrix} \mathcal{I} & 0 \\ W(z) & 1 \end{pmatrix} r.$$

Following this,

$$\begin{aligned}
\Phi_w^{-1} &= \begin{pmatrix} \mathcal{I} & -W(z)^* \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \Phi_{\bar{w}}^{-1} & 0 \\ 0 & \Phi^{-1}_{(v_L)^{\perp\bar{v}}} \end{pmatrix} \begin{pmatrix} \mathcal{I} & 0 \\ -W(z) & 1 \end{pmatrix} \\
&= \begin{pmatrix} \Phi_{\bar{w}}^{-1} + W(z)^* W(z) \Phi^{-1}_{(v_L)^{\perp\bar{v}}} & -W(z)^* \Phi^{-1}_{(v_L)^{\perp\bar{v}}} \\ -W(z) \Phi^{-1}_{(v_L)^{\perp\bar{v}}} & \Phi^{-1}_{(v_L)^{\perp\bar{v}}} \end{pmatrix}.
\end{aligned}$$

Since the term $\Phi^{-1}_{(v_L)^{\perp \bar{v}}}$ is scalar, it follows that the elements (i, L) and (L, i) of this matrix are zero if and only if $W_{iL}(z)$, respectively $W_{L,i}(z)$ are zero. ■

For proving Theorem 9.1 we now need to relate the sparsity of Φ_w^{-1} to the locality of w_j. To this end we consider the dynamic network and rewrite it as $w(t) = (\mathcal{I} - G(q))^{-1}v(t)$. Then Φ_w^{-1} can be written as

$$\Phi_w^{-1} = (\mathcal{I} - G)^*\Phi_v^{-1}(\mathcal{I} - G) = \Phi_v^{-1} - G^*\Phi_v^{-1} - \Phi_v^{-1}G + G^*\Phi_v^{-1}G$$

and the element (j, i) of (Φ_w^{-1}):

$$(\Phi_w^{-1})_{ji} \;=\; (\Phi_v^{-1})_{ji} \,-\, (G_{*j})^*(\Phi_v^{-1})_{*i} \,-\, (\Phi_v^{-1})_{j*}G_{*i} \,+\, (G_{*j})^*\Phi_v^{-1}G_{*i}. \quad (9.14)$$

The first term in (9.14) is zero if the process noise of nodes w_i and w_j are uncorrelated. The second and third terms in the expression are zero if no children of w_i have process noise correlated with w_j and vice versa. The last expression is zero if the nodes w_i and w_j have no common children or if the process noise of the children of the nodes w_i and w_j are uncorrelated. Therefore, if $(\Phi_w^{-1})_{ji}$ is non-zero, then node w_i is in the locality $\mathcal{LOC}_j$. Using the result of Lemma 9.1, we get that $W_{ji}(z) \neq 0$ implies $w_i \in \mathcal{LOC}_j$.

9.B Proof of Proposition 9.1

Starting from (9.2), we represent the network as follows:

$$\breve{H}w_{\mathcal{C}} = \breve{G}w_{\mathcal{C}} + \xi_{\mathcal{C}}, \quad (9.15)$$

where, $\breve{H} = \tilde{H}^{-1}$ and $\breve{G} = \tilde{H}^{-1}\tilde{G}$. Due to the presence of off-diagonal terms in $\tilde{H}$, $\breve{H}$ has off-diagonal terms and $\breve{G}$ has diagonal terms. As a result, we group the diagonal and off-diagonal terms of $\breve{H}$ and $\breve{G}$ into separate transfer function matrices as follows:

$$\begin{aligned} \breve{H} &= \breve{H}_D + \breve{H}_{ND}, \\ \breve{G} &= \breve{G}_D + \breve{G}_{ND}, \end{aligned} \quad (9.16)$$

where, the subscripts $(\cdot)_D$ and $(\cdot)_{ND}$ represent diagonal and non-diagonal respectively. Note that $\breve{H}_{ND}$ is strictly proper since $\breve{H}$ is monic. Substituting (9.16) in (9.15) then results in

$$(\breve{H}_D - \breve{G}_D)w_{\mathcal{B}} = (\breve{G}_{ND} - \breve{H}_{ND})w_{\mathcal{B}} + \xi_{\mathcal{B}} \quad (9.17)$$

which proves the result of the Proposition with $Q = (\breve{H}_D - \breve{G}_D)$ and $P = (\breve{H}_{ND} - \breve{G}_{ND})$.

9.C Computation of non-causal Wiener filter [122]

The Wiener filter determines the optimal projection of a signal $w_j(t)$ in the space $\mathcal{W}_j = \text{span}\{w_i(t+p) : p \in \mathbb{Z}\}_{i \neq j}$ Here we compute the Wiener filter by approximating it with a finite impulse response (FIR) filter, also known as FIR Wiener filter. Let the order of the FIR Wiener filter be F. Here the optimal estimate $\hat{w}_j(t)$ is written as,

$$\hat{w}_j(t) = \sum_{\substack{k=1 \\ k \neq j}}^{L} \sum_{p=-F}^{F} h_{k,p} w_k(t+p) \tag{9.18}$$

The Wiener filtering orthogonality condition is used to determine the constants $h_{k,p}$ in $\hat{W}_j(z)$. According to the orthogonality condition,

$$\mathbb{E}\left[\hat{w}_j(t) w_i(t+p)\right] = \mathbb{E}\left[w_j(t) w_i(t+p)\right] \tag{9.19}$$

where, $i \in \{1, \cdots, j-1, j+1, \cdots, L\}, p \in \{-F, -F+1, \cdots, F-1, F\}$. Combining (9.18) and (9.19),

$$\begin{aligned}[R_{w_1 w_i}(-F-p) \cdots R_{w_1 w_i}(F-p) \cdots R_{w_L w_i}(-F-p) \cdots \\ R_{w_N w_i}(F-p)]\, h = R_{w_j w_i}(-p),\end{aligned} \tag{9.20}$$

where, $i \in \{1, \cdots, L\} \backslash j, p \in \{-F, \cdots, F\}$,

$$\begin{aligned} h &:= \left[h_1^\top h_2^\top \cdots h_{j-1}^\top h_{j+1}^\top \cdots h_N^\top\right]^\top, \text{and} \\ h_i^\top &:= \left[h_{i,-F} \cdots h_{i,-1} h_{i,0} h_{i,1} \cdots h_{i,F}\right] \end{aligned} \tag{9.21}$$

The set of equations in (9.20) and (9.21) describe $(2F+1)(L-1)$ linear equations in $(2F+1)(L-1)$ unknowns in the vector h Thus, in combined form the equations become,

$$Rh = S$$

Thus, $h = R^{-1} S$ is used to compute the coefficients of the Wiener filters. Note that the matrix R and the vector S can be computed using the data $\{w_k\}_{t=1}^{N}, k \in \{1, \cdots, L\}$.

10 Chapter

An Empirical Bayes method for handling missing node observations

In order to identify a module embedded in a dynamic network, one has to formulate a multiple-input estimation problem that necessitates certain nodes to be measured and included as predictor inputs. However, some of these nodes may not be measurable in many practical cases due to sensor selection and placement issues. This may lead to non-satisfaction of the target module invariance conditions discussed in Chapters 3 and 4, resulting in biased estimates of the target module. Furthermore, the identification problem associated with the multiple-input structure may require determining a large number of parameters that are not of particular interest to the experimenter, with increased computational complexity in large-sized networks. In this chapter, we tackle these problems by using the data augmentation strategy that allows us to reconstruct the missing node measurements and increase the accuracy of the estimated target module. To this end, we develop a system identification method using regularized kernel-based methods coupled with approximate inference methods. Keeping a parametric model for the module of interest, we model the other modules as Gaussian Processes (GP) with a kernel given by the so-called stable spline kernel. An Empirical Bayes (EB) approach is used to estimate the parameters of the target module. The related optimization problem is solved using an Expectation-Maximization (EM) method, where we employ a Markov-chain Monte Carlo (MCMC) technique to reconstruct the unknown missing node information and the network dynamics. Numerical simulations on dynamic network examples illustrate the potentials of the developed method.

10.1 Introduction

When using the direct method for single module identification, the fundamental *parallel path/loop* condition needs to hold in order to achieve consistent estimates of the target module. This condition states that all parallel paths from the input to the output of the target module and all the loops around the output of the target module must pass through nodes that are measured and included as predictor inputs in the estimation problem. Therefore, it becomes quintessential to measure certain nodes to satisfy the *parallel path/loop* condition; however, measurement of such nodes might not always be possible. Therefore, to mitigate this issue and achieve reduced bias estimates, it becomes essential to develop identification methods to cope with networks with missing nodes/observations.

In this chapter, we introduce a novel identification method that handles the situation of non-measured inputs (i.e., missing node observations) that are required to obtain unbiased target module estimates. To handle the situation of missing node observations, we use a *data augmentation* strategy [123, 132], where the missing node observations are also estimated along with the parameters of the modules. For reconstructing the missing node information, we use the available information from nodes that lie both upstream and downstream compared to the missing node. To avoid model order selection and reduce the number of nuisance parameters to be estimated, we build on [101, 102] (Chapter 7 and Chapter 8) and employ non-parametric regularized kernel-based methods. We keep a parametric model only for the module of interest in order to have an accurate description of its dynamics. The rest of the modules are modeled as zero-mean Gaussian processes, with covariance matrix (kernel) given by a *first-order stable spline kernel* [21, 93], which encodes stability and smoothness of the processes. By this way of modeling, we reduce the number of estimated parameters and obtain a significant reduction in the variance of the estimates [93].

Using the above approach, we obtain a Gaussian probabilistic description that depends on a vector of parameters that also contains the parameters of interest. We use an Empirical Bayes approach to estimate such a vector; this amounts to maximizing the marginal likelihood of the observed data, obtained by integrating out the dependence on the missing node data and the impulse response of the modules. The solution to the maximization problem is obtained using an iterative scheme based on the Expectation-Maximization (EM) [35] algorithm. The E-step characterizing this scheme involves computing the expected value of a joint log-likelihood. Since in this problem the associated integral does not admit an analytical solution, we employ a Monte Carlo approximation method where samples of the target probability distributions are generated using an instance of the *Gibbs sampler* [52]. As for the M-step of the EM procedure, we show that it can be split into several small optimization problems that are simple to solve, making the whole optimization routine computationally cheap. Numerical simulation on a dynamic network with missing node observations shows the developed method's potentials compared to the available classical methods.

This chapter is organized as follows. In Section 10.2, the setup of the dynamic network and the problem statement is defined. Section 10.3 briefs about the standard direct method. Next, we provide the MIMO model, strategy to reduce the parameters of nuisance modules and solve the missing node observation problem, and solution to the marginal likelihood problem using the MCEM method in Section 10.4, 10.5, and 10.6. Next, numerical simulations and results are provided in Section 10.7, followed by discussions and conclusions. Finally, the technical proofs are provided in the appendix.

10.2 Problem setup

We consider the dynamic network setup in (2.2) with known network topology. We consider that the process noises are uncorrelated i.e refers to the situation that $\Phi_v(\omega)$ and H^0 are diagonal. We assume e to be a Gaussian white noise process. We assume that we have collected N measurements of the certain internal variables $\{w_k(t)\}_{t=1}^N$, and that we are interested in building a model of the module directly linking node i to node j, that is $G_{ji}^0(q)$, using the measurements of the available internal variables, and possibly u. To this end, we choose a parameterization of $G_{ji}^0(q)$, denoted as $G_{ji}(q,\theta)$, that describes the dynamics of the module of interest for a certain parameter vector $\theta = \theta_0 \in \mathbb{R}^{n_\theta}$.

10.3 The direct method and predictor input selection

As discussed in Section 7.3, not only the target module is parameterized in the standard MISO direct method. The prediction error also depends on the additional parameters entering the remaining modules $G_{jk}^0(q), k \in \mathcal{N}_j \backslash \{i\}$ and the noise model $H_j^0(q)$, which need to be estimated to guarantee consistent estimates of θ. Therefore, the total number of parameters may grow large if the cardinality of $\mathcal{N}_j$ is large, with a detrimental effect on the variance of the estimate of θ in the case where N is not very large.

According to [29], it is sufficient to have a set of node signals $\mathcal{D}_j$ to be measured and used as predictor inputs in the direct method, that satisfies an additional parallel path/loop condition (condition 2.1) and a confounding variable condition. However, if $\mathcal{Z}_j \subseteq \mathcal{D}_j$ represent the node signals that are not measured or are inaccessible (i.e. *missing node observations*) but required to satisfy the above condition, then identification through the direct method using the available signals leads to biased target module estimates [29]. From this, we note that the direct method requires the measurement of the node signals $w_k, k \in \mathcal{D}_j$ (see Chapter 3). For example, consider the network in Figure 10.1 with diagonal noise spectrum and $u_2 = r_2, u_4 = r_4$. In order to identify G_{31}^0 using the direct method, performing a MISO identification with w_3 as output and $w_{\mathcal{D}_j} = \{w_1, w_4\}$ as inputs when w_2 is not measured (i.e. missing node observation), leads to estimation of modules in an immersed network (a network with w_2 removed) as in Figure 10.2. As we can see, we now estimate $G_{31}^0 + G_{32}^0 G_{21}^0$ (from w_1 to w_3) and not the

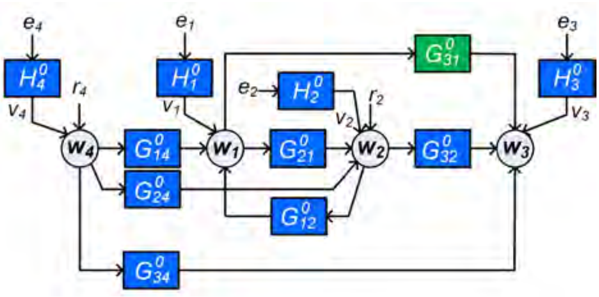


Figure 10.1: Network example with 4 internal nodes, 2 external excitation signals and a noise sources at each node. Target module is G_{31}.

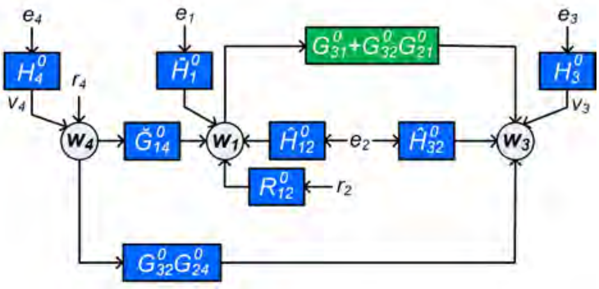


Figure 10.2: Immersed network of network in Figure 10.1 where the non-measured node w_2 is removed.

desired target module G_{31}^0, which leads to a biased estimate. Confounding variables like e_2 when w_2 is non-measured also create bias in the estimate of the target module [104].

In the sequel of this chapter, we will explain how we deal with the problem of missing node observations by re-constructing these measurements.

10.4 Concepts and Notations

The concepts and notations are the same as presented in Section 4.4. In addition, we denote $\bar{\mathcal{Y}}^{(j)} = \mathcal{Y}\backslash\{j\}$ and $\mathcal{W} = \mathcal{Y} \cup \mathcal{D}$. Let $w_\ell, \ell \in \mathcal{D}_j^w \subseteq \mathcal{W}\backslash\{j\}$ denote the node signals in w that have unmeasured paths[1] to w_j and $u_\ell, \ell \in \mathcal{D}_j^u$ denote the non-zero excitation signals in u that have unmeasured paths to w_j. Also, for all $k \in \bar{\mathcal{Y}}^{(j)}$, let $\mathcal{D}_k^w = \mathcal{W}\backslash\{k\}$ and let $u_\ell, \ell \in \mathcal{D}_k^u \subseteq \mathcal{W} \cup \mathcal{Z}$ denote the non-zero excitation signals in u.

[1] An unmeasured path is a path that does not pass through a node $w_\ell, \ell \in \mathcal{W}$. Analogously, we can define unmeasured loops through a node w_k.

10.5 An Empirical Bayes identification technique with missing node observations

10.5.1 Introduction

Following the result of Chapter 4, we have a MIMO representation of (part of) our network based on the chosen set of signals $(w_{\mathcal{Q}}, w_{\mathcal{O}}, w_{\mathcal{U}})$. It is known that there exists a representation of the network as in (4.2) in which there are no confounding variables for the estimation problem $w_{\mathcal{U}} \to w_{\mathcal{Y}}$. From the result of Proposition 4.2, it is known that the system's equations for the output variables $w_{\mathcal{Y}}$ can be written as,

$$w_{\mathcal{Y}} = \bar{G} w_{\mathcal{D}} + \bar{H} \xi_{\mathcal{Y}} + \bar{R} u \qquad (10.1)$$

with $\xi_{\mathcal{Y}}$ a Gaussian white noise process, while $\bar{H}$ is monic, stable and stably invertible.

For estimation purposes we are going to use a specific form of (10.1), as formulated next, for which we need two additional conditions.

Condition 10.1 *There are no confounding variables for the estimation problem* $w_j \to w_{\mathcal{W}\setminus\{j\}}$.

Condition 10.2 *All paths from* $w_h, h \in \mathcal{O}\setminus\{j\}$ *to* w_j *pass through a node in* $w_{\mathcal{W}}$.

Proposition 10.1 *Consider the network as represented by* (2.2) *where the set of all nodes* $w_{\mathcal{L}}$ *is decomposed into disjoint sets* $w_{\mathcal{Q}}$, $w_{\mathcal{O}}$, $w_{\mathcal{U}}$ *and* $w_{\mathcal{Z}}$. *If Conditions 10.1 and 10.2 are satisfied, then there exists a form* (10.1) *with* $\bar{H}$ *being (block) lower triangular as follows,*

$$\underbrace{\begin{bmatrix} w_j \\ w_{\bar{\mathcal{Y}}(j)} \end{bmatrix}}_{w_{\mathcal{Y}}} = \underbrace{\begin{bmatrix} \bar{G}_{j\mathcal{D}} \\ \bar{G}_{\bar{\mathcal{Y}}(j)\mathcal{D}} \end{bmatrix}}_{\bar{G}} w_{\mathcal{D}} + \underbrace{\begin{bmatrix} \bar{H}_{jj} & 0 \\ \bar{H}_{\bar{\mathcal{Y}}(j)j} & \bar{H}_{\bar{\mathcal{Y}}(j)\bar{\mathcal{Y}}(j)} \end{bmatrix}}_{\bar{H}} \underbrace{\begin{bmatrix} \xi_j \\ \xi_{\bar{\mathcal{Y}}(j)} \end{bmatrix}}_{\xi_{\mathcal{Y}}} + \bar{R} u. \qquad (10.2)$$

Proof: Collected in the appendix. ■

Condition 10.3 *Let* G_{ji} *be the target network module to be identified. In the network (2.2):*

- *Every path from* w_i *to* w_j, *excluding the path through* G_{ji}, *passes through a node* $w_k, k \in \mathcal{W}$, *and*

- *Every loop through w_j passes through a node in $w_k, k \in \mathcal{W}$.* □

In order to guarantee that $\bar{G}_{ji}(q) = G_{ji}(q)$, i.e the target module appearing in equation (10.2) is the target module of the original network (2.2) *(invariance of target module)*, we utilize the following result:

Proposition 10.2 *Consider the situation of Proposition 10.1. If condition 10.3 is also satisfied, then we have $\bar{G}_{ji}(q) = G_{ji}(q)$.*

Proof: Collected in the appendix. ■

Based on the above results, we can now re-write (10.1) in the predictor form as given below.

Proposition 10.3 *Consider the network as represented by* (2.2) *where the set of all nodes $w_{\mathcal{L}}$ is decomposed into disjoint sets $w_{\mathcal{Q}}$, $w_{\mathcal{O}}$, $w_{\mathcal{U}}$ and $w_{\mathcal{Z}}$ such that Conditions 10.1, 10.2 and 10.3 are satisfied. The network equation for the node w_j and $w_k, k \in \bar{\mathcal{Y}}^{(j)}$ can be written as*[a]*,*

$$\begin{aligned} w_j(t) =&\hat{w}_j(t|t-1) + \xi_j(t) = S_j(q)(w_j(t) - u_j(t)) + (1 - S_j(q))G_{ji}(q)w_i(t) \\ &+ \sum_{\ell \in \mathcal{D}_j^w \backslash \{i\}} S_\ell(q)w_\ell(t) + u_j(t) + \sum_{\ell \in \mathcal{D}_j^u \backslash \{j\}} S_{j\ell}(q)u_\ell(t) + \xi_j(t), \end{aligned} \tag{10.3}$$

$$\begin{aligned} w_k(t) =&\hat{w}_k(t|t-1) + \xi_k(t) = B_k(q)(w_k(t) - u_k(t)) + \sum_{\ell \in \mathcal{D}_k^w} B_\ell(q)w_\ell(t) \\ &+ \sum_{\ell \in \mathcal{D}_k^u} B_{k\ell}(q)u_\ell(t) + u_k(t) + \xi_k(t), \end{aligned} \tag{10.4}$$

where we isolate the target module $G_{ji}(q)$. $S_\star(q), B_\star(q)$ are strictly proper predictor filters constructed from the terms in (10.2)*, and $\xi_{\mathcal{Y}}(t)$ is Gaussian white noise with $cov(\xi_{\mathcal{Y}}) = \bar{\Sigma}$.*

[a] from now on superscript ^{0} is dropped for convenience.

Proof: Collected in the appendix. ■

The lower (block) triangular structure of $\bar{H}$ in (10.2) allows us to isolate the target module G_{ji} in (10.3). Realizing the above representation requires conditions on the selection of node signals in $w_{\mathcal{Q}}, w_{\mathcal{O}}, w_{\mathcal{U}}$ (i.e. conditions 10.1, 10.2 and 10.3). The conditions are not a restriction. They can always be satisfied by appropriate selection of signals in the sets $\mathcal{Q}, \mathcal{O}$ and $\mathcal{U}$.

10.5.2 Signal selection

The identification method developed in this chapter relies on data augmentation strategies [123, 132]. Using this strategy, we treat the non-measured missing node signals as latent variables which are estimated along with the parameters. In particular, we build a MIMO predictor model where we

- include the non-measured missing node(s) in $w_{\mathcal{D}}$ and copy it to the output $w_{\mathcal{Y}}$,
- extend the input $w_{\mathcal{D}}$ with measured node signals that are ascendants of the missing node (upstream nodes), so that we can add information for reconstructing the missing node observation, and
- extend the output $w_{\mathcal{Y}}$ with measured node signals that are descendants of the missing node (downstream nodes), which might further help in improving the variance of the estimates.

Note that not all signals in $w_{\mathcal{D}} \cup w_{\mathcal{Y}}$ are measured node signals since they contain the non-measured missing node(s). However, for all considerations on confounding variables, the missing node is considered to be a measured node signal.

For simplicity, we are going to assume one missing node w_m (i.e. the cardinality of $\mathcal{Z}_j$ is one) and one additional node w_a as a descendant of the missing node w_m. However, the method described in this chapter can be extended to any number of missing nodes and additional nodes. As mentioned above, we are going to formulate a MIMO model structure where the outputs are (1) the output of the target module i.e. w_j and, (2) the missing node output (3) additional nodes that are descendants of the missing node. We select $\mathcal{Y} = \{w_j, w_m, w_a\}$. $w_{\mathcal{D}}$ consists of any set of measurable node signals, and has to contain the missing node signal(s) w_m. We include measured node signals that have unmeasured paths to $w_{\mathcal{Y}}$ in $w_{\mathcal{D}}$. The addition of missing node signal(s) in $w_{\mathcal{W}}$ ensures that condition 10.3 is satisfied. We choose signals in $w_{\mathcal{Y}}$ and $w_{\mathcal{D}}$ such that conditions 10.1 and 10.2 are satisfied. Since w_m and w_a are both in the set $\bar{\mathcal{Y}}^{(j)}$ they can be written according to (10.4) as,

$$w_m(t)=B_m(q)(w_m(t)-u_m(t))+\sum_{\ell\in\mathcal{D}_m^w} B_\ell(q)w_\ell(t)+\sum_{\ell\in\mathcal{D}_m^u} B_{m\ell}(q)u_\ell(t)+u_m(t)+\xi_m(t), \tag{10.5}$$

$$w_a(t)=F_a(q)(w_a(t)-u_a(t))+\sum_{\ell\in\mathcal{D}_a^w} F_\ell(q)w_\ell(t)+\sum_{\ell\in\mathcal{D}_a^u} F_{a\ell}(q)u_\ell(t)+u_a(t)+\xi_a(t), \tag{10.6}$$

with

$$cov(\begin{bmatrix}\xi_j(t)\\ \xi_a(t)\\ \xi_m(t)\end{bmatrix}) := \bar{\Sigma} = \begin{bmatrix}\sigma_j^2 & 0 & 0\\ 0 & \sigma_a^2 & \sigma_{am}^2\\ 0 & \sigma_{am}^2 & \sigma_m^2\end{bmatrix}$$

.

From the above, it is very clear that if we use the additional node w_a in the output, we need to model additional modules. This increase in complexity counterbalances the gain obtained by using more information. In this chapter, we develop an identification framework that uses the additional node(s); we also provide the framework that does not use additional nodes as a special case of the former.

10.5.3 Vector description of the dynamics

In this section, we obtain a vector description of the network dynamics for the available N measurements. For notation purposes, we introduce the N-dimensional vector g_{ji} (which will also depend on θ, although we will keep this dependence tacit) as the first N coefficients of the impulse response of $G_{ji}(q,\theta)$. Similarly, we define the vector s_k, $k \in \{\mathcal{D}_j^w\}\backslash i$, s_{jk}, $k \in \{\mathcal{D}_j^u\}\backslash j$ and s_j as the vectors containing the first l coefficients of the impulse responses of $S_k(q)$, $k \in \{\mathcal{D}_j^w\}\backslash i$, $S_{jk}(q)$, $k \in \{\mathcal{D}_j^u\}\backslash i$, and $S_j(q)$, respectively. Similarly, b_k, b_{mk}, b_m, f_k, f_{ak}, f_a are defined as the vectors containing the first l coefficients of the impulse responses of B_k, B_{mk}, B_m, F_k, F_{ak}, F_a respectively. The integer l is chosen large enough to ensure $s_k(l+1), s_{jk}(l+1), s_j(l+1), b_k(l+1), b_{mk}(l+1), b_m(l+1), f_k(l+1), f_{ak}(l+1), f_m(l+1) \simeq 0$.

Lemma 10.1 *Let the vector notation for the node $w_k(t)$ be $w_k := \begin{bmatrix} w_k(1) & \dots & w_k(N)\end{bmatrix}^T$ where $k \in \{j,m,a\}$. Considering the parameterization of $G_{ji}(q)$ (i.e. $G_{ji}(q,\theta)$), the network dynamics in* (10.3), (10.5) *and* (10.6) *can be represented in the vector form as:*

$$
\begin{aligned}
w_m &= \tilde{W}_m b_m + \sum_{k\in\mathcal{D}_m^w} W_k b_k + \sum_{k\in\mathcal{D}_m^u} R_k b_{mk} + u_m + \xi_m, && (10.7)\\
w_j &= \tilde{W}_j s_j + W_{ji} g_{ji} + \sum_{k\in\mathcal{D}_j^w\backslash\{i\}} W_k s_k + \sum_{k\in\mathcal{D}_j^u\backslash\{j\}} R_k s_{jk} + u_j + \xi_j, && (10.8)\\
w_a &= \tilde{W}_a f_a + \sum_{k\in\mathcal{D}_a^w} W_k f_k + \sum_{k\in\mathcal{D}_a^u} R_k f_{mk} + u_a + \xi_a, && (10.9)
\end{aligned}
$$

where ξ_j, ξ_m, ξ_a are the vectorized noise and r_j, r_m, r_a are the vectorized excitation signal. $\tilde{W}_j$, $\tilde{W}_m$, $\tilde{W}_a$, W_{ji}, W_k and R_k are Toeplitz matrices constructed from measurements of the respective node and excitation signals.

Proof: We denote by $W_k \in \mathbb{R}^{N\times l}$ the Toeplitz matrix of the vector $\overrightarrow{w}_k := \begin{bmatrix} 0 & w_k(1) & \dots & w_k(N-1)\end{bmatrix}^T$, $k \in \mathcal{D}_j^w \cup \mathcal{D}_m^w \cup \mathcal{D}_a^w \cup \mathcal{Y}$ and $W_{ji} \in \mathbb{R}^{N\times N}$ the Toeplitz matrix of the vector $\overrightarrow{w}_i := \begin{bmatrix} 0 & w_i(1) & \dots & w_i(N-1)\end{bmatrix}^T$. Let $R_\ell \in \mathbb{R}^{N\times l}$ be the Toeplitz matrix of the vector $\overrightarrow{u}_\ell := \begin{bmatrix} 0 & r_\ell(1) & \dots & r_\ell(N-1)\end{bmatrix}^T$ where $\ell \in \mathcal{Y} \cup \mathcal{D}_m^u \cup \mathcal{D}_a^u \cup \mathcal{D}_j^u$. Similarly, we denote by $\overleftrightarrow{W}_k \in \mathbb{R}^{N\times l}$ the Toeplitz matrix of the vector $\overleftrightarrow{w}_k := \begin{bmatrix} 0 & 0 & -w_k(1) & \dots & -w_k(N-2)\end{bmatrix}^T$, $k \in \{i,j\}$, and by G_θ the Toeplitz of g_{ji}. Considering the parameterization of G_{ji}^0 and the above established notations, we can rewrite the network dynamics in (10.3) as (10.8), (10.6) as (10.9), and (10.5) as (10.7) where $\tilde{W}_j := W_j - R_j + G_\theta \overleftrightarrow{W}_i$, $\tilde{W}_m := W_m - R_m$, $\tilde{W}_a := W_a - R_a$. ■

10.5.4 Strategy to reduce the number of parameters for nuisance modules

Before we move to an estimation scheme that can deal with the missing node w_m, we explain how we can avoid having to estimate a huge number of parameters in the nuisance modules, i.e. the modules that we need to identify but that are not the target module of interest. For this we follow the work in Chapter 7, where no missing nodes are assumed and extend it to the situation where there are missing node(s). Our goal is to limit the number of parameters necessary to describe w_j, w_a and w_m, in order to increase the accuracy of the estimated parameter vector of interest θ. Therefore, while we keep a parametric model for G_{ji}, for the remaining impulse responses in (10.7), (10.8) and (10.9), we use nonparametric models induced by Gaussian processes [107]. The choice of Gaussian processes is motivated by the fact that, with a suitable choice of the prior covariance matrix, we can get a significant reduction in the variance of the estimated impulse responses [93]. Therefore, we model $s_k, k \in j \cup \mathcal{D}_j^w \backslash \{i\}$, s_{jk}, $k \in \mathcal{D}_j^u$, $b_k, k \in m \cup \mathcal{D}_m^w$, b_{mk}, $k \in \mathcal{D}_m^u$, $f_k, k \in a \cup \mathcal{D}_a^w$, f_{ak}, $k \in \mathcal{D}_a^u$ as independent Gaussian processes (vectors in this case) with zero-mean. The covariance matrix of these vectors, usually referred to as a kernel in this context, is chosen to be corresponding to the so-called *First-order Stable Spline kernel*. The general structure of this kernel is given by

$$\lambda[K_\beta]_{x,y} = \lambda \beta^{\max(x,y)}\,, \tag{10.10}$$

where $\beta \in [0,1)$ is a *hyperparameter* that regulates the decay velocity of the realizations of the corresponding Gaussian vector, while $\lambda \geqslant 0$ tunes their amplitude. The choice of this kernel is motivated by the fact that it enforces favorable properties such as stability and smoothness in the estimated impulse responses [91], [92]. Therefore, we have that

$$s_k \sim \mathcal{N}(0, \lambda_k^s K_{\beta_k^s}) \quad , k \in j \cup \mathcal{D}_j^w \backslash \{i\}, \tag{10.11}$$

$$s_{jk} \sim \mathcal{N}(0, \lambda_{jk}^s K_{\beta_{jk}^s}) \quad , k \in \mathcal{D}_j^u \backslash \{j\}, \tag{10.12}$$

$$b_k \sim \mathcal{N}(0, \lambda_k^b K_{\beta_k^b}) \quad , k \in m \cup \mathcal{D}_m^w, \tag{10.13}$$

$$b_{mk} \sim \mathcal{N}(0, \lambda_{mk}^b K_{\beta_{mk}^b}) \quad , k \in \mathcal{D}_m^u, \tag{10.14}$$

$$f_k \sim \mathcal{N}(0, \lambda_k^f K_{\beta_k^f}) \quad , k \in a \cup \mathcal{D}_a^w, \tag{10.15}$$

$$f_{ak} \sim \mathcal{N}(0, \lambda_{ak}^f K_{\beta_{ak}^f}) \quad , k \in \mathcal{D}_a^u, \tag{10.16}$$

where we have assigned different hyperparameters to the impulse response priors to guarantee flexible enough models. In (10.7) - (10.9) we have terms that are multiplication of the Toeplitz matrix related to missing node w_m and the impulse response models, where both the missing node data and the prior hyperparameters of the impulse models are unknown and need to be estimated. In order to tackle the identifiability issues, we set the hyperparameters λ_m^s, λ_m^b and λ_m^f (i.e. λ's corresponding to the modules having missing node as inputs) to be 1.

10.5.5 Incorporating Empirical Bayes approach

We now explain how the parameters of the priors and the target module are estimated using the Empirical Bayes approach. For this, we define

$$s := \begin{bmatrix} s_j^\top & s_{c_1}^\top & \dots & s_{c_p}^\top & s_{jk_1}^\top & s_{jk_2}^\top & \dots & s_{jk_p}^\top \end{bmatrix}^\top, \tag{10.17}$$

where $c_1, \dots, c_p$ and $k_1, \dots, k_p$ are the elements of the set $\mathcal{D}_j^w \backslash \{i\}$ and $\mathcal{D}_j^u \backslash \{j\}$, and

$$\mathbf{W} := \begin{bmatrix} \tilde{W}_j & W_{c_1} & W_{c_2} & \dots & R_{k_{p-1}} & R_{k_p} \end{bmatrix}, \tag{10.18}$$

$$\mathbf{K}_1 := \mathrm{diag}\{\lambda_j^s K_{\beta_j^s}, \lambda_{c_1}^s K_{\beta_{c_1}^s}, \dots, \lambda_{jk_p}^s K_{\beta_{jk_p}^s}\}. \tag{10.19}$$

Analogously we define

$$b := \begin{bmatrix} b_m^\top & b_{c_1}^\top & \dots & b_{c_p}^\top & b_{mk_1}^\top & b_{mk_2}^\top & \dots & b_{mk_p}^\top \end{bmatrix}^\top, \tag{10.20}$$

$$\mathbf{R} := \begin{bmatrix} \tilde{W}_m & W_{c_1} & W_{c_2} & \dots & R_{k_{p-1}} & R_{k_p} \end{bmatrix}, \tag{10.21}$$

$$\mathbf{K}_2 := \mathrm{diag}\{K_{\beta_m^b}, \lambda_{c_1}^s K_{\beta_{c_1}^b}, \dots, \lambda_{mk_p}^b K_{\beta_{mk_p}^b}\}. \tag{10.22}$$

where $c_1, \dots, c_p$ and $k_1, \dots, k_p$ are the elements of the set $\mathcal{D}_m^w$ and $\mathcal{D}_m^u$ respectively, and

$$f := \begin{bmatrix} f_a^\top & f_{c_1}^\top & \dots & f_{c_p}^\top & f_{ak_1}^\top & f_{ak_2}^\top & \dots & f_{ak_p}^\top \end{bmatrix}^\top, \tag{10.23}$$

$$\mathbf{Q} := \begin{bmatrix} \tilde{W}_a & W_{c_1} & W_{c_2} & \dots & R_{k_{p-1}} & R_{k_p} \end{bmatrix}, \tag{10.24}$$

$$\mathbf{K}_3 := \mathrm{diag}\{\lambda_a^f K_{\beta_a^f}, \lambda_{c_1}^f K_{\beta_{c_1}^f}, \dots, \lambda_{ak_p}^f K_{\beta_{ak_p}^f}\}. \tag{10.25}$$

where $c_1, \dots, c_p$ and $k_1, \dots, k_p$ are the elements of the set $\mathcal{D}_a^w$ and $\mathcal{D}_a^u$ respectively. Using the above, we can rewrite (10.7), (10.8) and (10.9) in compact form and we obtain the following model:

$$\underbrace{\begin{bmatrix} w_j \\ w_a \\ w_m \end{bmatrix}}_{w_\mathcal{Y}} = \underbrace{\begin{bmatrix} \mathbf{W} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R} \end{bmatrix}}_{\mathbf{W}_\mathcal{D}} \underbrace{\begin{bmatrix} s \\ f \\ b \end{bmatrix}}_{g} + \underbrace{\begin{bmatrix} W_{ji} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}}_{\mathbf{W}_{ji}} g_{ji} + \underbrace{\begin{bmatrix} u_j \\ u_a \\ u_m \end{bmatrix}}_{u_\mathcal{Y}} + \underbrace{\begin{bmatrix} \xi_j \\ \xi_a \\ \xi_m \end{bmatrix}}_{\xi_\mathcal{Y}},$$

$$\begin{aligned} s &\sim \mathcal{N}(0, \mathbf{K}_1), \\ b &\sim \mathcal{N}(0, \mathbf{K}_2), \end{aligned} \tag{10.26}$$

$$
\begin{aligned}
f &\sim \mathcal{N}(0, \mathbf{K}_3)\,, \\
\xi_{\mathcal{Y}} &\sim \mathcal{N}(0, \Sigma)\,,
\end{aligned}
$$

where s, b, f and $\xi_{\mathcal{Y}}$ are mutually independent and with $\Sigma = \bar{\Sigma} \otimes I_N$. We note that the above model depends upon the vector of parameters

$$
\begin{aligned}
\eta := [\, \theta\, \lambda_j^s \,\ldots\, \lambda_{jk_p}^s\, \lambda_{c_1}^b \,\ldots\, \lambda_{mk_p}^b\, \lambda_a^f \,\ldots\, \lambda_{ak_p}^f \quad & \beta_j^s \,\ldots\, \beta_{jk_p}^s\, \beta_m^b \,\ldots\, \beta_{mk_p}^b \\
& \beta_a^f \,\ldots\, \beta_{ak_p}^f\, \sigma_j^2\, \sigma_m^2\, \sigma_a^2\, \sigma_{am}^2 \,],
\end{aligned}
$$

which contains the parameter vector of the target module, the hyperparameters of the kernels of the impulse response models of the other modules, and the parameters related to the covariance of the noise corrupting $w_j(t)$, $w_a(t)$ and $w_m(t)$. Note that θ appears in g_{ji} while the other parameters in η appear in g and in covariance of $\xi_{\mathcal{Y}}$. Therefore, we focus on the estimation of η, since it contains the parameter of interest θ. For this, we apply an Empirical Bayes (EB) approach, where the estimate of η is obtained by maximizing the marginal likelihood of the *observed data* $w_{\mathcal{Y}} = \begin{bmatrix} w_j^\top & w_a^\top \end{bmatrix}^\top$, obtained by integrating out the dependence on the missing node data and the impulse response of the modules,

$$\hat{\eta} = \arg\max_{\eta} p(w_{\mathcal{Y}}; \eta). \tag{10.27}$$

> **Remark 10.1** *If we do not consider additional node w_a, we can remove the extra layer of equation in $w_{\mathcal{Y}}$. In the above model* (10.26)*, it will be the second (block) row of equation in $w_{\mathcal{Y}}$ and therefore we need not model f. Now the model will depend upon the vector of parameters,*
>
> $$\eta := [\, \theta\, \lambda_j^s \,\ldots\, \lambda_{jk_p}^s\, \lambda_{c_1}^b \,\ldots\, \lambda_{mk_p}^b\, \beta_j^s \,\ldots\, \beta_{jk_p}^s\, \beta_m^b \,\ldots\, \beta_{mk_p}^b\, \sigma_j^2\, \sigma_m^2].$$
>
> *We do not need to estimate extra parameters $\lambda_a^f, \lambda_{c_1}^f, \ldots, \lambda_{ak_p}^f, \beta_a^f, \beta_{c_1}^f, \ldots, \beta_{ak_p}^f, \sigma_{am}^2, \sigma_a^2$.*

The first important problem with the above approach of parameter η inference in (10.27) is that we need to deal with the unknown missing node observation w_m. Secondly, due to the incomplete model, the marginal pdf of $w_{\mathcal{Y}}$ (i.e. $p(w_{\mathcal{Y}}; \eta)$) does not admit an analytical expression and cannot be computed under a closed-form solution. Adding to it, the maximization problem does not admit an explicit solution. In the next section, we study how to solve the marginal likelihood problem through a dedicated iterative scheme.

10.6 Parameter Inference

In this section, we provide the approach to deal with the missing node w_m and the above discussed problems and solve the marginal likelihood problem in

(10.27). We use the strategy of *data augmentation* [123, 132] to deal with the unknown missing node observations. In this data augmentation strategy, we treat the unknown node signal as auxiliary variables which are estimated along with the parameters in η. This data augmentation strategy has been used for state inference in identification of state-space models [113]. There are various methods that use the data augmentation strategy like the EM algorithm [35] for a Frequentist formulation of the identification problem and the *Gibbs sampler* [52] for a Bayesian formulation.

For the problem in (10.27), which is a Frequentist formulation, we solve it by deriving an iterative solution scheme through the EM algorithm. For this, we need to first define the *latent variables* whose estimation simplifies the computation of the marginal likelihood. The first natural choice is w_m, which is the missing node observation. Also, s, b and f are latent variables. Then, the solution to (10.27) using the EM algorithm is obtained by iterating among the following two steps:

- *E-Step:* Given an estimate $\hat{\eta}^{(n)}$ computed at the n^{th} iteration, compute
$$Q^{(n)}(\eta) = \mathbb{E}[\log p(w_{\bar{y}}, w_m, s, f, b; \eta)] \,, \tag{10.28}$$
where the expectation of the joint log-likelihood of w_j, w_a, w_m, s, f and b is taken with respect to the posterior $p(w_m, s, b, f | w_{\bar{y}}; \hat{\eta}^{(n)})$;
- *M-Step:* Update $\hat{\eta}$ by solving
$$\hat{\eta}^{(n+1)} = \arg\max_{\eta} Q^{(n)}(\eta) \,. \tag{10.29}$$

When the two steps are iterated, convergence to a stationary point of the marginal likelihood (which can be a local minima or global minima) is ensured [16]. In the next section, we show the clear advantage of using the EM algorithm. We have transformed the original marginal likelihood problem (10.27) to a sequence of problems that require solving (10.29) using the EM algorithm. We show that, when we use the EM method, the nonlinear optimization problem becomes a problem of iteratively constructing analytical solutions and solving scalar optimization problems, which significantly simplifies solving (10.27).

Also, the E-step in the algorithm involves computing expectation with respect to the posterior distribution $p(w_m, s, p, f | w_{\bar{y}})$, which is non-Gaussian and does not have an analytical form. Thus the integral in (10.28) is not tractable. In the next section, we present a solution to this problem by using a Markov Chain Monte Carlo (MCMC) method, *Gibbs sampler*.

10.6.1 Computation of E-step

In order to perform the E-step we resort to the Monte Carlo approximation of (10.28). This method has been introduced in [147] and is known as Monte Carlo

Expectation Maximization (MCEM). In this, we approximate (10.28) as,

$$Q^{(n)}(\eta) \approx \frac{1}{M} \sum_{i=1}^{M} \log p(w_{\bar{y}}, \bar{w}_m^{(i,n)}, \bar{s}^{(i,n)}, \bar{p}^{(i,n)}, \bar{f}^{(i,n)}; \eta) \,, \tag{10.30}$$

where $\bar{s}^{(i,n)}, \bar{p}^{(i,n)}, \bar{f}^{(i,n)}, \bar{w}_m^{(i,n)}$ are samples drawn at the n^{th} iteration from the posterior $p(w_m, s, p, f|w_{\bar{y}}; \hat{\eta}^{(n)})$. In order to draw samples from the posterior, we use the *Gibbs sampler*[2]. The idea behind the *Gibbs sampler*, which is a MCMC method, is to generate samples from a desired target distribution by simulating a Markov chain, with the target distribution as its stationary distribution. The Gibbs sampler produces samples from the posterior distribution by iteratively sampling each random variable conditioned on all other random variables [52]. Therefore to create samples from the joint posterior distribution, starting from an initialization $\bar{s}^{(0,n)}, \bar{p}^{(0,n)}, \bar{f}^{(0,n)}, \bar{w}_m^{(0,n)}$, we iteratively perform Algorithm 5 for a large number of iterations keeping the hyperparameters value fixed. Normally, we discard first few samples since the Markov chain will be poorly mixed and the obtained samples will be far away from the stationary distribution, which is the target distribution for the *Gibbs sampler*. Therefore, we discard the first B samples, and this is known as *burn-in* period. If the burn-in period is large enough, then we produce samples that come from the stationary distribution[3]. Another approach called *thinning* can be used to reduce the correlation in generated samples, where after the burn-in period each sample can be collected after κ iterations.

Algorithm 5 Gibbs sampler

```
1. sample $\bar{w}_m^{(i+1,n)} \sim p(w_m|w_j, \bar{s}^{(i,n)}, \bar{b}^{(i,n)}, \bar{f}^{(i,n)})$ ,
2. sample $\bar{s}^{(i+1,n)} \sim p(s|w_j, \bar{w}_m^{(i+1,n)}, \bar{b}^{(i,n)}, \bar{f}^{(i,n)})$ ,
3. sample $\bar{b}^{(i+1,n)} \sim p(b|w_j, \bar{w}_m^{(i+1,n)}, \bar{s}^{(i+1,n)}, \bar{f}^{(i,n)})$ ,
4. sample $\bar{f}^{(i+1,n)} \sim p(f|w_j, \bar{w}_m^{(i+1,n)}, \bar{s}^{(i+1,n)}, \bar{b}^{(i+1,n)})$,
```

It is important to note that in order to use the Gibbs sampler, the above conditional distributions should be known and we should be able to generate samples from them. Next we show that these conditional distributions have a convenient form.

Proposition 10.4 *Consider the model in* (10.26). *The conditional*

[2] There are other joint posterior approximation techniques like Variational Bayes approximations [8] and other MCMC methods [55], which can also be applied. In this chapter we resort to *Gibbs sampler*. *Gibbs sampler* does not require any tuning of proposal density and does not include any rejection step.

[3] The choice of burn-in period is a non-trivial problem which is not in the scope of this chapter and methods to address this problem have been provided in [55].

distributions of s, p, f, w_m are Gaussian and given by,

$$
\begin{aligned}
p(w_m|w_{\bar{\mathcal{Y}}}, s, b, f) &\sim \mathcal{N}(\mu_w, P_w)\,, && (10.31)\\
p(s|w_{\bar{\mathcal{Y}}}, w_m, b, f) &\sim \mathcal{N}(\mu_s, P_s)\,, && (10.32)\\
p(b|w_{\bar{\mathcal{Y}}}, w_m, s, f) &\sim \mathcal{N}(\mu_b, P_b)\,, && (10.33)\\
p(f|w_{\bar{\mathcal{Y}}}, w_m, s, b) &\sim \mathcal{N}(\mu_f, P_f)\,, && (10.34)
\end{aligned}
$$

where

$$
\begin{aligned}
P_w &= (\bar{\mu}_2^\top \Sigma^{-1} \bar{\mu}_2 + \Lambda_{22} - \begin{bmatrix}\Lambda_{21} & \Lambda_{22}\end{bmatrix} \bar{\mu}_2 - \bar{\mu}_2^\top \begin{bmatrix}\Lambda_{12}^\top & \Lambda_{22}^\top\end{bmatrix}^\top)^{-1}\,, && (10.35)\\
\mu_w &= P_w(\bar{\mu}_2^\top \begin{bmatrix}\Lambda_{11} & \Lambda_{12}\end{bmatrix}^\top w_{\bar{\mathcal{Y}}} + \begin{bmatrix}\Lambda_{21} & \Lambda_{22}\end{bmatrix} \bar{\mu}_1 - \Lambda_{21} w_{\bar{\mathcal{Y}}} - \bar{\mu}_2^\top \Sigma^{-1} \bar{\mu}_1), && (10.36)\\
P_s &= \left(\mathbf{K}_1^{-1} + \bar{\mathbf{W}}^\top \Sigma^{-1} \bar{\mathbf{W}}\right)^{-1}\,, && (10.37)\\
\mu_s &= P_s \bar{\mathbf{W}}^\top \Sigma^{-1} \left(w_{\mathcal{Y}} - \bar{\mu}_4\right)\,, && (10.38)\\
P_b &= \left(\mathbf{K}_2^{-1} + \bar{\mathbf{R}}^\top \Sigma^{-1} \bar{\mathbf{R}}\right)^{-1}\,, && (10.39)\\
\mu_b &= P_b \bar{\mathbf{R}}^\top \Sigma^{-1} \left(w_{\mathcal{Y}} - \bar{\mu}_5\right)\,, && (10.40)\\
P_f &= \left(\mathbf{K}_3^{-1} + \bar{\mathbf{Q}}^\top \Sigma^{-1} \bar{\mathbf{Q}}\right)^{-1}\,, && (10.41)\\
\mu_f &= P_f \bar{\mathbf{Q}}^\top \Sigma^{-1} \left(w_{\mathcal{Y}} - \bar{\mu}_6\right)\,. && (10.42)
\end{aligned}
$$

Proof: Collected in the appendix. The expressions for $\bar{\mathbf{W}}, \bar{\mathbf{R}}, \bar{\mathbf{Q}}, \bar{\mu}_1, \bar{\mu}_2, \bar{\mu}_4, \bar{\mu}_5$ and $\bar{\mu}_6$ are provided in the appendix. ■

Therefore, it is easy to set up the Gibbs sampler and sample from the joint posterior distribution, thereby approximating (10.28) using (10.30).

> **Remark 10.2** *When we do not consider additional node w_a, we do not consider the extra layer of equation in $w_{\bar{\mathcal{Y}}}$ (i.e. $w_{\mathcal{Y}}$ becomes w_j) and we need not model f. Therefore, we discard the use of f and expressions related to it. The conditional distributions follow the same equations as above.*

10.6.2 Computation of M-step

Next we move to the M-step where we update the vector of parameters according to (10.29). We need to maximize (10.30) with respect to the vector of parameters in η. We will now show that the optimization problem can be split into several independent optimization problems that depend on different components of the vector of parameters η. We can split the optimization problem as,

$$
\begin{aligned}
Q^{(n)}(\eta) &= \arg\max_{\eta} \frac{1}{M} \sum_{i=1}^{M} \log p(w_{\bar{\mathcal{Y}}}, \bar{s}^{(i,n)}, \bar{b}^{(i,n)}, \bar{f}^{(i,n)}, \bar{w}_m^{(i,n)}; \eta)\\
&= A + B + C + D\,, && (10.43)
\end{aligned}
$$

where,

$$\begin{aligned}
A &= \arg\max_{\theta,\bar\Sigma} \frac{1}{M}\sum_{i=1}^{M} \log p(w_{\bar y}, \bar w_m^{(i,n)} | \bar s^{(i,n)}, \bar b^{(i,n)}, \bar f^{(i,n)}) \\
B &= \arg\max \frac{1}{M}\sum_{i=1}^{M}\Bigg[\sum_{k\in\mathcal{D}_m^w} \log p(\bar b_k^{(i,n)}; \lambda_k^b, \beta_k^b) \\
&\quad + \log p(\bar b_m^{(i,n)}; \beta_m^b) + \sum_{k\in\mathcal{D}_m^u} \log p(\bar b_{mk}^{(i,n)}; \lambda_{mk}^b, \beta_{mk}^b)\Bigg] \\
C &= \arg\max \frac{1}{M}\sum_{i=1}^{M}\Bigg[\sum_{k\in j\cup\mathcal{D}_j^w\setminus\{i,m\}} \log p(\bar s_k^{(i,n)}; \lambda_k^s, \beta_k^s) \\
&\quad + \log p(\bar s_m^{(i,n)}; \beta_m^s) + \sum_{k\in\mathcal{D}_j^u\setminus\{j\}} \log p(\bar s_{jk}^{(i,n)}; \lambda_{jk}^s, \beta_{jk}^s)\Bigg] \\
D &= \arg\max \frac{1}{M}\sum_{i=1}^{M}\Bigg[\sum_{k\in a\cup\mathcal{D}_a^w\setminus\{m\}} \log p(\bar f_k^{(i,n)}; \lambda_k^f, \beta_k^f) \\
&\quad + \log p(\bar f_m^{(i,n)}; \beta_m^f) + \sum_{k\in\mathcal{D}_a^u} \log p(\bar f_{ak}^{(i,n)}; \lambda_{ak}^f, \beta_{ak}^f)\Bigg],
\end{aligned}$$

and $\bar\Sigma$ represent the parameters in the covariance matrix $\sigma_j^2, \sigma_m^2, \sigma_a^2, \sigma_{am}^2$.

Update of kernel hyperparameters

From (10.44), we can see that the hyperparameters of each kernel can be updated independently from the rest of the parameters in η. The following proposition provides a means to update the kernel hyperparameters, except the hyperparameters of the kernel for which λ's are set to 1.

Proposition 10.5 *Let*

$$\hat s_k^{(n)} = \frac{1}{M}\sum_{i=1}^{M} \bar s_k^{(i,n)},$$

$$\hat{\mathbf{S}}_k^{(n)} = \frac{1}{M}\sum_{i=1}^{M} (\bar s_k^{(i,n)} - \hat s_k^{(n)})(\bar s_k^{(i,n)} - \hat s_k^{(n)})^\top,$$

and analogously define $\hat b_k^{(n)}, \hat f_k^{(n)}, \hat{\mathbf{B}}_k^{(n)}, \hat{\mathbf{F}}_k^{(n)}$. *Define*

$$Q_{\beta_k}^{(n)}(\beta_k) = \log\det(K_{\beta_k}) + l\log\Big(\hat s_k^{\top(n)}(K_{\beta_k})^{-1}\hat s_k^{(n)} + \mathrm{tr}\big((K_{\beta_k})^{-1}\hat{\mathbf{S}}_k^{(n)}\big)\Big) \quad (10.44)$$

for $k = \{j, c_1, \ldots, jk_p\}$ *where* $c_1, \ldots, c_p$ *and* $k_1, \ldots, k_p$ *are the elements of* *the set $\mathcal{D}_j^w \backslash \{i\}$ and $\mathcal{D}_j^u \backslash \{j\}$ respectively. Then,*

$$\hat{\beta_k}^{(n+1)} = \arg\min_{\beta_k \in [0,1)} Q_{\beta}{}_k^{(n)}(\beta_k); \tag{10.45}$$

$$\hat{\lambda_k}^{(n+1)} = \frac{1}{l}(\hat{s}_k^{\top(n)}(K_{\hat{\beta}_k^{(n+1)}})^{-1}\hat{s}_k^{(n)} + \operatorname{tr}\big((K_{\hat{\beta}_k^{(n+1)}})^{-1}\hat{\mathbf{S}}_k^{(n)}\big)). \tag{10.46}$$

The updates for β and λ for impulse responses $b_k, k \in m \cup \mathcal{D}_m^w$, $b_{mk}, k \in \mathcal{D}_m^u$, and $f_k, k \in a \cup \mathcal{D}_a^w$, $f_{ak}, k \in \mathcal{D}_a^u$ are updated analogously by using $\hat{b}_k^{(n)}, \hat{\mathbf{B}}_k^{(n)}$ and $\hat{f}_k^{(n)}, \hat{\mathbf{F}}_k^{(n)}$ respectively.

Proof: See the appendix. ■

To tackle the identifiability issues, we have fixed the hyperparameters λ_m^s, λ_m^b and λ_m^f (i.e. λ's corresponding to the modules having missing node as inputs) to be 1. We now provide means to update the respective β hyperparameters of the kernel of the corresponding modules using the following proposition.

Proposition 10.6 *The updates of kernel's hyperparameters related to the impulse response of modules with missing node as inputs are obtained by solving the scalar optimization problem in the domain $[0,1)$,*

$$\hat{\beta}_m^{s(n+1)} = \arg\min_{\beta_m^s} \log\det(K_{\beta_m^s}) + \hat{s}_m^{\top(n)}(K_{\beta_m^s})^{-1}\hat{s}_m^{(n)} + \operatorname{tr}\big((K_{\beta_m^s})^{-1}\hat{\mathbf{S}}_m^{(n)}\big) \tag{10.47}$$

The updates for β_m^b and β_m^f for impulse responses b_m and f_m are updated analogously by using $\hat{b}_m^{(n)}, \hat{\mathbf{B}}_m^{(n)}$ and $\hat{f}_m^{(n)}, \hat{\mathbf{F}}_m^{(n)}$ respectively.

Proof: See the appendix. ■

Remark 10.3 *The optimization problem in* (10.45) *and* (10.47) *can be difficult to perform in practice when the determinant of the kernel has a very low value or when the inversion of the kernel becomes difficult. To tackle this, we exploit the factorization of the* first order stable spline kernel *as in [14] by writing $K_{\beta_k} = LD(\beta)L^T$, where L is lower-triangular with known entries (essentially, an "integrator") and $D(\beta)$ is diagonal with entries essentially being an exponential functions of β. Using the above technique also increases the computation speed of the algorithm.*

Remark 10.4 *When we do not consider an additional node w_a, we need to update only the hyperparameter λ's and β's related to impulse responses in s, b.*

Update of θ and noise covariance

Following (10.44), the updates of θ and the noise covariance parameters in η are independent of the kernel hyperparameters. Following a reasoning similar to [3], θ and Σ are updated as per the *Proposition 10.7*.

Proposition 10.7 *Let* $\bar{\varepsilon}_j^{(i,n)}(\theta) = w_j - W_{ji}g_{ji}(\theta) - u_j - \bar{\mathbf{W}}^{(i,n)}\bar{s}^{(i,n)}$, $\tilde{\Sigma} = \begin{bmatrix} \sigma_a^2 & \sigma_{am}^2 \\ \sigma_{am}^2 & \sigma_m^2 \end{bmatrix}$*, and*

$$\bar{\varepsilon}^{(i,n)}(t) = \begin{bmatrix} w_a(t) \\ \bar{w}_m^{(i,n)}(t) \end{bmatrix} - \begin{bmatrix} u_a(t) \\ u_m(t) \end{bmatrix} - \begin{bmatrix} \bar{\mathbf{Q}}^{(i,n)}(t,\star) & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{R}}^{(i,n)}(t,\star) \end{bmatrix} \begin{bmatrix} \bar{f}^{(i,n)} \\ \bar{b}^{(i,n)} \end{bmatrix},$$

where $\bar{\mathbf{Q}}^{(i,n)}(t,\star)$, $\bar{\mathbf{R}}^{(i,n)}(t,\star)$ *corresponds to the* t^{th} *row of the matrix* $\mathbf{Q}$, $\mathbf{R}$ *respectively, with* w_m *in the matrices substituted with* $\bar{w}_m^{(i,n)}$. $\bar{\mathbf{W}}^{(i,n)}$ *corresponds to the matrix* $\mathbf{W}$, *with* w_m *in the matrices substituted with* $\bar{w}_m^{(i,n)}$. *Define*

$$\hat{\varepsilon}^{(n)}(t) = \frac{1}{M}\sum_{i=1}^{M} \bar{\varepsilon}^{(i,n)}(t),$$

$$\hat{\mathbf{E}}^{(n)}(t) = \frac{1}{M}\sum_{i=1}^{M} (\bar{\varepsilon}^{(i,n)}(t) - \hat{\varepsilon}^{(n)}(t))(\bar{\varepsilon}^{(i,n)}(t) - \hat{\varepsilon}^{(n)}(t))^\top.$$

Then

$$\begin{aligned} \hat{\theta}^{(n+1)} &= \arg\min_{\theta} \left[g_{ji}^\top \hat{A}^{(n)} g_{ji} - 2\hat{b}^{(n)\top} g_{ji} \right], \\ \hat{\sigma}_j^{2(n+1)} &= \frac{1}{NM}\sum_{i=1}^{M} \left\| \bar{\varepsilon}_j^{(i,n)}(\hat{\theta}^{n+1}) \right\|^2, \\ \hat{\tilde{\Sigma}}^{(n+1)} &= \frac{1}{N}\left(\sum_{t=1}^{N} [\hat{\varepsilon}^{(n)}(t,\hat{\theta}^{(n+1)})\hat{\varepsilon}^{(n)}(t,\hat{\theta}^{(n+1)})^\top + \hat{\mathbf{E}}^{(n)}(t)]\right). \end{aligned} \tag{10.48}$$

Proof: See the appendix. ■

> **Remark 10.5** *If* g_{ji} *is linearly parameterized in* θ *(e.g. in case of FIR models), the above problem related to the update of* θ *becomes quadratic and a closed-form solution is achieved. That is, if* $g_{ji} = M\theta$ *where* $M \in \mathbb{R}^{N \times n_\theta}$*, then*
>
> $$\hat{\theta}^{(n+1)} = \left(M^\top \hat{A}^{(n)} M\right)^{-1} M^\top \hat{b}^{(n)}. \tag{10.49}$$

Note that, as shown in Chapter 7, we can update the parameter of the target module (i.e. θ) using similar analytical expression as in (10.49) for any other rational model structures as well (e.g. BJ models). This can be done by following the similar approach of Chapter 7. From (10.48), we can observe that the update of $\bar{\Sigma}$ in each iteration of the MCEM algorithm is a closed form analytical solution and the update of θ is a nonlinear least-squares problem with decision variables being the parameters of the target module which are fewer than a direct PEM that includes the nuisance modules parameters as well in the problem as decision variables. Also the result of Propositions 10.5 and 10.6 show that the update of kernel hyperparameter β's are scalar optimization problems and λ's have closed form solutions. Therefore, we have obtained a fast iterative procedure that follows simple rules to update the parameters, and provides a local solution to the marginal likelihood problem (10.27) under the presence of missing node observations. Algorithm 6 summarizes the steps to follow to obtain $\hat{\eta}$ and therefore $\hat{\theta}$.

Algorithm 6 Algorithm for local module identification in dynamic networks under missing node observations

```
1. Set $n = 0$, Initialize $\hat{\eta}^{(0)}$.

2. Run Gibbs sampler according to Algorithm 5 and collect $M$ samples after
   discarding first $B$ samples for burn-in period using the result of Proposition
   10.4.

3. Update kernel hyperparameters using the result of Proposition 10.5 and
   10.6.

4. Update $\hat{\theta}^{(n+1)}$ and $\hat{\Sigma}^{(n+1)}$ using result of proposition 10.7.

5. Update $\hat{\eta}^{(n+1)}$ using the above updated values of the parameters.

6. Set $n = n + 1$.

7. Repeat from steps (2) to (8) until convergence.
```

The initialization can be done by randomly choosing η considering the constraints of the hyperparameters. The convergence criterion for the algorithm depend on the value of $\frac{\left\|\hat{\eta}^{(n)}-\hat{\eta}^{(n-1)}\right\|}{\left\|\hat{\eta}^{(n-1)}\right\|}$. This value should be small for convergence so that the algorithm can be terminated. A value of 10^{-2} is considered for the numerical simulations in Section 10.7. The other convergence criterion is the maximum number of iterations. It is taken as 50. For the numerical simulations, the initialization of the latent variables (s, p, f, w_m) for the Gibbs sampler are taken as a zero vector. The number of samples M for the Gibbs sampler is taken as 100 and the burn-in period B is equal to 2000.

Remark 10.6 *The above developed method can be applied when we have the input of the target module (i.e. w_i) as the missing node observation with slight modifications in the above results. However, in this case, we might* *face identifiability issues of the target module. This is because of the fact that we can estimate the missing node signal and the target module up to a scaling factor. This has been a common issue in blind system identification [1].*

10.7 Numerical simulations

Numerical simulations are performed to evaluate the performance of the developed method. The simulations are performed on the dynamic network depicted in Figure 10.1. The goal is to identify G_{31}^0, which is the target module. The modules of the network in Figure 10.1 are given by,

$$\begin{aligned}
G_{31}^0 &= \frac{q^{-1}+0.05q^{-2}}{1+q^{-1}+0.6q^{-2}} = \frac{b_1^0q^{-1}+b_2^0q^{-2}}{1+a_1^0q^{-1}+a_2^0q^{-2}}\\
G_{32}^0 &= \frac{0.225q^{-1}}{1+0.5q^{-1}};\\
G_{34}^0 &= \frac{1.184q^{-1}-0.647q^{-2}+0.151q^{-3}-0.082q^{-4}}{1-0.8q^{-1}+0.279q^{-2}-0.048q^{-3}+0.01q^{-4}};\\
G_{14}^0 = G_{21}^0 &= \frac{0.4q^{-1}-0.5q^{-2}}{1+0.3q^{-1}}; H_1^0 = \frac{1}{1+0.2q^{-1}};\\
G_{12}^0 = G_{24}^0 &= \frac{0.4q^{-1}+0.5q^{-2}}{1+0.3q^{-1}}; H_2^0 = \frac{1}{1+0.3q^{-1}};\\
H_4^0 = 1; H_3^0 &= \frac{1-0.505q^{-1}+0.155q^{-2}-0.01q^{-3}}{1-0.729q^{-1}+0.236q^{-2}-0.019q^{-3}}.
\end{aligned}$$

For estimation G_{31}^0 using the direct method, we need to measure w_1, w_2, w_3 and w_4 and solve a 3-input/1-output MISO identification problem with $w_1(t)$, $w_2(t)$ and $w_4(t)$ as inputs. w_2 needs to be included as predictor input in order to satisfy the parallel path/ loop condition 2.1 and w_4 needs to be included as predictor input to satisfy the confounding variable condition. Now, we consider the case where we cannot measure w_2, which leads to lack of consistency in the direct method since we cannot satisfy the parallel path/loop condition. In this case, we resort to the approach developed in this chapter and resort to the following options:

1. consider w_2 as missing node, i.e. $w_m = w_2$, and consider the predictor model with $w_\mathcal{Y} = \{w_3, w_2\}$. We add to $w_\mathcal{D}$ the measured node signals that have unmeasured paths to $w_\mathcal{Y}$ (i.e. w_1, w_4) and the missing node signal w_2. Therefore $w_\mathcal{D} = \{w_1, w_4, w_2\}$, $w_\mathcal{W} = \{w_1, w_2, w_3, w_4\}$, $w_\mathcal{O} = \{w_3\}$ and $w_\mathcal{Q} = \{w_2\}$. By this signal selection , we can verify that the conditions 10.1, 10.2 and 10.3 are satisfied. This identification strategy is mentioned below as MC-EBDM;

2. consider w_2 as missing node and add the descendant w_1 of w_2 as additional output, i.e. $w_m = w_2$, $w_a = w_1$ and consider the model with $w_\mathcal{Y} = \{w_3, w_2, w_1\}$. We add to $w_\mathcal{D}$ the measured node signals that have

unmeasured paths to $w_{\mathcal{Y}}$ (i.e. w_1, w_4) and the missing node signal w_2. Therefore $w_{\mathcal{D}} = \{w_1, w_4, w_2\}$, $w_{\mathcal{W}} = \{w_1, w_2, w_3, w_4\}$, $w_{\mathcal{O}} = \{w_3\}$ and $w_{\mathcal{Q}} = \{w_1, w_2\}$. By this signal selection , we can verify that the conditions 10.1, 10.2 and 10.3 are satisfied. This identification strategy is mentioned below as MC-EBDMA.

We compare the following identification strategies:

MC-EBDM This is the method developed in this chapter, namely Empirical Bayes Direct method with Monte Carlo sampling to deal with missing nodes; in particular, this estimator does not use additional node(s) and considers the predictor model $\{w_1, w_2, w_4\} \to \{w_3, w_2\}$;

MC-EBDMA This is a variant of MC-EBDM that uses w_1 as additional node; and considers the predictor model $\{w_1, w_2, w_4\} \to \{w_1, w_2, w_3\}$

EBDM+M This is the EBDM method developed in Chapter 7; this estimator does not encompass missing nodes and considers the predictor model $\{w_1, w_4\} \to \{w_3\}$;in other words it discards the non-measured node signal w_2;

EBDM This is the same estimator as the previous one, with the assumption that the missing node w_2 is measurable (oracle assumption). We use this estimator as an upper bound of the performance of our developed method to reconstruct the missing node observation and identify the target module. Therefore, it considers a predictor model $\{w_1, w_2, w_4\} \to \{w_3\}$ where w_2 is known;

DM+TO This is the standard MISO direct method first proposed in [124], with the assumption that the missing node w_2 is measurable (oracle assumption). Therefore, it considers a predictor model $\{w_1, w_2, w_4\} \to \{w_3\}$ where w_2 is known and assumes a fully parametric model structure. Note that in order to avoid biased target module estimates in the direct method framework, we need w_2 to be measured and included as one of the predictor inputs [29];

DM+TO+M This is the same estimator as the previous one; it assumes a fully parametric model structure and has no specific way to deal with missing nodes and considers a predictor model $\{w_1, w_4\} \to \{w_3\}$.

We run 50 independent Monte Carlo experiments where the data are generated using known reference signals $r_2(t)$ and $r_4(t)$ that are realizations of white noise with unit variance. The number of data samples is $N = 150$. The noise sources $e_1(t)$, $e_2(t)$, $e_3(t)$ and $e_4(t)$ have variance 0.05, 0.08, 0.5, 0.1, respectively. We assume that we know the model order of $G^0_{31}(q)$. For the method DM+TO+M, we solve a 2-input/1-output MISO identification problem with $w_1(t)$ and $w_4(t)$ as inputs, which should lead to a biased target module estimate [29]. As for DM+TO, with the assumption that the missing node w_2 is measurable (oracle assumption), we solve a 3-input/1-output MISO identification problem with $w_1(t)$, $w_2(t)$ and $w_4(t)$ as inputs in order to compare the results of our developed method. For both these cases we consider that the model orders of all the modules in the MISO structure are known. Analogously, EBDM considers a 3-input/1-output MISO identification problem, while EBDM+M solves a 2-input/1-output MISO identification problem. For MC-EBDM, MC-EBDMA, EBDM, EBDM+M we choose $l = 15$.

To evaluate the performance of the methods, we use the standard goodness-of-fit metric,

$$\text{Fit}_{imp} = 1 - \frac{\left\| g_{ji}^0 - \hat{g}_{ji} \right\|_2}{\left\| g_{ji}^0 - \bar{g}_{ji} \right\|_2}, \; ; \; \text{Fit}_\theta = 1 - \frac{\left\| \theta^0 - \hat{\theta} \right\|_2}{\left\| \theta^0 - \bar{\theta} \right\|_2},$$

where Fit_{imp} and Fit_θ are the fit of the estimated impulse response and estimated parameters of the target module respectively. g_{ji}^0 is the true value of the impulse response of G_{ji}^0, $\hat{g}_{ji}$ is the impulse response of the estimated target module, $\bar{g}_{ji}$ is the sample mean of g_{ji}^0, θ^0 is the true parameter of the target module, $\hat{\theta}$ is the estimated value of the parameter and $\bar{\theta}$ is the sample mean of θ^0. The box plots of the fit of the impulse response and box plots of the fit of the parameters of $G_{31}(q)$ are shown in Figure 10.3 and 10.4 respectively for the above mentioned methods.

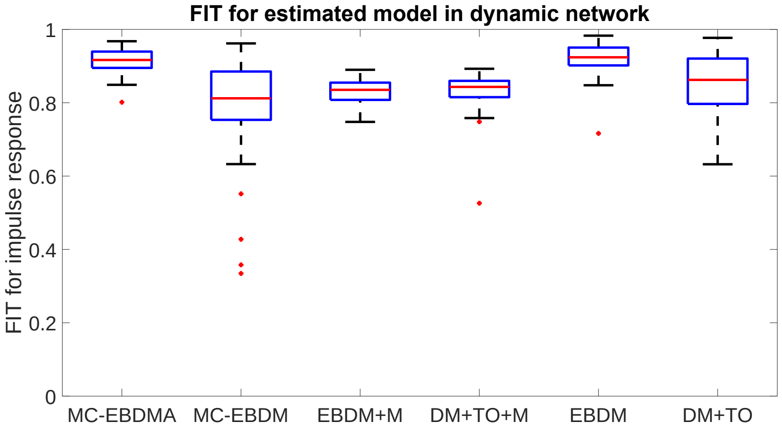


Figure 10.3: Box plot of the fit of the impulse response of $\hat{G}_{31}$ obtained using different methods. EBDM and DM+TO assumes that the missing node w_2 is measurable (oracle assumption) and use it for the estimation.

It can be noted that both MC-EBDMA, despite considering w_2 to be non-measured, achieve significantly better fit than the other methods that do not consider w_2 to be known (i.e. DM+TO+M and EBDM+M). Comparing with methods that consider w_2 to be known (i.e. DM+TO and EBDM), the novel estimator MC-EBDMA performs better than the direct method; furthermore, MC-EBDMA achieves a fit comparable to the fit obtained by the oracle EBDM. Also, the performance of MC-EBDM is poor compared to other methods, and thus shows the importance of including additional node(s) (i.e. MC-EBDMA). In Figure 10.5, we show the re-constructed signal w_2 for one MC simulation using MC-EBDM and MC-EBDMA. It can be seen that considering additional nodes

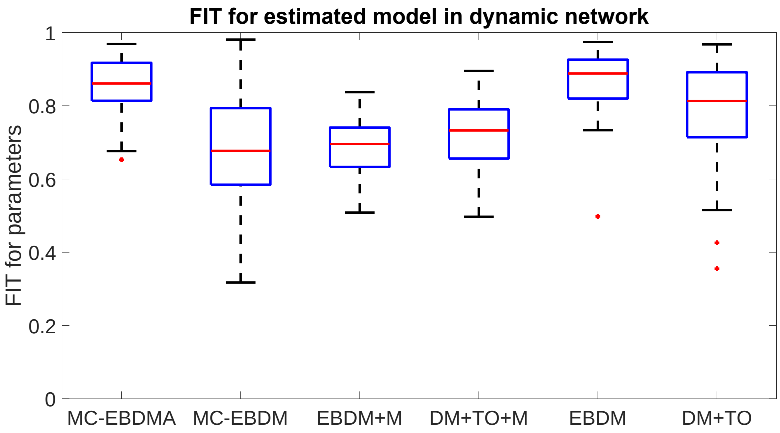


Figure 10.4: Box plot of the fit of the parameters of $\hat{G}_{31}$ obtained using different methods.

aid better reconstruction of the missing node observation and provides better estimates. Figure 10.6 shows the box plot of each parameter estimates of G_{31}. It is evident that the developed method MC-EBDMA provides smaller bias and greatly reduced variance under the case of missing node observations compared to the other methods. The reduction in variance is attributed to the regularization approach used in this method. Figure 10.6 again shows the importance of adding additional node(s) (i.e. MC-EBDMA) since the MC-EBDM has a larger variance compared to the other methods. Therefore, compared to the available methods for network identification, the developed framework stands out as an effective method that can handle the situation of missing node observations by reconstructing the node signal and offer reduced variance estimates. Considering the situation of large-sized networks, the developed method also circumvents the model order selection step that is required for the standard direct methods, which leads to computational burden and inaccurate results.

10.8 Conclusions

Sensor selection and placement has been a important practical problem in dynamic networks and it is not always possible to have measurements of certain node signals. When certain node measurements are not available, the identification performed using the available methods leads to less accurate and biased target module estimates. In this chapter, we have introduced an effective method to identify modules embedded in a dynamic networks under the situation of missing node observation by re-constructing these node observations and using regularized kernel-based approach. The introduced method also circumvents the model order selection step for all the modules other than the desired target module and offer reduction in the number of parameters to be

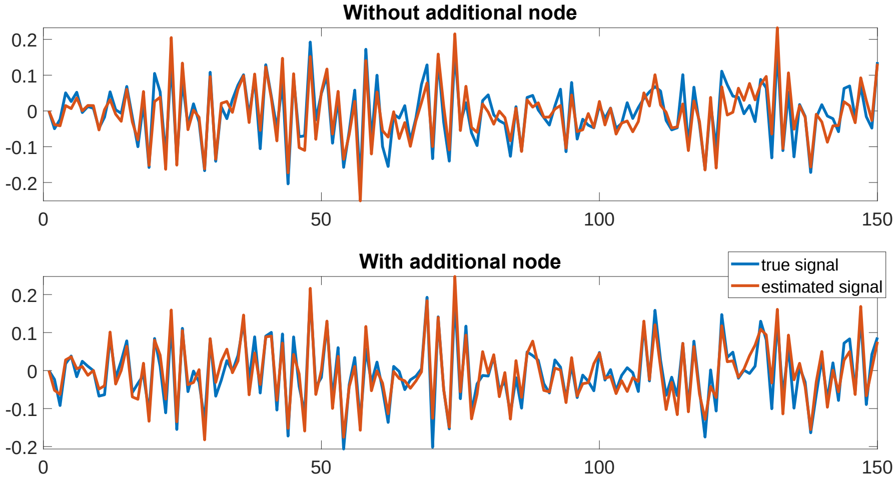


Figure 10.5: Re-constructed missing observation w_2 (normalized) signal for one MC simulation using MC-EBDM (top) and MC-EBDMA (bottom), compared with the measured value (blue) of w_2 over $N = 150$ data points.

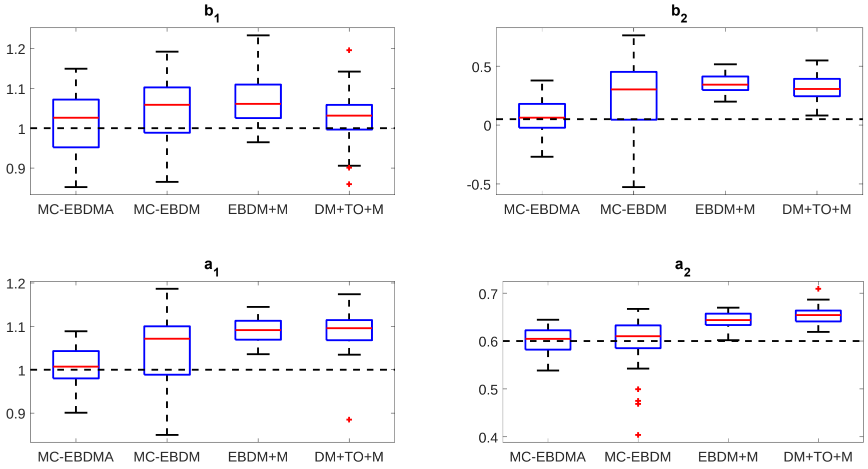


Figure 10.6: Box plot of the estimate for each parameter obtained from 50 MC simulations using different identification methods.

estimated by incorporating regularized non-parametric methods. The former offers lower computational burden and the latter offer reduced mean squared error of the estimated target module. Numerical simulations performed with a dynamic network example shows promising results, and illustrate the potentials of the developed method to reconstruct the missing node observations and provide reduced variance estimates due to the integration of a regularization approach.

In this chapter, we provide a framework to deal with the problem of missing node observations using a data augmentation strategy and regularized kernel-based methods. For this, we use the particle approximation MCEM approach to solve the E-step of the EM algorithm by drawing samples from the posterior using Gibbs sampler. There are other approaches like Variational Bayes EM (VBEM) [8] that can solve the E-step of the EM algorithm and may offer lower computational complexity compared to MCEM. The introduced framework in this chapter is flexible to tools that can solve the E-step.

Appendices

10.A Proof of Proposition 10.1

Using the result of Proposition 4.2, the systems equations for the output variables in $w_{\mathcal{Y}}$ for the network represented by (2.2) can always be written as,

$$w_{\mathcal{Y}} = \bar{G}w_{\mathcal{D}} + \bar{H}\xi_{\mathcal{Y}} + \bar{R}u, \tag{10.50}$$

where $\xi_{\mathcal{Y}}$ a white noise process with dimensions conforming to $w_{\mathcal{Y}}$, with $cov(\xi_{\mathcal{Y}}) = \bar{\Lambda}$ and with $\bar{H}$ being monic, stable and stably invertible. $\bar{G}, \bar{H}, \bar{R}$ are already defined in the proof of Proposition 4.2. Also, for this proof we use the notations and terms that are already defined in 4.2.

Now we present the following preparatory lemmas.

Lemma 10.2 *If Condition 10.1 is satisfied, then* $\tilde{H}_{jk} = \tilde{H}_{kj} = 0$ *for all* $k \in \mathcal{Y} \cup \mathcal{U}\backslash\{j\}$ *and* $\grave{H}_{jk} = 0, k \in \mathcal{U}$*.*

Proof: Using the result in Corollary 4.1, considering $\mathcal{C} = \mathcal{Y} \cup \mathcal{U}$, $\mathcal{C}_1 = \{j\}$ and $\mathcal{C}_2 = \mathcal{Y} \cup \mathcal{U}\backslash\{j\}$, we have $\tilde{H}_{jk} = \tilde{H}_{kj} = 0$ for all $k \in \mathcal{Y} \cup \mathcal{U}\backslash\{j\}$ if condition 10.1 is satisfied. $\grave{H}_{j\mathcal{U}}$ is the row vector corresponding to the row of node signal j. From the definition of $\grave{H}_{j\mathcal{U}}$, we can write $\grave{H}_{j\mathcal{U}} = \tilde{H}_{j\mathcal{U}}\tilde{H}_{\mathcal{U}\mathcal{U}}^{-1}$. Since $\tilde{H}_{j\mathcal{U}} = 0$ if condition 10.1 is satisfied, we have $\grave{H}_{j\mathcal{U}} = 0$. ■

Lemma 10.3 *Consider that condition 10.1 and 10.2 are satisfied, then* $\breve{G}'_{j\mathcal{O}} = 0$ *when* $j \in \mathcal{Q}$*.*

Proof: We have $\breve{G}'_{j\mathcal{O}} = \breve{G}_{j\mathcal{O}} - \grave{H}_{j\mathcal{U}}\breve{G}_{\mathcal{U}\mathcal{O}}$. From result of Lemma 10.2, if condition 10.1 is satisfied, we have $\grave{H}_{j\mathcal{U}} = 0$. When condition 10.2 is satisfied and since $j \in \mathcal{Q}$, we have $\breve{G}_{j\mathcal{O}} = 0$. Thus, we get the result of the Lemma. ■

Having presented the preparatory lemmas, we now provide the proof of proposition 10.1 which is based on conditions 10.1 and Condition 10.2 being satisfied.

Let $\ell \in \mathcal{O} \cup \mathcal{Q}\backslash\{j\}$. If $j \in \mathcal{Q}$, from (4.32) we have $\check{H}_{j\ell} = \tilde{H}'''_{j\ell} = (I - \breve{G}''_{jj})^{-1}\tilde{H}''_{j\ell} = (I - \breve{G}''_{jj})^{-1}(\tilde{H}'_{j\ell} + \breve{G}'_{j\mathcal{O}}\tilde{H}''_{\mathcal{O}\ell})$ and if $j \in \mathcal{O}$ we have $\check{H}_{j\ell} = \tilde{H}''_{j\ell} = (I - \breve{G}'_{jj})^{-1}\tilde{H}'_{j\ell}$. Here, $\breve{G}'_{j\mathcal{O}} = \breve{G}_{j\mathcal{O}} - \grave{H}_{j\mathcal{U}}\breve{G}_{\mathcal{U}\mathcal{O}}$ and $\tilde{H}'_{j\ell} = \tilde{H}_{j\ell} - \grave{H}_{j\mathcal{U}}\tilde{H}_{\mathcal{U}\ell}$. Since, $\grave{H}_{j\mathcal{U}} = 0$ and condition 10.2 is satisfied, we have $\breve{G}'_{j\mathcal{O}} = 0$, $\tilde{H}'_{j\ell} = \tilde{H}_{j\ell}$ and $\breve{G}''_{jj} = \breve{G}'_{jj} = \breve{G}_{jj}$. Therefore, $\check{H}_{j\ell} = (I - \breve{G}_{jj})^{-1}\tilde{H}_{j\ell}$. Since we know that $\tilde{H}_{j\ell} = 0$ for all $\ell \in \mathcal{O} \cup \mathcal{Q}\backslash\{j\}$, we conclude that $\check{H}_{j\ell} = 0$ for all $\ell \in \mathcal{O} \cup \mathcal{Q}\backslash\{j\}$. This will lead to $\bar{H}_{j\ell} = 0$ for all $\ell \in \mathcal{Q} \cup \mathcal{O}\backslash\{j\}$ when spectral factorization is performed on $\tilde{H}_r$ in the last steps of Proposition 4.2.

10.B Proof of Proposition 10.2

First we present the following preparatory lemma.

Lemma 10.4 *Consider the network equation in (10.50). If Condition 10.1 and Condition 10.2 are satisfied, then:*

$$\bar{G}_{j\ell}{=}(I{-}\breve{G}_{jj})^{-1}\breve{G}_{j\ell}\ \forall\ \ell \in \mathcal{Q} \cup \mathcal{U}, \tag{10.51}$$

$$\bar{R}_{jj} = (I{-}\breve{G}_{jj})^{-1}, \tag{10.52}$$

$$\bar{R}_{j\ell} = (I{-}\breve{G}_{jj})^{-1}\breve{R}_{j\ell}\ \forall\ \ell \in \mathcal{Z}, \tag{10.53}$$

$$\bar{R}_{j\ell} = 0\ \forall\ \ell \in \mathcal{Y} \cup \mathcal{U}\backslash\{j\}. \tag{10.54}$$

Proof: Using the result in Lemma 10.2, if Condition 10.1 is satisfied, then $\grave{H}_{j\mathcal{U}} = 0$. We have the following cases that can occur:

1. $j = \mathcal{O}$ and $\ell \in \mathcal{U}$. From (4.23) we have $\bar{G}_{j\ell} = [(I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}]_{(j,:)}\breve{G}'_{\mathcal{O}\ell}$ where $\breve{G}'_{\mathcal{O}\mathcal{O}} = \breve{G}_{\mathcal{O}\mathcal{O}} - \grave{H}_{\mathcal{O}\mathcal{U}}\breve{G}_{\mathcal{U}\mathcal{O}}$ is given by (4.17) and $\breve{G}'_{\mathcal{O}\ell} = \breve{G}_{\mathcal{O}\ell} - \grave{H}_{\mathcal{O}\mathcal{U}}\breve{G}_{\mathcal{U}\ell} + \grave{H}_{\mathcal{O}\ell}$ is given by (4.18). Since $\grave{H}_{j\mathcal{U}} = \grave{H}_{j\ell} = 0$ and condition 10.2 is satisfied, we have $\bar{G}_{j\ell} = [(I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}]_{(j,:)}\breve{G}'_{\mathcal{O}\ell} = (I - \breve{G}'_{jj})^{-1}\breve{G}'_{j\ell}$ where $\breve{G}'_{jj} = \breve{G}_{jj} - \grave{H}_{j\mathcal{U}}\breve{G}_{\mathcal{U}j}$ is given by (4.17) and $\breve{G}'_{j\ell} = \breve{G}_{j\ell} - \grave{H}_{j\mathcal{U}}\breve{G}_{\mathcal{U}\ell} + \grave{H}_{j\ell}$ is given by (4.18). Since $\grave{H}_{j\mathcal{U}} = \grave{H}_{j\ell} = 0$, we have $\breve{G}'_{jj} = \breve{G}_{jj}$ and $\breve{G}'_{j\ell} = \breve{G}_{j\ell}$, this directly leads to (10.51).

2. $j = \mathcal{O}$ and $\ell \in \mathcal{Q}$. From (4.23) we have $\bar{G}_{j\ell} = [(I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}]_{(j,:)}\breve{G}'_{\mathcal{O}\ell}$. Since $\grave{H}_{j\mathcal{U}} = \grave{H}_{j\ell} = 0$ and condition 10.2 is satisfied, we have $\bar{G}_{j\ell} = [(I - \breve{G}'_{\mathcal{O}\mathcal{O}})^{-1}]_{(j,:)}\breve{G}'_{\mathcal{O}\ell} = (I - \breve{G}'_{jj})^{-1}\breve{G}'_{j\ell}$ where $\breve{G}'_{jj}$ and $\breve{G}'_{j\ell}$ are given by (4.17). Since $\grave{H}_{j\mathcal{U}} = 0$, we have $\breve{G}'_{jj} = \breve{G}_{jj}$ and $\breve{G}'_{j\ell} = \breve{G}_{j\ell}$, leading to (10.51).

3. $j \in \mathcal{Q}$ and $\ell \in \mathcal{U}$. From (4.31) we have $\bar{G}_{j\ell} = (I - \breve{G}''_{jj})^{-1}\breve{G}''_{j\ell}$ where $\breve{G}''_{jj}$ and $\breve{G}''_{j\ell}$ are given by (4.25). If Condition 10.1 and 10.2 are satisfied, as a result

of Lemma 10.3, we have $\breve{G}'_{j\mathcal{O}}\bar{G}_{\mathcal{O}\ell} = \breve{G}'_{j\mathcal{O}}\bar{G}_{\mathcal{O}j} = 0$. Therefore (4.25) leads to $\breve{G}''_{jj} = \breve{G}'_{jj}$ which is specified by (4.17), and $\breve{G}''_{j\ell} = \breve{G}'_{j\ell}$ which is given by (4.18). Since $\grave{H}_{j\mathcal{U}} = \grave{H}_{j\ell} = 0$, we have $\breve{G}'_{jj} = \breve{G}_{jj}$ and $\breve{G}'_{j\ell} = \breve{G}_{j\ell}$, this leads to (10.51).

4. $j \in \mathcal{Q}$ and $\ell \in \mathcal{Q}$. Since $j \neq \ell$ it follows from (4.30) that $\bar{G}_{j\ell} = (I - \breve{G}''_{jj})^{-1}\breve{G}''_{j\ell}$ where $\breve{G}''_{jj}$ and $\breve{G}''_{j\ell}$ are given by (4.25). (4.25) leads to $\breve{G}''_{j\ell} = \breve{G}'_{j\ell}$ and $G''_{jj} = \breve{G}'_{jj}$ when Condition 10.2 and Condition 10.1 are satisfied. Therefore for this case, $\breve{G}''_{jj} = \breve{G}'_{jj}$ and $\breve{G}''_{j\ell} = \breve{G}'_{j\ell}$, which are given by (4.17). Since $\grave{H}_{j\mathcal{U}} = 0$, we have $\breve{G}'_{jj} = \breve{G}_{jj}$ and $\breve{G}'_{j\ell} = \breve{G}_{j\ell}$, and this leads to (10.51).

From (4.29) it follows that

$$\bar{u} = \bar{R}u = \begin{bmatrix} \bar{R}_{\mathcal{Q}\mathcal{Q}} & \bar{R}_{\mathcal{Q}\mathcal{O}} & \bar{R}_{\mathcal{Q}\mathcal{U}} & \bar{R}_{\mathcal{Q}\mathcal{Z}} \\ 0 & \bar{R}_{\mathcal{O}\mathcal{O}} & \bar{R}_{\mathcal{O}\mathcal{U}} & \bar{R}_{\mathcal{O}\mathcal{Z}} \\ 0 & 0 & I & \breve{R}_{\mathcal{U}\mathcal{Z}} \end{bmatrix} \begin{bmatrix} u_{\mathcal{Q}} \\ u_{\mathcal{O}} \\ u_{\mathcal{U}} \\ u_{\mathcal{Z}} \end{bmatrix}. \tag{10.55}$$

Directly from (10.55), we can infer that $\bar{R}_{\mathcal{O}\mathcal{Q}} = 0$ and $\bar{R}_{j\mathcal{Q}} = 0$ if $j \in \mathcal{O}$. For $j \in \mathcal{O}$, if Condition 10.2 and Condition 10.1 are satisfied, from (4.27) we have $\bar{R}_{j\ell} = (I - \breve{G}'_{jj})^{-1}\breve{R}'_{j\ell}$ where $\ell \in \mathcal{U} \cup \mathcal{Z}$, $\bar{R}_{jj} = (I - \breve{G}'_{jj})^{-1}$ and $\bar{R}_{j\ell} = 0$ where $\ell \in \mathcal{O}\backslash\{j\}$. From (4.20) and (4.21), we have $\breve{R}'_{j\mathcal{U}} = -\grave{H}_{j\mathcal{U}}$ and $\breve{R}'_{j\mathcal{Z}} = \breve{R}_{j\mathcal{Z}} - \grave{H}_{j\mathcal{U}}\breve{R}_{\mathcal{U}\mathcal{Z}}$ respectively. Owing to the fact that $\grave{H}_{j\mathcal{U}} = 0$ and $\breve{G}'_{jj} = \breve{G}_{jj}$, we have $\bar{R}_{j\mathcal{U}} = 0$, $\bar{R}_{jj} = (I - \breve{G}_{jj})^{-1}$ and $\bar{R}_{j\mathcal{Z}} = (I - \breve{G}_{jj})^{-1}\breve{R}_{j\mathcal{Z}}$.

For $j \in \mathcal{Q}$, from (4.34) we have $\bar{R}_{j\ell} = (I - \breve{G}''_{jj})^{-1}\breve{R}''_{j\ell}$ where $\ell \in \mathcal{O} \cup \mathcal{U} \cup \mathcal{Z}$, and from (4.33) we have $\bar{R}_{jj} = (I - \breve{G}''_{jj})^{-1}$ and $\bar{R}_{j\ell} = 0$ where $\ell \in \mathcal{Q}\backslash\{j\}$. If Condition 10.1 and 10.2 are satisfied, as a result of Lemma 10.3, we have $\breve{G}'_{j\mathcal{O}}\bar{R}_{\mathcal{O}\ell} = 0$. Therefore, from (4.28) we have $\breve{R}''_{j\ell} = 0$ if $\ell \in \mathcal{O}$, $\breve{R}''_{j\ell} = \breve{R}'_{j\ell}$ if $\ell \in \mathcal{U} \cup \mathcal{Z}$. From (4.20) and (4.21), we have $\breve{R}'_{j\mathcal{U}} = -\grave{H}_{j\mathcal{U}}$ and $\breve{R}'_{j\mathcal{Z}} = \breve{R}_{j\mathcal{Z}} - \grave{H}_{j\mathcal{U}}\breve{R}_{\mathcal{U}\mathcal{Z}}$ respectively. Owing to the fact that $\grave{H}_{j\mathcal{U}} = 0$ and $\breve{G}''_{jj} = \breve{G}_{jj}$, we have $\bar{R}_{j\ell} = 0$ if $\ell \in \mathcal{O} \cup \mathcal{U}$ and $\bar{R}_{j\ell} = (I - \breve{G}_{jj})^{-1}\breve{R}_{j\ell}$ if $\ell \in \mathcal{Z}$. This leads to the result in (10.52) - (10.54). ■

Lemma 10.5 *Consider the situation of Lemma 10.4. If Condition 10.3 is satisfied, then* $\bar{G}_{ji} = G_{ji}$, $\bar{G}_{jk} = \breve{G}_{jk} \forall k \in \mathcal{D}\backslash\{i\}$, $\bar{R}_{jj} = 1$ *and* $\bar{R}_{jk} = \breve{R}_{jk} \forall k \in \mathcal{Z}$.

Proof: With Lemma 10.4 it follows that $\bar{G}_{ji}$ is given by (10.51). For analysing the expression, we first are going to specify $\breve{G}_{ji}$ and $\breve{G}_{jj}$. From (4.8), we have $\breve{G}_{ji} = G_{ji} + G_{j\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}G_{\mathcal{Z}i}$ and $\breve{G}_{jj} = G_{jj} + G_{j\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}G_{\mathcal{Z}j}$. The first terms on the right hand sides reflect the direct connections from w_i to w_j (similarly w_j to w_j) and the second terms reflect the connections that pass only through nodes in $\mathcal{Z}$. By definition, $G_{jj} = 0$ since the G matrix in the network in (2.2) is hollow. Under Condition 10.3 being satisfied, the second terms on the right hand sides are zero, so that $\breve{G}_{ji} = G_{ji}$ and $\breve{G}_{jj} = 0$. Therefore from (10.51) - (10.54), we have $\bar{G}_{ji} = G_{ji}$, $\bar{G}_{jk} = \breve{G}_{jk}, k \in \mathcal{Q} \cup \mathcal{U}\backslash\{i\}$, $\bar{R}_{jj} = 1$ and $\bar{R}_{j\mathcal{Z}} = \breve{R}_{j\mathcal{Z}}$. ■

From the result of Lemma 10.5, we can obtain the result of Proposition 10.2.

10.C Proof of Proposition 10.3

We now provide the proof of proposition 10.3 which is based on conditions 10.1, Condition 10.2 and Condition 10.3 being satisfied.

Equation for w_j:
We know that $\breve{G}_{jk}$ is non-zero if there are unmeasured paths to w_j from w_k. Since $\mathcal{D}_j^w \subseteq \mathcal{D}\backslash\{j\}$ represents the node signals that has unmeasured paths to w_j, any transfer $\breve{G}_{jk}$ is zero if $k \notin \mathcal{D}_j^w$. Similarly, $\breve{R}_{jk}u_k = 0$ if $k \notin \mathcal{D}_j^u$. Now, considering the network equation in (10.2) which is the result of Proposition 10.1 and also using the result of Lemma 10.5, the equation of the output node of target module w_j can be given by,

$$w_j(t) \;=\; \sum_{k\in\mathcal{D}_j^w\backslash\{i\}} \breve{G}_{jk}w_k(t) + G_{ji}w_i(t) + \bar{H}_{jj}\xi_j(t) + u_j(t) + \sum_{k\in\mathcal{D}_j^u\backslash\{j\}} \breve{R}_{jk}u_k.$$

Pre-multiplying with $\bar{H}_{jj}^{-1}$ on both sides and keeping $w_j(t)$ on the left hand side we get,

$$w_j(t) = (1 - \bar{H}_{jj}^{-1})(w_j(t) - r_j(t)) + \bar{H}_{jj}^{-1}G_{ji}w_i(t)$$
$$+ \textstyle\sum_{k\in\mathcal{D}_j^w\backslash\{i\}} \bar{H}_{jj}^{-1}\breve{G}_{jk}w_k(t) + r_j(t) + \sum_{k\in\mathcal{D}_j^r\backslash\{j\}} \bar{H}_{jj}^{-1}\breve{R}_{jk}r_k + \xi_j(t).$$

Considering $S_j(q) = 1 - \bar{H}_{jj}^{-1}(q)$, $S_k(q) = \bar{H}_{jj}^{-1}(q)G_{jk}(q)$ for $k \in \mathcal{D}_j^w\backslash\{i\}$ and $S_{jk}(q) = \bar{H}_{jj}^{-1}(q)\breve{R}_{jk}(q)$ for $k \in \mathcal{D}_j^u\backslash\{j\}$, we get the result of (10.3) in the proposition.

Equation for $w_k, k \in \mathcal{Y}\backslash\{j\}$:
Considering the network equation in (10.50) which is the result of Proposition 4.2, the equation of $w_\mathcal{Y}$ in predictor form is given by,

$$w_\mathcal{Y} \;=\; (I - \bar{H}^{-1})w_\mathcal{Y} + \bar{H}^{-1}\sum_{\ell\in\mathcal{D}} \bar{G}_{\mathcal{Y}\ell}w_\ell + \bar{H}^{-1}\sum_{\ell\in\mathcal{Y}\cup\mathcal{U}\cup\mathcal{Z}} \bar{R}_{\mathcal{Y}\ell}u_\ell + \xi_\mathcal{Y}. \tag{10.56}$$

Therefore we can write the equation for node $w_k, k \in \mathcal{Y}\backslash\{j\}$ as,

$$w_k(t) = (1 - \bar{H}_{kk}^i)w_k(t) - \sum_{\ell\in\mathcal{Y}\backslash\{k\}} \bar{H}_{k\ell}^i w_\ell(t) +$$

$$\sum_{h\in\mathcal{Y}} \left[\sum_{\substack{\ell\in\mathcal{D}\\ h\neq\ell}} \bar{H}_{kh}^i \bar{G}_{h\ell}(q)w_\ell(t) + \sum_{\ell\in\mathcal{O}\cup\mathcal{U}\cup\mathcal{Z}\cup\{h\}} \bar{H}_{kh}^i \bar{R}_{h\ell}r_\ell(t) \right] + \xi_k(t). \tag{10.57}$$

Here, $\bar{H}^i_{h\ell}$ represent the $(h,\ell)^{th}$ element of the matrix $\bar{H}^{-1}$. The only product that is not strictly proper in (10.57) is $\bar{H}^i_{kk}\bar{R}_{kk}$. However, we can re-write it as $\bar{H}^i_{kk}\bar{R}_{kk} = \bar{H}^i_{kk}(1 + \bar{R}^{sp}_{kk})$ where $\bar{R}^{sp}_{kk}$ is a strictly proper transfer function. This leads to re-writing (10.57) as (10.4), where all the $B_\star$ predictor filters are strictly proper and hence the result of the proposition. □

10.D Proof of Proposition 10.4

Let us first consider the conditional distribution $p(w_m|w_{\bar{y}}, s, p, f)$. We first write,

$$\begin{aligned} w_m &= \mathbf{R}_{\backslash m} b + W_m b_m + u_m + \xi_m \,, & (10.58) \\ w_{\bar{y}} &= \mathbf{W}_{\mathcal{D}_{\backslash m}} g_{\backslash m} + \mathbf{W}_m g_m + \mathbf{W}_{ji} g_{ji} + u_{\bar{y}} + \xi_{\bar{y}} \,, & (10.59) \end{aligned}$$

where $\mathbf{W}_{\mathcal{D}_{\backslash m}}$ is constructed after excluding W_m in the matrix $\mathbf{W}_{\mathcal{D}}$ and $g_{\backslash m}$ is the vector constructed after excluding s_m and f_m in g. Here, $g_m = \begin{bmatrix} s_m^\top & f_m^\top \end{bmatrix}^\top$ and $\mathbf{W}_m = blkdiag(W_m, W_m)$. Grouping terms in (10.58) and (10.59) we get,

$$\begin{aligned} w_m &= \mu_3 + \bar{B}_m w_m + \xi_m \,, & (10.60) \\ w_{\bar{y}} &= \mu_1 + \mu_2 w_m + \xi_{\bar{y}} \,, & (10.61) \end{aligned}$$

where $\mu_1 = \mathbf{W}_{\mathcal{D}_{\backslash m}} g_{\backslash m} + \mathbf{W}_{ji} g_{ji} + u_{\bar{y}}$, $\mu_2 = \begin{bmatrix} \bar{S}_m \\ \bar{F}_m \end{bmatrix}$, $\mu_3 = \mathbf{R}_{\backslash m} p + u_m$ and $\Sigma_{\bar{y}} = blkdiag(\sigma_j^2 I_N, \sigma_a^2 I_N)$. By the law of conditional expectation and ignoring terms independent of w_m, we write

$$\begin{aligned} \log p(w_m|w_{\bar{y}}, b, s, f) &\cong \log p(w_{\bar{y}}, w_m|s, b, f) \\ &\cong -\frac{1}{2}|| \begin{bmatrix} w_{\bar{y}} \\ w_m \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_3 \end{bmatrix} - \begin{bmatrix} \mu_2 \\ \bar{B}_m \end{bmatrix} w_m ||^2_{\Sigma^{-1}} \\ &\cong -\frac{1}{2}||w_m||^2_{P_w^{-1}} + w_m^\top P_w^{-1} \mu_w \,, & (10.62) \end{aligned}$$

where $\bar{\mu}_1 = \begin{bmatrix} \mu_1 \\ \mu_3 \end{bmatrix}$, $\bar{\mu}_2 = \begin{bmatrix} \mu_2 \\ \bar{B}_m \end{bmatrix}$, $\Sigma^{-1} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix}$, $P_w = \left(\bar{\mu}_2^\top \Sigma^{-1} \bar{\mu}_2 + \Lambda_{22} - 2 \begin{bmatrix} \Lambda_{21} & \Lambda_{22} \end{bmatrix} \bar{\mu}_2 \right)^{-1}$, $\mu_w = P_w(\bar{\mu}_2^\top \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \end{bmatrix}^\top w_{\bar{y}} + \begin{bmatrix} \Lambda_{21} & \Lambda_{22} \end{bmatrix} \bar{\mu}_1 - \Lambda_{21} w_{\bar{y}} - \bar{\mu}_2^\top \Sigma^{-1} \bar{\mu}_1)$. The above log density is quadratic and represents a Gaussian distribution with covariance P_w and mean μ_w.

Let us now consider the conditional distribution $p(s|w_{\bar{y}}, w_m, b, f)$. By the law of conditional expectation and ignoring terms independent of s, we write

$$\begin{aligned} \log \quad & p(s|w_{\bar{y}}, w_m, b, f) \cong \log p(w_{\bar{y}}, w_m|s, b, f) + p(s) \\ \cong \quad & -\frac{1}{2\sigma_j^2}||w_{y} - \bar{\mu}_4 - \bar{\mathbf{W}} s||^2_{\Sigma^{-1}} - \frac{1}{2}||s||^2_{\mathbf{K}_1^{-1}} \,, \\ \cong \quad & -\frac{1}{2}||s||^2_{P_s^{-1}} + s^\top P_s^{-1} \mu_s \,, & (10.63) \end{aligned}$$

where $\bar{\mu}_4 = \begin{bmatrix} \mathbf{W}_{ji} \\ \mathbf{0}_N \end{bmatrix} g_{ji} + \bar{\mathbf{Q}}f + \bar{\mathbf{R}}b + \begin{bmatrix} u_{\bar{\mathcal{Y}}} \\ u_m \end{bmatrix}$, $\bar{\mathbf{W}} = \begin{bmatrix} \mathbf{W} \\ \mathbf{0}_N \\ \mathbf{0}_N \end{bmatrix}$, $\bar{\mathbf{Q}} = \begin{bmatrix} \mathbf{0}_N \\ \mathbf{Q} \\ \mathbf{0}_N \end{bmatrix}$, $\bar{\mathbf{R}} = \begin{bmatrix} \mathbf{0}_N \\ \mathbf{0}_N \\ \mathbf{R} \end{bmatrix}$, $P_s = \left(\mathbf{K}_1^{-1} + \bar{\mathbf{W}}^\top \Sigma^{-1} \bar{\mathbf{W}}\right)^{-1}$, $\mu_s = P_s \bar{\mathbf{W}}^\top \Sigma^{-1} \left(w_{\mathcal{Y}} - \bar{\mu}_4\right)$ and $\mathbf{0}_N$ is a zero matrix with N rows. The above log density is quadratic and represents a Gaussian distribution with covariance P_s and mean μ_s.

Let us now consider the conditional distribution $p(f|w_{\bar{\mathcal{Y}}}, w_m, b, s)$. By the law of conditional expectation and ignoring terms independent of s, we write

$$\begin{aligned} \log \quad & p(f|w_{\bar{\mathcal{Y}}}, w_m, b, s) \cong \log p(w_{\bar{\mathcal{Y}}}, w_m|s, b, f) + p(f) \\ \cong \quad & -\frac{1}{2\sigma_j^2}||w_{\mathcal{Y}} - \bar{\mu}_6 - \bar{\mathbf{Q}}f||^2_{\Sigma^{-1}} - \frac{1}{2}||f||^2_{\mathbf{K}_3^{-1}}\,, \\ \cong \quad & -\frac{1}{2}||f||^2_{P_f^{-1}} + f^\top P_f^{-1} \mu_f\,, \end{aligned} \tag{10.64}$$

where $\bar{\mu}_6 = \begin{bmatrix} \mathbf{W}_{ji} \\ \mathbf{0}_N \end{bmatrix} g_{ji} + \bar{\mathbf{W}}s + \bar{\mathbf{R}}b + \begin{bmatrix} u_{\bar{\mathcal{Y}}} \\ u_m \end{bmatrix}$, $P_f = (\mathbf{K}_3^{-1} + \bar{\mathbf{Q}}^\top \Sigma^{-1} \bar{\mathbf{Q}})^{-1}$, and $\mu_f = P_f \bar{\mathbf{Q}}^\top \Sigma^{-1} \left(w_{\mathcal{Y}} - \bar{\mu}_6\right)$. The above log density is quadratic and represents a Gaussian distribution with covariance P_f and mean μ_f.

Let us now consider the conditional distribution $p(b|w_{\bar{\mathcal{Y}}}, w_m, s, f)$. By the law of conditional expectation and ignoring terms independent of p, we write

$$\begin{aligned} \log p(b|w_{\bar{\mathcal{Y}}}, w_m, s, f) \quad & \cong \quad \log p(w_{\bar{\mathcal{Y}}}, w_m|s, b, f) + p(b) \\ & \cong \quad -\frac{1}{2}||w_{\mathcal{Y}} - \bar{\mu}_5 - \bar{\mathbf{R}}b||^2_{\Sigma_{\mathcal{Y}}^{-1}} - \frac{1}{2}||b||^2_{\mathbf{K}_2^{-1}} \\ & \cong \quad -\frac{1}{2}||p||^2_{P_p^{-1}} + p^\top P_p^{-1} \mu_p\,, \end{aligned} \tag{10.65}$$

where $\bar{\mu}_5 = \begin{bmatrix} \mathbf{W}_{\mathcal{D}} \\ \mathbf{0}_N \end{bmatrix} \begin{bmatrix} s \\ f \end{bmatrix} + \begin{bmatrix} \mathbf{W}_{ji} \\ \mathbf{0}_N \end{bmatrix} g_{ji} + \begin{bmatrix} u_{\bar{\mathcal{Y}}} \\ u_m \end{bmatrix}$, $P_b = (\mathbf{K}_2^{-1} + \bar{\mathbf{R}}^\top \Sigma^{-1} \bar{\mathbf{R}})^{-1}$, and $\mu_b = P_b \bar{\mathbf{R}}^\top \Sigma^{-1} \left(w_{\mathcal{Y}} - \bar{\mu}_5\right)$. The above log density is quadratic and represents a Gaussian distribution with covariance P_b and mean μ_b.

10.E Proof of Proposition 10.5

From (10.44) we have,

$$C \quad = \quad \arg\max \left[\sum_k {Q_s}_k^{(n)}(\lambda_k^s, \beta_k^s) + \frac{1}{M} \sum_{i=1}^{M} \log p(\bar{s}_m^{(i,n)}; \beta_m^s) \right] \tag{10.66}$$

with $k = \{j, c_1, \ldots, c_p, jk_1, \ldots, jk_p\}$ where $c_1, \ldots, c_p$ and $k_1, \ldots, k_p$ are the elements of the set $\mathcal{D}_j^w \backslash \{i, m\}$ and $\mathcal{D}_j^r \backslash \{j\}$ respectively, and

$$Q_{s_k}{}^{(n)}(\lambda_k^s, \beta_k^s) = \frac{1}{M} \sum_{i=1}^{M} \log p(\bar{s}_k^{(i,n)}; \lambda_k^s, \beta_k^s)$$

$$\cong \frac{1}{M}\sum_{i=1}^{M} -\log[\det(\lambda_k^s K_{\beta_k^s})] - \mathrm{tr}((\lambda_k^s K_{\beta_k^s})^{-1} \bar{s}_k^{(i,n)} \bar{s}_k^{\top(i,n)})$$

$$\cong -\log[\det(\lambda_k^s K_{\beta_k^s})] - \mathrm{tr}((\lambda_k^s K_{\beta_k^s})^{-1} \hat{\mathbf{S}}_k^{(n)}) - \hat{s}_k^{\top(n)} (\lambda_k^s K_{\beta_k^s})^{-1} \hat{s}_k^{(n)}. \quad (10.67)$$

Next, the proof follows the procedure used in [14]. We partially differentiate (10.67) with respect to λ_k^s and equate to zero to get the λ_k^{s*} expression. Substituting this λ_k^{s*} in (10.67) we get the expression for (10.44) using which we obtain $\hat{\beta}_k^{s(n+1)}$. Equation (10.46) is the expression of λ_k^{s*} after substituting $\hat{\beta}_k^{s(n+1)}$.

10.F Proof of Proposition 10.6

Considering $\arg\max \frac{1}{M}\sum_{i=1}^{M} \log p(\bar{s}_m^{(i,n)}; \beta_m^s)$ in (10.66) and expanding it as in (10.67) where $\lambda_m^s = 1$, we get the result of the proposition.

10.G Proof of Proposition 10.7

From (10.44) we have,

$$\begin{aligned} A &= \arg\min_{\theta,\sigma_j^2} \frac{1}{M}\sum_{i=1}^{M}\left[N\log\sigma_j^2 + \frac{1}{\sigma_j^2}\bar{\varepsilon}_j^{(i,n)\top}\bar{\varepsilon}_j^{(i,n)})\right] \\ &+ \arg\min_{\tilde{\Sigma}} \frac{1}{M}\sum_{i=1}^{M}\left[\sum_{t=1}^{N}\log\det(\tilde{\Sigma}) + \sum_{t=1}^{N}\mathrm{tr}(\tilde{\Sigma}^{-1}\bar{\varepsilon}^{(i,n)}(t)\bar{\varepsilon}^{(i,n)}(t)^\top)\right] \\ &= A_1 + A_2. \end{aligned}$$

We now write,

$$A_2 = \arg\min_{\tilde{\Sigma}}\left[\sum_{t=1}^{N}\log\det(\tilde{\Sigma}) + \sum_{t=1}^{N}\mathrm{tr}\left(\tilde{\Sigma}^{-1}\left[\hat{\varepsilon}^{(n)}(t)\hat{\varepsilon}^{(n)}(t)^\top + \hat{\mathbf{E}}^{(n)}(t)\right]\right)\right].$$

For the optimization problem A_2, we can follow the similar reasoning as the maximum likelihood proof in [3], which yields the result of the proposition for estimating parameters in $\tilde{\Sigma}$. This is done by differentiating the above cost function with respect to the elements of $\tilde{\Sigma}^{-1}$ and using the relations

$$\begin{aligned} \log\det\tilde{\Sigma} &= -\log\det\tilde{\Sigma}^{-1} \\ \det P &= \sum_i p_{ij}p^{ij} \\ (P^{-1})_{ij} &= p^{ij}/\det P \end{aligned} \quad (10.68)$$

where p^{ij} denotes the cofactor of the ij^{th} element p_{ij} of the matrix P.

Now considering A_1, we can write

$$A_1 = \arg\min_{\theta,\sigma_j^2}\left[N\log\sigma_j^2 + \frac{1}{\sigma_j^2}\frac{1}{M}\sum_{i=1}^{M}\bar{\varepsilon}_j^{(i,n)\top}\bar{\varepsilon}_j^{(i,n)}\right] \tag{10.69}$$

We notice that the optimum with respect to θ does not depend on the optimal value of σ_j^2. Then, we can first update θ and then use its optimal value to update σ_j^2. In order to find $\hat{\theta}^{(n)}$, σ_j^2 is fixed to $\hat{\sigma}_j^{2(n)}$ and substituted in Eq. (10.69). After substitution, the terms that are independent of θ can be removed from the objective function since it becomes a constant. Then we get,

$$\hat{\theta}^{(n+1)} = \arg\min_{\theta}\sum_{i=1}^{M}\bar{\varepsilon}_j^{(i,n)\top}\bar{\varepsilon}_j^{(i,n)} \tag{10.70}$$

Let us define $\breve{s}^{(i,n)} \in \mathbb{R}^N$ be a vector such that, if $N \leqslant l$, $\breve{s}^{(i,n)}$ is the vector of first N elements of $\bar{s}^{(i,n)}$ and if $N > l$, $\breve{s}^{(n)}$ is a vector with the first l elements equal to $\bar{s}^{(i,n)}$ and the remaining ones equal to 0. Let $\breve{S}^{(i,n)}$, $\breve{W}_i \in \mathbb{R}^{N\times N}$ be the Toeplitz matrix of $\hat{s}^{(n)}$ and $\bar{\mathbf{w}}_i$ respectively. Then

$$\bar{\mathcal{X}}^{(i,n)} = \begin{bmatrix}W_j - R_j & W_{c_1} & \dots & R_{k_p}\end{bmatrix}, \quad \bar{\mathcal{Y}}^{(i,n)} = \breve{S}^{(i,n)}\breve{W}_i.$$

We now re-write, $\bar{\mathbf{W}}^{(i,n)}\bar{s}^{(i,n)} = \bar{\mathcal{X}}^{(i,n)}\bar{s}^{(i,n)} + G_\theta\bar{W}_i\bar{s}_j^{(i,n)} = \bar{\mathcal{X}}^{(i,n)}\bar{s}^{(i,n)} + \bar{\mathcal{Y}}^{(i,n)}g_{ji}$. Therefore,

$$\begin{aligned}\hat{\theta}^{(n+1)} = \arg\min_{\theta}\sum_{i=1}^{M}\Big[& -2w_j^\top W_{ji}g_{ji} - 2w_j^\top\bar{\mathcal{Y}}^{(i,n)}g_{ji} \\ & + g_{ji}^\top W_{ji}^\top W_{ji}g_{ji} + 2g_{ji}^\top W_{ji}^\top r_j + 2g_{ji}^\top W_{ji}^\top\bar{\mathcal{X}}^{(i,n)}\bar{s}^{(i,n)} \\ & + 2g_{ji}^\top W_{ji}^\top\bar{\mathcal{Y}}^{(i,n)}g_{ji} + 2r_j^\top\bar{\mathcal{Y}}^{(i,n)}g_{ji} \\ & + 2\bar{s}^{(i,n)\top}\bar{\mathcal{X}}^{(i,n)\top}\bar{\mathcal{Y}}^{(i,n)}g_{ji} + g_{ji}^\top\bar{\mathcal{Y}}^{(i,n)\top}\bar{\mathcal{Y}}^{(i,n)}g_{ji}\Big]\end{aligned}$$

Defining

$$\hat{A}^{(n)} = \sum_{i=1}^{M}[W_{ji}^\top W_{ji} + 2W_{ji}^\top\bar{\mathcal{Y}}^{(i,n)} + \bar{\mathcal{Y}}^{(i,n)\top}\bar{\mathcal{Y}}^{(i,n)}],$$

and

$$\begin{aligned}\hat{b}^{(n)} = \sum_{i=1}^{M}\Big[& w_j^\top W_{ji} + w_j^\top\bar{\mathcal{Y}}^{(i,n)} - r_j^\top W_{ji} - \bar{s}^{(i,n)\top}\bar{\mathcal{X}}^{(i,n)\top}W_{ji} \\ & - r_j^\top\bar{\mathcal{Y}}^{(i,n)} - \bar{s}^{(i,n)\top}\bar{\mathcal{X}}^{(i,n)\top}\bar{\mathcal{Y}}^{(i,n)}\Big]^\top,\end{aligned}$$

we get the statement of the proposition for updating $\hat{\theta}^{(n+1)}$.

In order to find $\hat{\sigma}_j^{2(n)}$, θ is fixed to $\hat{\theta}^{(n+1)}$ and substituted in Eq. (10.69). After substitution, $A_1(\sigma_j^2, \hat{\theta}^{(n+1)})$ is differentiated w.r.t. σ_j^2 and equated to zero which leads to the result of the proposition.

11 CHAPTER

Conclusions and Future Outlook

11.1 Conclusions

The advancements in science and technology have made modern-day systems increasingly complex, large-scale, and interconnected. These systems comprise several sub-systems interconnected with each other, which can be modeled as a dynamic network. High-quality models of the dynamic behavior of these systems are required in many applications, like vehicle platooning, power networks. The advancements in sensor technology have increased the possibility of measuring a variety of relevant process variables and we would like to get high-quality models of the interconnected systems in the dynamic network using the measured variables, which resulted in the following research question:

How to effectively identify a module embedded in a dynamic network and obtain accurate estimates?

To answer the research question, several sub-questions were formulated by exploring the open problems in identifying a module in a dynamic network. In each chapter of this thesis, we addressed a specific sub-question that led to the contribution of this thesis.

11.1.1 The Local direct method

When using the direct method for network identification, confounding variables need to be properly handled to avoid bias in the estimate of our module dynamics. Correlated noise is an example of a direct confounding variable and having limited measured nodes can create an indirect confounding variable. In Chapter 4, we addressed the issue of confounding variables by developing the Local direct method, where we introduced the general theory for the construction of the predictor model to handle the confounding variables in the direct method of identification. Moving from a MISO predictor model to a MIMO predictor model and appropriate signal selection in the predictor model have been the key elements in the Local direct method to handle the confounding variables. The LDM is an asymptotically efficient identification method that can handle correlated noise networks. The estimates are consistent and asymptotically efficient. The variance of the estimates obtained with the Local direct method reaches the Cramér-Rao lower bound related to the particular chosen experimental setup. Using the provided general theory, numerous algorithms for predictor model selection can be formulated. Three algorithms were developed for use: A full input case algorithm where all node measurements are considered available; a minimum measurement case algorithm where limited node measurements are expected to be used; and a user selection case algorithm that provides a predictor model for the Local direct method based on the experimenter's input on the measured nodes.

Apart from the signal selection framework in Chapter 3, in order to guarantee consistency, data-informativity and the delay in path/loop condition need to be satisfied. Path-based conditions for data-informativity which can check data-informativity (in a generic sense) have been formulated in Chapter 5. However, based on the signal selection algorithm used, multiple experimental setups can be formulated using the local direct method for estimating the single target module. Therefore, the data-informativity conditions and the statistical properties of the target module estimates are different for different setups. The former necessitates the integration of path-based data-informativity conditions in the signal selection algorithms, and this integration will lead to selecting an experimental setup that provides least cost for sensors and actuators and guarantee consistent estimate of the target module. The latter necessitates the quantification of the variance of the target module since the minimum variance expressions, in the form of the related CRLB, will typically be different for the different experimental setups. Such quantification will lead to selecting an experimental setup that provides the consistent target module estimate with minimum variance. In case of no prior information on the delays in the network, it can be detected using works like [66] and be used to check the satisfaction of delay in path/loop condition. However, steps to avoid the delay conditions can also be made through modifications of predictors as in [142].

11.1.2 The Generalized method

The development of the Local direct method triggered the realization that all direct methods require the parallel path/loop condition to be satisfied, which requires certain nodes to be measured. This requirement is restrictive in many cases. The Generalized method in Chapter 4 has been the follow-up to answer the question for a consistent identification method with freedom in sensor selection and actuation. This is achieved by combining the features of both the direct method (using node signals as predictor inputs) and the indirect method (allowing post-processing of the estimates). The former allows us to utilize the excitation provided by the process noise and the latter allows us to relax the parallel path/loop condition. To get back the target module of interest using post-processing, there are conditions on having excitation signals exciting the parallel path/loop with no measured nodes and measurement of nodes that are descendant from that excited node. However, the measured nodes can be anywhere in the network.

11.1.3 Path based data-informativity conditions

The prediction error methods for local module identification in dynamic networks require data-informativity conditions (i.e. conditions for sufficient excitation of relevant dynamics) to be satisfied for consistent estimation. The condition is to have a positive definite spectrum of a vector of signals for a sufficient number of frequencies, where the vector of signals includes internal node signals which cannot be directly manipulated by the experimenter. In chapter 5, these conditions have been translated to path-based conditions (in a generic sense) that depend on paths from external signals to the internal node signals, which can be easily verified by the experimenter and implemented using graphical algorithms if the topology is known. This result paves the way to a synthesis problem of allocating excitation to guarantee data-informativity using graphical conditions.

11.1.4 Effective Algorithms

Most of the network identification methods, including the Local direct method and the Generalized method, typically require solving poorly scalable non-convex optimization problems and a model order selection step. The latter requires a large number of combinations to be solved in a large-dynamic network, which is computationally challenging and can affect the accuracy of your estimated dynamics. Dynamic networks are complex and large-scale and therefore they require effective algorithms in order to put the identification techniques into practice.

An effective algorithm for full network identification based on the application of least squares estimations in multiple steps has been introduced in Chapter 6. The algorithm does not require the noise topology (i.e., the correlation structure of the process noises) to be known and has an integrated topology estimation procedure. However, the algorithm can also encode known topology as well. The algorithm provides consistent estimates while avoiding local minima. Relying only on least-squares solutions and with parallel computation capabilities, the algorithm is expected to scale well to large-sized networks. The algorithm is also suitable for identifying a part of the network or a single module in the network.

Effective algorithms that avoid model order selection for nuisance modules and provide reduced MSE estimates in the local module identification problem are introduced from Chapter 7 to Chapter 10. These algorithms are suitable when the data records are limited. The nuisance modules are modeled as zero-mean Gaussian processes with the covariance matrix (kernel) given by the first-order stable spline kernel, which is described by only two hyperparameters and also encodes stability and smoothness of the processes. Keeping a parametric model for the module that needs to be identified, an Empirical Bayes approach is used to find the parameters of the target module and the kernel hyperparameters. For solving the related optimization problem, an Expectation-Maximization scheme that has substantially reduced computational complexity is used. Chapter 7 and Chapter 8 provide the algorithm for local module identification where it is required to solve a MISO and MIMO estimation problem respectively. Chapter 9 considers the situation of the unknown topology of the network and provides a complete algorithm to estimate a single module that also incorporates topology estimation using non-causal Wiener filter approaches. Chapter 10 considers the situation where some of the nodes that are required for estimation may not be measurable, due to sensor selection and placement issues. These nodes may be the node measurements required to satisfy parallel path/loop conditions or the nodes required to handle confounding variables. The algorithm provided in Chapter 10 reconstructs these missing node observations by combining the above kernel-based approach with approximate inference methods.

Chapter 4 and Chapter 10 provide methods for single module identification under the presence of non-measured nodes. However, full-network identification and topology identification under non-measured nodes is an interesting problem that has not received much attention. The method in Chapter 6 assumes that the measurements of all nodes are available. Also, topology identification methods based on direct approach frameworks like [116] require all nodes in the network to be measured. There are topology identification approaches that exploits indirect approaches like [13, 152] where the topology of the network is obtained by estimating transfers between external excitations and the different nodes, and then by back-computing the transfer between the nodes. Exploiting algorithms using the indirect method, generalized method, and the method in Chapter 10 using the node reconstruction method could be a possible direction to solve the topology detection and full network identification problem under non-measured node signals.

11.1.5 Reflecting back on the research question

This thesis first provides a theory for signal selection and building a predictor model to identify a module in a dynamic network using PEM. This also includes conditions to check data informativity. The signal selection and data-informativity conditions are graphical conditions of the network. This paves the way to automate the signal selection and excitation allocation procedure using graph-based algorithms that are scalable to large-scale networks. Then, building upon the theory, we developed algorithms for module estimation that are scalable to large networks. The contributions of each chapter in this thesis are the pieces of the puzzle to answer the research question of how to effectively estimate a module in a dynamic network and obtain accurate estimates. Assembling the puzzle pieces (i.e. the contributions of each chapter) leads to the decision flow-chart in Figure 11.1, which guides the user to learn a module in a dynamic network effectively.

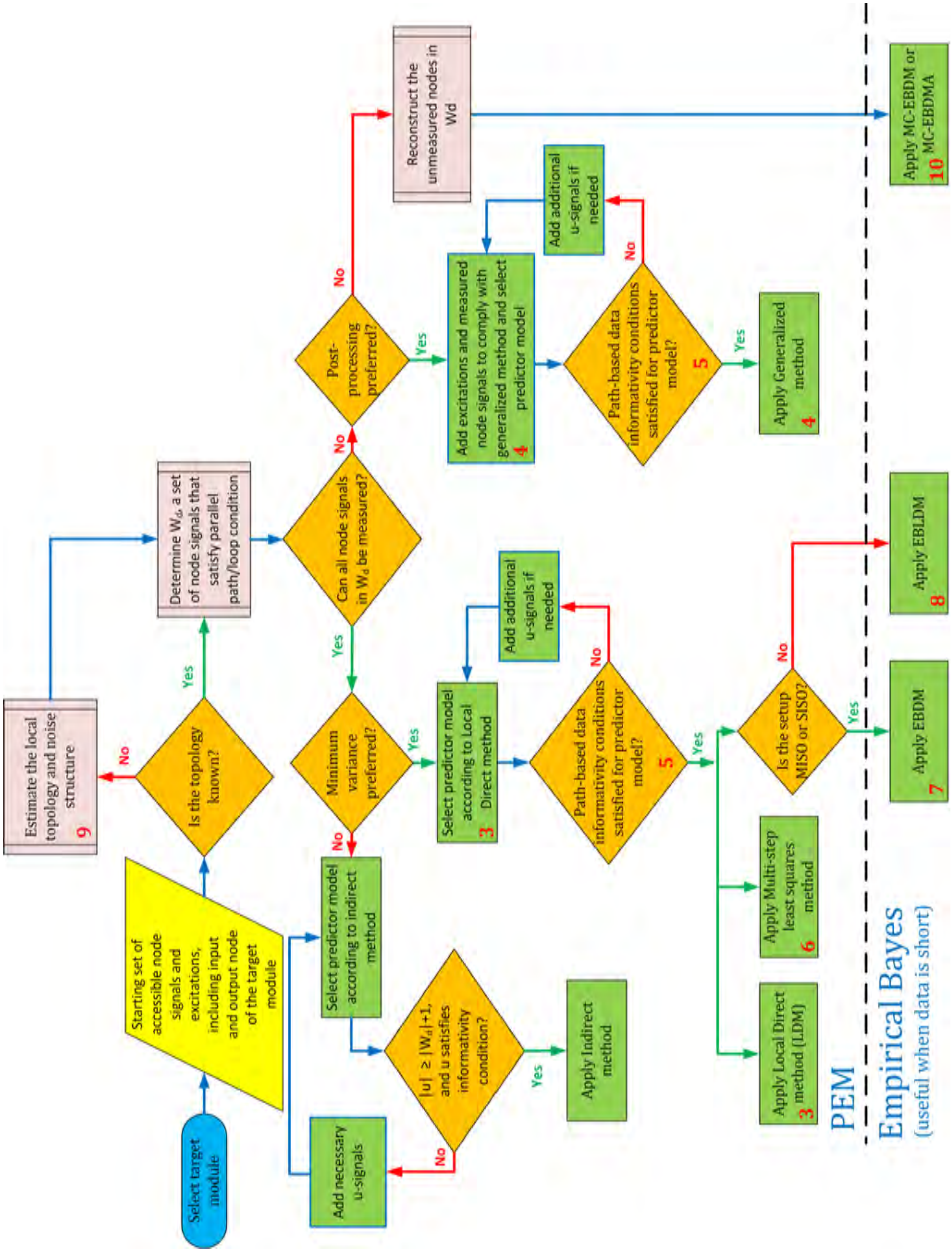

Select target module
Starting set of accessible node signals and excitations, including input and output node of the target module
Is the topology known?
No
Estimate the local topology and noise structure
9
Yes
Determine W_d a set of node signals that satisfy parallel path/loop condition
Can all node signals in W_d be measured?
Yes
No
Minimum variance preferred?
No
Select predictor model according to indirect method
$|u| \geq |W_d|+1$, and u satisfies informativity condition?
Yes
Apply Indirect method
No
Add necessary u-signals
Yes
Select predictor model according to Local Direct method
3
Path-based data informativity conditions satisfied for predictor model?
5
No
Add additional u-signals if needed
Yes
Apply Local Direct method (LDM)
3
Apply Multi-step least squares method
6
Is the setup MISO or SISO?
Yes
Apply EBDM
7
No
Apply EBLDM
8
Post-processing preferred?
No
Reconstruct the unmeasured nodes in Wd
Apply MC-EBDM or MC-EBDMA
10
Yes
Add excitations and measured node signals to comply with generalized method and select predictor model
4
Path-based data informativity conditions satisfied for predictor model?
5
No
Add additional u-signals if needed
Yes
Apply Generalized method
4
PEM
Empirical Bayes
(useful when data is short)

Figure 11.1: A guide for learning modules in a dynamic network. The number in red specifies the chapter number of this thesis that contributes to the decision chart.

11.2 Recommendations for future research

11.2.1 Low hanging fruits

- In this thesis we focussed on identifying a module in a dynamic network and built a solid theory for predictor model selection and effective algorithms that are scalable. Based on the results in this thesis, extensions to the situation where a set of modules in a network are chosen as the target modules of interest can be possibly be developed.

- For single module identification, we have explored the framework of immersion [29] to remove the unmeasured nodes from the network. But a more general theory of abstraction has been presented in [141]. If the abstraction principle is used, the module variance conditions change. Hence, the parallel path/loop condition is no more needed and can be replaced by a different set of conditions [141]. However, using the abstraction principle, we might need to model non-proper modules and handle confounding variables. In this case, the theory of handling confounding variables can be extended. Effective algorithms using regularized kernel-based methods can be developed that can handle non-proper modules and also offers the advantages mentioned in this thesis.

- The generalized method discussed in this thesis considers the input and output of the target module to be measured. However, this requirement is not strict. The generalized method can also be extended to consider the situations where either the input or output of the target module is not measured or both the input and output are not measured. This step might help us in understanding the relation between identifiability and identification in dynamic networks. This was indicative from the identifiability results in [117].

- This thesis deals with dynamic networks with only process noise and no sensor noise. Sensor noise does not enter the network and affect the dynamics but only affects the measurements of the node signals. There are effective algorithms using regularized kernel-based methods by extending the framework of indirect methods [38] that handle networks with sensor noise. However, a network with sensor noise can be translated to a network with correlated process noise [126]. Having a framework to handle correlated process noise in Chapter 3, direct identification methods for a dynamic network with both sensor and process noise is a possible extension that can be made.

11.2.2 Nonlinear dynamics

Linear dynamic networks have been treated in this thesis. Since many real-world systems are non-linear, it is vital to model the non-linear phenomenon in dynamic networks. The viable next step to venture into the non-linear world is by including non-linear models for modules by adding static non-linearities along with linear models. Effective algorithms integrating machine learning techniques into the identification of SISO Wiener systems [110] and Hammerstein systems [109] are available, which can be possibly extended to dynamic networks. However, this area is new and needs to be explored to come up with identification methods, theory for signal selection, data informativity, identifiability, etc. Similarly, dynamic network with switching topology or modules is an interesting framework that needs attention.

11.2.3 Optimal signal selection

The results of Chapter 3 provide multiple predictor models that can guarantee maximum likelihood properties. Similarly, Chapter 4 can provide different predictor models that can guarantee consistency. This raises the question of which predictor model needs to be chosen i.e. the optimal signal selection problem. This requires solving an optimization problem that minimizes a cost function. A relatively easy selection scheme is to minimize the cost for sensors through an optimization problem. However, selecting signals that provide us with an estimate of the target module(s) with the least variance would be a problem for future scope. This requires developing a theory for variance quantification in dynamic networks.

11.2.4 Data informativity and experiment design

Path-based data informativity conditions have been derived in this thesis. However, these are generic conditions i.e. do not take the numeric values of the transfer functions into account. Recently, steps are being taken to include the model orders into account for data informativity. In [12], extensions have been made to provide necessary and sufficient conditions for data informativity (a condition that takes model order into account). Looking into data informativity raises the question of optimal experiment design like how to design excitation with the least excitation power in order to get accurate estimates. This also involves selecting the location of excitation as well.

Bibliography

[1] K. Abed-Meraim, W. Qiu, and Y. Hua. Blind system identification. *Proceedings of the IEEE*, 85(8):1310–1322, 1997.

[2] B. D. O. Anderson and J. B. Moore. *Optimal filtering*. Englewood Cliffs, N.J., USA: Prentice-Hall, 1979.

[3] K. J. Åström. Maximum likelihood and prediction error methods. *Automatica*, 16(5):551 – 574, 1980.

[4] F. Bach, R. Jenatton, J. Mairal, G. Obozinski, et al. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, 5:19–53, 2011.

[5] J. S. Bailly, P. Monestiez, and P. Lagacherie. Modelling spatial variability along drainage networks with geostatistics. *Mathematical Geology*, 38:515–539, 2006.

[6] A. S. Bazanella, M. Gevers, and J. M. Hendrickx. Network identification with partial excitation and measurement. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 5500–5506, 2019.

[7] A. S. Bazanella, M. Gevers, J. M. Hendrickx, and A. Parraga. Identifiability of dynamical networks: which nodes need to be measured? In *Proc. 56th IEEE Conference on Decision and Control (CDC)*, pages 5870–5875, 2017.

[8] M. J. Beal. *Variational algorithms for approximate Bayesian inference*. PhD dissertation, Gatsby Computational Neuroscience Unit, University College London, 2003.

[9] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, second edition, 1985.

[10] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[11] A. Bolstad, B.D. Van Veen, and R. Nowak. Causal network inference via group sparse regularization. *IEEE Transactions on Signal Processing*, 59(6):2628–2641, 2011.

[12] X. Bombois, K. Colin, P.M.J. Van den Hof, and H. Hjalmarsson. On the informativity of direct identification experiments in dynamical networks. *Automatica*, 2021. hal-03329586. Submitted for publication.

[13] X. Bombois and H. Hjalmarsson. Network topology detection via uncertainty analysis of an identified static model. *IFAC-PapersOnLine*, 54(7):595–600, 2021. 19th IFAC Symposium on System Identification SYSID 2021.

[14] G. Bottegal, A. Y. Aravkin, H. Hjalmarsson, and G. Pillonetto. Robust EM kernel-based methods for linear system identification. *Automatica*, 67:114–126, 2016.

[15] S. Boyd, N. Parikh, and E. Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[16] R. A. Boyles. On the convergence of the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 45(1):47–50, 1983.

[17] G. Camba-Méndez and G. Kapetanios. Statistical tests and estimators of the rank of a matrix and their applications in econometric modelling. *Econometrics Reviews*, 28(6):581–611, 2009.

[18] W. Cao, G. Picci, and A. Lindquist. Identification of low rank vector processes, 2021. arXiv:2111.10899 [eess.SY].

[19] T. Chen. On kernel design for regularized lti system identification. *Automatica*, 90:109–122, 2018.

[20] T. Chen and L. Ljung. Constructive state space model induced kernels for regularized system identification. *IFAC Proceedings Volumes*, 47(3):1047–1052, 2014. 19th IFAC World Congress.

[21] T. Chen, H. Ohlsson, and L. Ljung. On the estimation of transfer functions, regularizations and gaussian processes - revisited. *Automatica*, 48(8):1525–1535, 2012.

[22] X. Cheng, S. Shi, and P. M.J. Van den Hof. Allocation of excitation signals for generic identifiability of linear dynamic networks. *IEEE Transactions on Automatic Control*, 67(2), 2022. to appear.

[23] A. Chiuso and G. Pillonetto. A Bayesian approach to sparse dynamic network identification. *Automatica*, 48(8):1553—–1565, 2012.

[24] J. Chow and P. Kokotovic. Time scale modeling of sparse dynamic networks. *IEEE Transactions on Automatic Control*, 30(8):714–722, 1985.

[25] R. D'Andrea and G.E. Dullerud. Distributed control design for spatially interconnected systems. *IEEE Transactions on Automatic Control*, 48(9):1478–1495, 2003.

[26] A. G. Dankers. *System identification in dynamic networks*. PhD dissertation, Delft University of Technology, 2014.

[27] A. G. Dankers, P. M. J. Van den Hof, X. Bombois, and P. S. C. Heuberger. Errors-in-variables identification in dynamic networks – consistency results for an instrumental variable approach. *Automatica*, 62:39–50, 2015.

[28] A. G. Dankers, P. M. J. Van den Hof, P. S. C. Heuberger, and X. Bombois. Dynamic network structure identification with prediction error methods - basic examples. In J. Schoukens and R. Bitmead, editors, *Proc. 16th IFAC Symposium on System Identification, Brussels, Belgium*, pages 876–881, 2012.

[29] A. G. Dankers, P. M. J. Van den Hof, P. S. C. Heuberger, and X. Bombois. Identification of dynamic models in complex networks with prediction error methods: Predictor input selection. *IEEE Trans. on Automatic Control*, 61(4):937–952, 2016.

[30] A. G. Dankers, P. M. J. Van den Hof, D. Materassi, and H. H. M. Weerts. Conditions for handling confounding variables in dynamic networks. *IFAC-PapersOnLine*, 50(1):3983–3988, 2017. Proc. 20th IFAC World Congress.

[31] A.G. Dankers. Optimization method for obtaining estimates in a dynamic network. Technical note, 2019.

[32] M. Deistler, B.D.O. Anderson, A. Filler, Ch. Zinner, and W. Chen. Generalized linear dynamic factor models: An approach via singular autoregressions. *European Journal of Control*, 16(3):211–224, 2010.

[33] M. Deistler, W. Scherrer, and B. D. O. Anderson. The structure of generalized linear dynamic factor models. *Empirical Economic and Financial Research*, 2015. Beran, J. and Feng, Y. and Hebbel, H. (eds.).

[34] D. del Vecchio, A. J. Ninfa, and E. D. Sontag. Modular cell biology: retroactivity and insulation. *Molecular Systems Biology*, 4(1):161, 2008.

[35] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[36] R. Diestel. *Graph theory*, volume 173. Graduate texts in mathematics, 3rd edition, 2005.

[37] M. Dimovska and D. Materassi. Granger-causality meets causal inference in graphical models: Learning networks via non-invasive observations. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 5268–5273, 2017.

[38] N. Everitt, G. Bottegal, and H. Hjalmarsson. An empirical bayes approach to identification of modules in dynamic networks. *Automatica*, 91:144–151, 2018.

[39] N. Everitt, G. Bottegal, C. R. Rojas, and H. Hjalmarrsson. On the effect of noise correlation in parameter identification of SIMO systems. In *Proc. 17th IFAC Symposium on System Identification (SYSID2015)*, pages 326–331, 2015.

[40] N. Everitt, G. Bottegal, C. R. Rojas, and H. Hjalmarrsson. Variance analysis of linear SIMO models with spatially correlated noise. *Automatica*, 77:68–81, 2017.

[41] N. Everitt, H. Hjalmarsson, and C. Rojas. A geometric approach to variance analysis of cascaded systems. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, page 6496 – 6501, Florence, Italy, 2013.

[42] N. Everitt, H. Hjalmarsson, and C. Rojas. Variance results for parallel cascade serial systems. In *Proceedings of 19th IFAC World Congress*, Cape Town, South Africa, 2014.

[43] S. J. M. Fonken, M. Ferizbegovic, and H. Hjalmarsson. Consistent identification of dynamic networks subject to white noise using weighted null-space fitting. *IFAC-PapersOnLine*, 53(2):46–51, 2020. 21th IFAC World Congress.

[44] S. J. M. Fonken, K. R. Ramaswamy, and P. M. J. Van den Hof. A scalable multi-step least squares method for network identification with unknown disturbance topology. *Automatica*, July 2022. ArXiv: 2106.07548. To appear.

[45] U. Forssell and L. Ljung. Identification of unstable systems using Output Error and Box-Jenkins model structures. In *Proc. 37th IEEE Conf. on Decision and Control (CDC)*, pages 3932–3937, Tampa, FL, USA, 1998. IEEE.

[46] K. J. Friston. Functional and effective connectivity: a review. *Brain connectivity*, 1(1):13–36, 2011.

[47] M. Galrinho, N. Everitt, and H. Hjalmarsson. ARX modeling of unstable linear systems. *Automatica*, 75:167–171, 2017.

[48] M. Galrinho, N. Everitt, and H. Hjalmarsson. Incorporating noise modeling in dynamic networks using non-parametric models. *IFAC-PapersOnLine*, 50(1):10568–10573, 2017. 20th IFAC World Congress.

[49] M. Galrinho, R. Prota, M. Ferizbegovic, and H. Hjalmarsson. Weighted null-space fitting for identification of cascade networks. *IFAC-PapersOnLine*, 51(15):856–861, 2018. 18th IFAC Symposium on System Identification SYSID 2018.

[50] M. Galrinho, C. R. Rojas, and H. Hjalmarsson. A weighted least squares method for estimation of unstable systems. In *Proc. 55th IEEE Conf. on Decision and Control (CDC)*, pages 341–346, Las Vegas, NV, USA, 2016. IEEE.

[51] M. Galrinho, C. R. Rojas, and H. Hjalmarsson. Parametric identification using weighted null-space fitting. *IEEE Transactions on Automatic Control*, 64(7):2798–2813, 2019.

[52] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[53] M. Gevers and A. S. Bazanella. Identification in dynamic networks: Identifiability and experiment design issues. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 4005–4010, 2015.

[54] M. Gevers, A.S. Bazanella, and G. Vian da Silva. A practical method for the consistent identification of a module in a dynamical network. *IFAC-PapersOnLine*, 51(15):862–867, 2018. 18th IFAC Symposium on System Identification SYSID 2018.

[55] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall London, 1996.

[56] J. Goncalves, R. Howes, and S. Warnick. Dynamical structure functions for the reverse engineering of LTI networks. In *2007 46th IEEE Conference on Decision and Control*, pages 1516–1522, 2007.

[57] J. Gonçalves and S. Warnick. Necessary and sufficient conditions for dynamical structure reconstruction of LTI networks. *IEEE Trans. Automatic Control*, 53(7):1670–1674, August 2008.

[58] R. D. Gudi and J. B. Rawlings. Identification for decentralized model predictive control. *AIChE Journal*, 52(6):2198–2210, 2006.

[59] B. Günes, A. G. Dankers, and P. M. J. Van den Hof. A variance reduction for identification in dynamic networks. *IFAC-PapersOnLine*, 47(3):2842–2847, 2014. 19th IFAC World Congress.

[60] A. Haber and M. Verhaegen. Subspace identication of large-scale interconnected systems. *IEEE Transactions on Automatic Control*, 59(10):2754–2759, 2014.

[61] K. Hajian-Tilaki. Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation. *Caspian Journal of Internal Medicine*, 4(2):627, 2013.

[62] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97 – 109, 1953.

[63] D. Hayden, Y. H. Chang, J. Goncalves, and C. J. Tomlin. Sparse network identifiability via compressed sensing. *Automatica*, 68:9–17, 2016.

[64] M. Hecker, S. Lambeck, S. Toepfer, E. van Someren, and R. Guthke. Gene regulatory network inference: Data integration in dynamic models—a review. *Biosystems*, 96(1):86–103, 2009.

[65] J.M. Hendrickx, M. Gevers, and A.S. Bazanella. Identifiability of dynamical networks with partial node measurements. *IEEE Trans. Autom. Control*, 64(6):2240–2253, 2019.

[66] S. Jahandari and D. Materassi. Sufficient and necessary graphical conditions for miso identification in networks with observational data. *IEEE Transactions on Automatic Control*, pages 1–1, 2021.

[67] M. Jamshidi. *Large-Scale Systems: Modeling and Control*. North-Holland, New York, 1983.

[68] C. S. Jensen, U. Kjærulff, and A. Kong. Blocking gibbs sampling in very large probabilistic expert systems. *International Journal of Human-Computer Studies*, 42(6):647–666, 1995.

[69] K. E. Johnson and N. Thomas. Wind farm control: Addressing the aerodynamic interaction among wind turbines. In *2009 American Control Conference*, pages 2104–2109, 2009.

[70] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, New Jersey, 2000.

[71] R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

[72] S. Kathari and A. K. Tangirala. Efficient reconstruction of granger-causal networks in linear multivariable dynamical processes. *Industrial & Engineering Chemistry Research*, 58(26):11275–11294, 2019.

[73] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.

[74] D. Kroening and O. Strichmann. *Decision Procedures - An algorithmic point of view*. Springer, 2 edition, 2016.

[75] P. Kundur. *Power System Stability and Control*. EPRI Power System Engineering Series. McGraw-Hill Inc., 1994.

[76] K. Y. Liang, J. Mårtensson, and K. H. Johansson. Heavy-duty vehicle platoon formation for fuel efficiency. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1051–1061, 2016.

[77] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, NJ, 1999.

[78] L. Ljung and B. Wahlberg. Asymptotic properties of the least-squares method for estimating transfer functions and disturbance spectra. *Advances in Applied Probability*, 24(2):412–440, 1992.

[79] M. Mansoori, A. Dankers, and P.M.J Van den Hof. Errors-in-variables identification in bilaterally coupled systems with application to oil well testing. *IFAC Proceedings Volumes*, 47(3):4656–4661, 2014.

[80] J. S. Maritz and T. Lwin. *Empirical Bayes Methods*. Chapman and Hall, 1989.

[81] D. Materassi and G. Innocenti. Unveiling the connectivity structure of financial networks via high-frequency analysis. *Physica A: Statistical Mechanics and its Applications*, 388(18):3866–3878, 2009.

[82] D. Materassi and G. Innocenti. Topological identification in networks of dynamical systems. *IEEE Trans. Automatic Control*, 55(8):1860–1871, 2010.

[83] D. Materassi and M. Salapaka. Identification of network components in presence of unobserved nodes. In *Proc. 2015 IEEE 54th Conf. Decision and Control, Osaka, Japan*, pages 1563–1568, 2015.

[84] D. Materassi and M. V. Salapaka. Signal selection for estimation and identification in networks of dynamic systems: a graphical model approach. *IEEE Trans. Automatic Control*, 65(10):4138–4153, October 2020.

[85] D. Materassi and M.V. Salapaka. On the problem of reconstructing an unknown topology via locality properties of the Wiener filter. *IEEE Trans. Automatic Control*, 57(7):1765–1777, 2012.

[86] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087 – 1092, 1953.

[87] R. M. Neal. Slice sampling. *Annals of Statistics*, 31:705–767, 2003.

[88] B. Ø. Palsson. *Systems Biology: Simulation of Dynamic Network States*. Cambridge University Press, 2011.

[89] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[90] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, 2000.

[91] G. Pillonetto, A. Chiuso, and G. De Nicolao. Prediction error identification of linear systems: a nonparametric gaussian regression approach. *Automatica*, 47(2):291–305, 2011.

[92] G. Pillonetto and G. De Nicolao. A new kernel-based approach for linear system identification. *Automatica*, 46(1):81–93, 2010.

[93] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung. Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3):657–682, 2014.

[94] G. A. Pimentel, R. de Vasconcelos, A. Salton, and A. Bazanella. Network topology impact on the identification of dynamic network models with application to autonomous vehicle platooning. *IFAC-PapersOnLine*, 53(2):1031–1036, 2020. 21st IFAC World Congress.

[95] R. Pintelon and J. Schoukens. *System identification: a frequency domain approach*. John Wiley & Sons, 2012.

[96] J. G. Proakis and M. Salehi. *Communication Systems Engineering*. Pearson Education, 2nd edition edition, 2001.

[97] C. J. Quinn, T. P. Coleman, N. Kiyavash, and N. G. Hatsopoulos. Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. *J Comput Neurosci.*, 30(1):17–44, 2011.

[98] V. C. Rajagopal. Learning local modules in dynamic networks without prior topology information. Master's thesis, Eindhoven University of Technology, October 2020.

[99] V. C. Rajagopal, K. R. Ramaswamy, and P. M. J. Van den Hof. A regularized kernel-based method for learning a module in a dynamic network with correlated noise. In *Proc. 59th IEEE Conf. on Decision and Control (CDC)*, pages 4348–4353, Jeju Island, Korea, 2020.

[100] V. C. Rajagopal, K. R. Ramaswamy, and P. M. J. Van den Hof. Learning local modules in dynamic networks without prior topology information. In *Proc. 60th IEEE Conf. on Decision and Control (CDC)*, pages 840–845, Austin, TX, USA, 2021.

[101] K. R. Ramaswamy, G. Bottegal, and P. M. J. Van den Hof. Local module identification in dynamic networks using regularized kernel-based methods. In *Proc. 57th IEEE Conf. on Decision and Control (CDC)*, pages 4713–4718, Miami Beach, FL, 2018.

[102] K. R. Ramaswamy, G. Bottegal, and P. M. J. Van den Hof. Learning linear models in a dynamic network using regularized kernel-based methods. *Automatica*, 129(109591), 2021.

[103] K. R. Ramaswamy, P. Z. Csurcsia, J. Schoukens, and P. M. J. Van den Hof. A frequency domain approach for local module identification in dynamic networks. *Automatica*, November 2022. ArXiv: 2105.10901. To appear.

[104] K. R. Ramaswamy and P. M. J. Van den Hof. A local direct method for module identification in dynamic networks with correlated noise. *IEEE Trans. Automatic Control*, 66(11):5237–5252, 2021.

[105] K. R. Ramaswamy, P. M. J. Van den Hof, and A. G. Dankers. Generalized sensing and actuation schemes for local module identification in dynamic networks. In *Proc. 58th IEEE Conf. on Decision and Control (CDC)*, pages 5519–5524, Nice, France, 2019. IEEE.

[106] K.R. Ramaswamy, R.M. Fonseca, O. Leeuwenburgh, M. M. Siraj, and P. M. J. Van den Hof. Improved sampling strategies for ensemble-based optimization. *Computational Geosciences*, 24:1057–1069, 2020.

[107] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. The MIT Press, 2006.

[108] W. Ren and R.W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer, 2008.

[109] R.S. Risuleo, G. Bottegal, and H. Hjalmarsson. A nonparametric kernel-based approach to hammerstein system identification. *Automatica*, 85:234–247, 2017.

[110] R.S. Risuleo, F. Lindsten, and H. Hjalmarsson. Bayesian nonparametric identification of wiener systems. *Automatica*, 108:108480, 2019.

[111] T.C.J. Romijn, M.C.F. Donkers, J.T.B.A. Kessels, and S. Weiland. A distributed optimization approach for complete vehicle energy management. *IEEE Transactions on Control Systems Technology*, 27(3):964–980, 2019.

[112] B. M. Sanandaji, T. L. Vincent, and M. B. Wakin. Exact topology identification of large-scale interconnected dynamical systems from compressive observations. In *Proc. American Control Conference (ACC)*, pages 649–656, San Francisco, CA, USA, 2011.

[113] T. B Schön, F. Lindsten, J. Dahlin, J. Wågberg, A. C. Naesseth, A. Svensson, and L. Dai. Sequential Monte Carlo methods for system identification. In *Proc. of the 17th IFAC Symposium on System identification*, pages 775–786, 2015.

[114] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

[115] S. Shi. *Topological Aspects of Linear Dynamic Networks: Identifiability and Identification.* PhD dissertation, Eindhoven University of Technology, 2020.

[116] S. Shi, G. Bottegal, and P. M. J. Van den Hof. Bayesian topology identification of linear dynamic networks. In *Proc. 18th European Control Conference*, pages 2814–2819, 2019.

[117] S. Shi, X. Cheng, and P. M. J. Van den Hof. Single module identifiability in linear dynamic networks with partial excitation and measurement. *IEEE Trans. on Automatic Control*, 2021. ArXiv: 2012.11414. Provisionally accepted for publication.

[118] S. Shi, X. Cheng, and P. M. J. Van den Hof. Generic identifiability of subnetworks in a linear dynamic network: The full measurement case. *Automatica*, 137:110093, 2022.

[119] P. Skudlarski, K. Jagannathan, V. D Calhoun, M. Hampson, B. A. Skudlarska, and G. Pearlson. Measuring brain connectivity: diffusion tensor imaging validates resting state temporal correlations. *Neuroimage*, 43(3):554–561, 2008.

[120] T. Söderström and P. Stoica. *System Identification.* Prentice-Hall International, Hemel Hempstead, UK, 1989.

[121] M. Soleimanzadeh and R. Wisniewski. Controller design for a wind farm, considering both power and load aspects. *Mechatronics*, 21(4):720–727, 2011.

[122] Saurav Talukdar, Deepjyoti Deka, Blake Lundstrom, Michael Chertkov, and Murti V. Salapaka. Learning exact topology of a loopy power grid from ambient dynamics. In *Proceedings of the Eighth International Conference on Future Energy Systems*, e-Energy '17, page 222–227, New York, NY, USA, 2017. Association for Computing Machinery.

[123] M. A. Tanner and W. H. Wong. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540, 1987.

[124] P. M. J. Van den Hof, A. G. Dankers, P. S. C. Heuberger, and X. Bombois. Identification of dynamic models in complex networks with prediction error methods - basic methods for consistent module estimates. *Automatica*, 49(10):2994–3006, 2013.

[125] P. M. J. Van den Hof, A. G. Dankers, and Weerts H. H. M. Identification in dynamic networks. *Computers & Chemical Engineering*, 109:23–29, January 2018.

[126] P. M. J. Van den Hof, A. G. Dankers, and H. H. M. Weerts. From closed-loop identification to dynamic networks: generalization of the direct method. In *Proc. 56nd IEEE Conf. on Decision and Control (CDC)*, pages 5845–5850, Melbourne, Australia, 2017. IEEE.

[127] P. M. J. Van den Hof and K. R. Ramaswamy. Path-based data-informativity conditions for single module identification in dynamic networks. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 4354–4359, Jeju Island, Republic of Korea, 2020.

[128] P. M. J. Van den Hof, K. R. Ramaswamy, A. G. Dankers, and G. Bottegal. Local module identification in dynamic networks with correlated noise: the full input case. In *Proc. 58th IEEE Conf. on Decision and Control (CDC)*, pages 5494–5499, Nice, France, 2019.

[129] P. M.J. Van den Hof and K.R. Ramaswamy. Learning local modules in dynamic networks. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*, pages 176–188. PMLR, 2021.

[130] P.M.J. Van den Hof and K.R. Ramaswamy. Single module identification in dynamic networks - the current status. In *Preprints 21st IFAC World Congress*, pages 52 – 55, Berlin, Germany, 2020. Extended abstract.

[131] J.W van der Woude. A graph-theoreric characterization for the rank of the transfer matrix of a structured system. *Mathematics of Control, Signals, and Systems*, 4(1):33–40, 1991.

[132] D. A. Van Dyk and X. L. Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

[133] H. J. van Waarde, P. Tesi, and M. K. Camlibel. Topological conditions for identifiabaility of dynamical networks with partial node measurements. *IFAC-PapersOnLine*, 51-23:319–324, 2018. Proc. 7th IFAC Workshop on Distrib. Estim. and Control in Networked Systems.

[134] M. S. Veedu and M. V. Salapaka. Topology identification under spatially correlated noise, 2020. ArXiv:2012.04175.

[135] A. Venkitaraman, H. Hjalmarsson, and B. Wahlberg. Learning sparse linear dynamic networks in a hyper-parameter free setting. *IFAC-PapersOnLine*, 53(2):75–79, 2020. 21st IFAC World Congress.

[136] G. Wahba. A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem. *The Annals of Statistics*, 13(4):1378 – 1402, 1985.

[137] G. Wahba. *Spline Models for Observational Data*, volume 59. SIAM, 1990.

[138] B. Wahlberg, H. Hjalmarsson, and J. Mårtensson. Variance results for identification of cascade systems. *Automatica*, 45(6):1443–1448, 2009.

[139] L. Wasserman. Bayesian model selection and model averaging. *Journal of Mathematical Psychology*, 44(1):92 – 107, 2000.

[140] H. H. M. Weerts, M. Galrinho, G. Bottegal, H. Hjalmarsson, and P. M. J. Van den Hof. A sequential least squares algorithm for ARMAX dynamic network identification. *IFAC-PapersOnLine*, 51-15:844–849, 2018. Proc. 18th IFAC Symp. System Identification.

[141] H. H. M. Weerts, J. Linder, M. Enqvist, and P. M. J. Van den Hof. Abstractions of linear dynamic networks for input selection in local module identification. *Automatica*, 117(108975), 2020.

[142] H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers. Identification of dynamic networks operating in the presence of algebraic loops. In *Proc. 55nd IEEE Conf. on Decision and Control (CDC)*, pages 4606–4611. IEEE, 2016.

[143] H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers. Identifiability of linear dynamic networks. *Automatica*, 89:247–258, March 2018.

[144] H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers. Prediction error identification of linear dynamic networks with rank-reduced noise. *Automatica*, 98:256–268, December 2018.

[145] H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers. Single module identifiability in linear dynamic networks. In *Proc. 57th IEEE Conf. on Decision and Control (CDC)*, pages 4725–4730, Miami Beach, FL, 2018. IEEE.

[146] H.H.M. Weerts. *Identifiability and Identification Methods for Dynamic Networks*. PhD dissertation, Eindhoven University of Technology, 2018.

[147] G. C. G. Wei and M. A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

[148] Jan C. Willems. The behavioral approach to open and interconnected systems. *IEEE Control Systems Magazine*, 27(6):46–99, 2007.

[149] N. Woodbury, A. Dankers, and S. Warnick. Dynamic networks: representations, abstractions and well-posedness. In *Proc. 57th IEEE Conf. on Decision and Control*, pages 4719–4724, Miami Beach, FL, 2018.

[150] D. Youla. On the factorization of rational matrices. *IRE Transactions on Information Theory*, 7(3):172–189, 1961.

[151] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, 68:49–67, 02 2006.

[152] Y. Yuan, G-B. Stan, S. Warnick, and J. Gonçalves. Robust dynamical network structure reconstruction. *Automatica*, 47(6):1230–1235, 2011.

[153] P. Zhang. On the convergence rate of model selection criteria. *Communications in Statistics-Theory and Methods*, 22(10):2765–2775, 1993.

[154] M. Zorzi and A. Chiuso. Sparse plus low rank network identification: a nonparametric approach. *Automatica*, 76:355–366, 2017.

List of symbols

L	Number of internal variables or nodes in a dynamic network
K	Number of external variables in r in a dynamic network
$\mathcal{L}$	Index set of all node signals: $[1, L]$
$\mathcal{R}$	Index set of all excitation signals in r: $[1, K]$
$\Phi_{ab}(\omega)$	cross power spectral density of vector signals $a(t)$ and $b(t)$
$\Phi_{a}(\omega)$	auto power spectral density of vector signal $a(t)$
$\mathcal{N}_j$	Set of indices of node signals with direct causal connection to node w_j
$\mathcal{R}_j$	Set of indices of excitation signals in r with direct causal connection to node w_j
$\mathcal{N}_j^+$	Set of indices of node signals that are w-out neighbors of w_j
$\mathcal{N}_j^-$	Set of indices of node signals that are w-in neighbors of w_j
$\mathcal{F}$	discrete-time Fourier transform
N	Length of the data
$\mathbb{E}$	Expectation operator
$\bar{\mathbb{E}}$	It refers to $\lim_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} \mathbb{E}$
G^0	Network matrix with modules
H^0	Network noise model
w_j	Node signal w_j, output of the target module
w_i	Node signal w_i, input of the target module
$\mathcal{Y}$	Set of indexes of nodes that appear in the vector of predicted outputs
$\mathcal{D}$	Set of indexes of nodes that appear in the vector of predictor inputs for predicted outputs $w_{\mathcal{Y}}$
$\mathcal{D}_j$	Set of indexes of nodes that appear in the vector of predictor inputs for prediction of node w_j
w_o	Output node signal w_j if it is not in set $w_{\mathcal{Q}}$
$\mathcal{Q}$	Set of indexes of nodes that appear both in the predicted output, and in the predictor input
$\mathcal{O}$	Set of indexes of nodes that only appear as predicted output: $\mathcal{O} = \mathcal{Y} \backslash \mathcal{Q}$
$\mathcal{U}$	Set of indexes of nodes that only appear as predictor input: $\mathcal{U} = \mathcal{D} \backslash \mathcal{Q}$
$\mathcal{A}$	Set of indexes of nodes that only appear as predictor input, that do not have any confounding variable effect: $\mathcal{A} \subseteq \mathcal{U}$

$\mathcal{B}$	Set of indexes of nodes that only appear as predictor input: $\mathcal{B} = \mathcal{U} \backslash \mathcal{A}$
$\mathcal{Z}$	Set of indexes of nodes that are removed (immersed) from the network when predicting w_y
$\mathcal{Z}_j$	Set of indexes of nodes that are removed (immersed) from the network when predicting w_j
v_k	Disturbance signal on node w_k
e	(White noise) innovation of the noise process v
ξ	(White noise) innovation of the noise process in the immersed and transformed network
$:=$	is defined as
$\varnothing$	empty set
$\mathcal{S}$	System
$\mathcal{M}$	Model set
T^0	Open-loop transfer function matrix of the network
$\mathbb{R}(z)$	Field of rational transfer functions

List of Abbreviations

SOC	State of Charge
BMS	Battery Management System
SEM	Structural Equation Model
BOLD	Blood Oxygenation Level Dependent
fMRI	functional Magnetic Resonance Imaging
HVAC	Heat Ventilation Air Conditioning
CVEM	Complete Vehicle Energy Management
LTI	Linear Time-Invariant
DSF	Dynamic Structure Functions
WNSF	Weighted Null-space fitting
SMPE	Simultaneous Minimization of the Prediction Error
EIV	Errors-in -variables
SISO	Single-Input Single-Output
MISO	Multiple-Input Single-Output
SIMO	Single-Input Multiple-Output
MIMO	Multiple-Input Multiple-Output
CLS	Constrained Least Squares
CRLB	Cramér-Rao Lower Bound
MCMC	Markov Chain Monte Carlo
MSE	Mean-squared Error
PEM	Prediction Error Method
EB	Empirical Bayes
AIC	Akaike information criterion
BIC	Bayesian information criterion
CV	Cross Validation

ML	Maximum Likelihood
PDF	Probability Density Function
OLS	Ordinary Least Squares
RLS	Regularized Least Squares
SNR	Signal-to-Noise Ratio
RKHS	Reproducing Kernel Hilbert Space
MAP	maximum-a-posteriori
MMSE	minimum mean-square error
EM	Expectation-Maximization
MCEM	Monte-Carlo Expectation-Maximization
SLR	Sequential Linear Regression
SLS	Sequential Least Squares
BJ	Box-Jenkins
FIR	Finite Impulse Response
ARX	AutoRegressive with eXternal input
ARMAX	AutoRegressive Moving Average with eXternal input
OE	Output error
IV	Instrumental variables
w.p.	with probability
w.p. 1	with probability one
w.r.t.	with respect to
ROC	Receiver Operating Characteristic
TP	True Positive
FP	False Positive
TPR	True Positive Rate
FPR	False Positive Rate
RMSE	Root Mean Squared Error
LDM	Local Direct Method

List of Publications

Peer-reviewed Journals

K.R. Ramaswamy and P.M.J. Van den Hof. A local direct method for module identification in dynamic networks with correlated noise. *IEEE Trans. Automatic Control*, Vol. 66, no. 11, pp. 5237-5252, November 2021.

K.R. Ramaswamy, G. Bottegal and P.M.J. Van den Hof. Learning linear models in a dynamic network using regularized kernel-based methods. *Automatica*, Vol. 129, Article 109591, July 2021.

S.J.M. Fonken, K.R. Ramaswamy and P.M.J. Van den Hof. A scalable multi-step least squares method for network identification with unknown disturbance topology. *To appear in Automatica, July 2022*. ArXiv: 2106.07548.

K.R. Ramaswamy, P.Z. Csurcsia, J. Schoukens and P.M.J. Van den Hof. A frequency domain approach for local module identification in dynamic networks. *To appear in Automatica, November 2022*. ArXiv: 2105.10901.

K.R. Ramaswamy, O. Leeuwenburgh, R.M. Fonseca, M.M. Siraj and P.M.J. Van den Hof. Improved sampling strategies for ensemble-based optimization. *Computational Geosciences*, May 2020.

K.R. Ramaswamy, G. Bottegal and P.M.J. Van den Hof. Learning linear modules in a dynamic network with missing node observations. *In preparation for submission to Automatica.*

Peer-reviewed Conferences

V.C. Rajagopal, K.R. Ramaswamy and P.M.J. Van den Hof (2021). Learning local modules in dynamic networks without prior topology information. *Proc. 60th IEEE Conf. Decision and Control (CDC)*, Austin, TX, USA, December 13-15, 2021, pp. 840-845.

P.M.J. Van den Hof and K.R. Ramaswamy (2021). Learning local modules in dynamic networks. *In Proceedings of the 3rd Conference on Learning for Dynamics and Control, volume 144 of Proceedings of Machine Learning Research*, pages 176–188. PMLR, ETH Zurich, Switzerland.

V.C. Rajagopal, K.R. Ramaswamy and P.M.J. Van den Hof (2020). A regularized kernel-based method for learning a module in a dynamic network with correlated noise. *Proc. 59th IEEE Conf. Decision and Control (CDC)*, Jeju Island, Republic of Korea, 15-18 December 2020, pp. 4348-4353.

P.M.J. Van den Hof and K.R. Ramaswamy (2020). Path-based data-informativity conditions for single module identification in dynamic networks. *Proc. 59th IEEE Conf. Decision and Control (CDC)*, Jeju Island, Republic of Korea, 15-18 December 2020, pp. 4354-4359.

P.M.J. Van den Hof and K.R. Ramaswamy (2020). Single module identification in dynamic networks - the current status. Extended abstract, *Preprints 21st IFAC World Congress*, 12-17 July 2020, Berlin, Germany, pp. 52-55. Invited survey paper

K.R. Ramaswamy, P.M.J. Van den Hof and A.G. Dankers (2019). Generalized sensing and actuation schemes for local module identification in dynamic networks. *Proc. 58th IEEE Conf. Decision and Control (CDC)*, Nice, France, 11-13 December 2019, pp. 5519-5524.

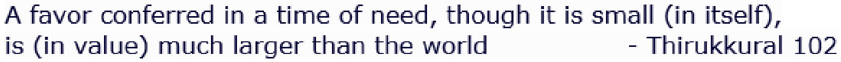
P.M.J. Van den Hof, K.R. Ramaswamy, A.G. Dankers and G. Bottegal (2019). Local module identification in dynamic networks with correlated noise: the full input case. *Proc. 58th IEEE Conf. Decision and Control (CDC)*, Nice, France, 11-13 December 2019, pp. 5494-5499.

K.R. Ramaswamy, G. Bottegal and P.M.J. Van den Hof (2018). Local module identification in dynamic networks using regularized kernel-based methods.

Proc. 57th IEEE Conf. Decision and Control (CDC), 17-19 December 2018, Miami Beach, FL, pp. 4713-4718.

Acknowledgments

காலத்தி னாற்செய்த நன்றி சிறிதெனினும்
ஞாலத்தின் மாணப் பெரிது. - திருக்குறள் 102



A favor conferred in a time of need, though it is small (in itself),
is (in value) much larger than the world - Thirukkural 102

I would like to use this section to thank many people without whom I might not be where I am today.

First and foremost, I would like to extend my deepest gratitude to my mentor and promotor, prof. dr. ir. Paul Van den Hof. The greatest help one can offer to someone is to provide them an opportunity. During my Master's graduation project, I informed him that I want to pursue a Ph.D. under his supervision. He believed in my capabilities and offered me a Ph.D. position. I would be forever grateful for that. He has always been my mentor, patiently guided me throughout my academic life, and always supported my decisions. I have never felt uncomfortable at any point during my academic life and all the credit for that goes to him. I always enjoyed the whiteboard discussions with him. He granted me the full freedom to come up with ideas and discuss them with him. In addition to the research skills, I learned a lot about the way of working, communication, leadership and management by working closely with him for the past 4 years. It has been an absolute pleasure and privilege to work with him, and I thank him for his teaching, advice, guidance, and inspiration.

I am incredibly grateful to my other promotor prof. dr. Siep Wieland. He is the sole reason I selected the Control systems research group when I came to the Netherlands for my Masters's studies. After a gap of 4 years from academia, I came to the Netherlands for my master's. I was quite apprehended since I had to refresh my knowledge and had a lot of catching up to do. His beautiful way of teaching in the modeling dynamics course inspired me, built my confidence, and also made me choose the control systems group. He is one of the best teachers I have ever seen. His enthusiasm is quite contagious and I have always enjoyed the discussions with him after courses, seminars, and exams. He has always guided me in shaping my research by providing valuable inputs in my every year evaluations. I thank him wholeheartedly for playing a very important role in shaping my career.

I would also like to extend my deepest gratitude to my co-promotor, dr. Giulio Bottegal. He is responsible for kick-starting my Ph.D. career and I am very grateful for that. He has always supported me throughout my Ph.D. and whom I will turn to for guidance. His invaluable advice, constructive criticism, and willingness to brainstorm ideas have tremendously helped to shape my research and future career. I always loved the effective discussions I had with him, even though it is always short. I would also like to acknowledge the guidance and advice I got during my collaboration with prof. dr. ir. Johan Schoukens and dr. ir. Péter Zoltán Csurcsia. I thank them for their collaboration and help me explore new ideas. I would also like to offer my sincere thanks to my students Stefanie Fonken and Venkat Rajagopal. They have played an important role in developing new algorithms. My thanks to dr. Rahul Mark Fonseca and dr. Olwijn Leeuwenburgh for their guidance at TNO during my master's graduation project, for encouraging me to pursue Ph.D., and also for recommending me to Paul for it.

I would like to express my most profound appreciation to my committee members prof. dr. Håkan Hjalmarsson, prof. dr. Donatello Materassi, prof. dr. ir. Tom Oomen, and dr. Arne Dankers. I thank them for making time to be on my thesis committee as well as read my thesis and approve it. Their comments and suggestions are highly appreciated.

It has been a great pleasure to be a part of the CS group for the past six years including my time in my Master's studies. I would like to thank Roland for providing crucial feedback and suggestions during the annual evaluation. Discussions with Maarten, Mircea, and Jobert have always been illuminating. Tijs and Henk Jan have always been friendly, and I shall never forget the time spent with the CS group at Miami and Nice. I thank Will and Wim for all the technical support that has been offered.

The time I spent during my Ph.D. would not be enjoyable without the friendship of Amritam, whom I call "Boss" and he calls me back "Big Boss". The coffee breaks we have has always been the energizer for me to go back to the office and work in full force. We have had numerous conversations, and jovial chitchats and I always enjoy how we brainstorm new ideas, and also futuristic thoughts on machine learning and control systems. He has always been my companion at conferences. He is one person who is very passionate about academia and I wish him all the best for his future career. I would like to thank Mohsin for his advice and assistance during my master's and first year as a Ph.D. student. I would like to thank Tom for the friendship, scientific debates, coffee, brainstorming, Benelux conferences, the car rides he has offered to conferences, and not to forget for the spicy noodle challenge. I would also like to thank my other SYDYNET project team members Harm, Shengling, Lizan, Xiaodong, and Mannes for their friendship, fun working environment, and the Friday discussions. Thanks to Paul Padilla, Zuan, David, Beppe, Harm, Dhruv, Ruben, Henrik, Tuan, Feye, Chris, Yanin, Tom and Carlos for making the environment fun and casual. I thank all the past and present other CS members as well. Also, I would like to thank Diana and Hiltje. Without them, the CS group is incomplete. I thank them for their care, kindness, and tolerating all my requests and questions.

I thank my friend Roshni for putting up with me as a project team partner during our master's studies. Thanks to her, Koen, and Sambu for the game nights, good food, and their company which made the weekends and holidays fun. I thank Roshni's parents for the hospitality they offered. I would also like to thank Rokesh, Pallavi, Nikilesh, and Amulya for the friendship and the kind hospitality they offered. I would like to thank Laxman and Shylu for the care, support, and hospitality they offered.

Finally, my sincere gratitude goes to my family for their unparalleled love, support, and encouragement. It would be very incomplete without thanking these two people: my sister-in-law Aarthi and brother-in-law Vijay. I would like to call them my own sister and brother rather than my in-laws. Without them, I do not know whether I would have survived when I came to the Netherlands or completed my master's and Ph.D. I am greatly indebted to the care, love, kindness, support, guidance, knowledge, and wisdom that they provided to me and that debt is very hard to pay back. I would like to thank my niece Tashvi. I watched her grow from a newborn baby along with my master's and Ph.D. I always enjoyed the talks and fun we had, and she has always been the person who makes me forget all my stress and worries. I am very grateful to my parents-in-law Narayanaswamy and Sasikala. It is hard for any parents-in-law to support the decision that I took 6 years ago. They faced a lot of hardships due to that but they supported me in all ways and said to me only one thing "Go on. We know you will succeed". I am forever grateful for the trust and love they had in me. My sincere love goes to my late grandfather Thiruvengadam who has always been my friend and mentor and taught me an important lesson about how to lead my life by being an example himself. He constantly asks about my Ph.D. graduation and always wanted to see it. He is one person whom I will miss and I wish he is watching my graduation from heaven. My sincere love to my grandmother Vedavalli for her love and taking care of me most of my life from my very childhood. I would also acknowledge my brother Karthik, sister-in-law Samya, nephew Hari and Harshith, and all my family members and friends from India. My mom and dad, Vasantha and Ramaswamy, I thank you for the sacrifices, unconditional love, prayers, support, encouragements, care and instilling values to make me who I am today.

Last but definitely not least, I would like to dedicate my special thanks with deep love to my beloved wife, Adithi. Without her, this thesis or my successful Ph.D. journey would not be possible. She has always been my best friend and my pillar of support, cherishing the high and sharing the low during this journey. She is my no.1 critic, guiding light, and always pushed me to achieve greater heights. She is the one who made me realize my capabilities, taught me to move out of my comfort zone, and pushed me towards achieving greater heights. I still cannot imagine how she managed to listen to all my blah-blahs related to my Ph.D. Most importantly, I would like to thank her for the trust and confidence she had in me. She stood along with me in the past when I was in a very difficult situation, encouraged me to follow my ambitions, sacrificed a lot, and she walked with me holding my hands throughout my journey to where I am today. Thanks is merely a word and it cannot express my feelings for what you have done.

Karthik Raghavan

Eindhoven, 02-04-2022

Curriculum Vitae

Karthik Raghavan Ramaswamy was born in Chennai, India on 27th November 1989. After finishing high school, he pursued his bachelor's degree at Anna University in India in Electrical and Electronics Engineering from 2007 till 2011. During his bachelor's study, he was the valedictorian of his class and was awarded the gold medal. From 2011 to 2015, he worked at L&T Kobelco, a joint venture between Larsen & Toubro and Kobe steel, as a Control and Automation design engineer for Heavy machinery. In 2014, he obtained his M.B.A from Annamalai University. In 2015, he was admitted to the Eindhoven University of Technology for the master's program in Systems and Control and completed it with Great Appreciation in 2017. During his master's studies, he did an internship at Cleantron, a battery manufacturer and the graduation thesis at TNO, the Netherlands Organisation for applied scientific research. After his master's studies, he has been working towards a Ph.D. degree at the Control Systems Group of the Electrical Engineering Department at the same university.

In his 4 years of work experience as a design engineer for a machinery manufacturer, he developed strong communication skills and an understanding of the industrial process and business. He has interacted with several customers across the globe and has been involved in translating their requests into high-quality designs and software. He had the chance to appreciate the value of working in multi-cultural teams with different technical backgrounds, coordinating with them to bring up the final product.

In his Ph.D. research, his primal objective was to develop scalable and computationally less complex identification methods/algorithms for finding accurate models of systems in a dynamic network using supervised machine learning, Bayesian methods, system identification, and signal processing. As a result, his doctoral thesis has successfully provided a practical guide to learning interconnected systems in a complex dynamic network.

Since March 2022, he has been working at ASML in the Netherlands.

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

The advancements in technology have made real-life systems increasingly complex. They encompass various sub-systems (modules) that are interconnected. These large-scale interconnected systems, known as dynamic networks, are becoming increasingly ubiquitous. The data-driven model learning of these dynamic networks has become of paramount importance in various domains and is needed for designing, analyzing, monitoring, and controlling these systems. How do we learn/identify the dynamic network under different scenarios? This dissertation guides and provides the users/experimenters with the needed tools to learn modules embedded in a dynamic network.