Technische Universiteit Eindhoven
University of Technology
Department of Mechanical Engineering
Dynamics and Control Research Group

Biped side stepping

DC2016.077

Richard Kooijman

Supervisors:
Dr. MSc D.Kostic
Prof. Dr. H.Nijmeijer
Commission:
Prof. Dr. H.Nijmeijer
Dr.Ir.A.G.de.Jager
Dr. A.Saccon

Version 1.0

Eindhoven, September 2016

Abstract

In order for robots to most effectively move from industry and into our homes, it is our belief that robots need to be physically similar to humans and thus scale down in: power, mass, and size and move around in a similar manner to humans. This requires them to also perform autonomous biped locomotion. These so-called humanoid robots are still in early stages of development with much of the essential functionality still under development. The focus of this thesis is stable and robust sidestepping locomotion of a humanoid robot. Tulip, a 12 joint humanoid developed to participate in the RoboCup Humanoid Football Championships, is used as the test bed for the study on which biped locomotion modeling and control techniques are developed. A 12-DoF model that is able to match the 3D motions that the Tulip robot hardware is capable of performing are derived and verified by comparing simulated and experimental data. Biped stability is examined and sidestepping locomotion states are introduced. Joint trajectories are evaluated considering the resulting joint torques. An algorithm is developed to produce joint trajectories as a function of the desired stable sidestepping speed. The algorithm utilizes position control and inverse kinematic computations for the joints. A relation is found for the maximum sidestepping velocity for a biped as a function of step size and vertical height of the centre of mass. Simulations of robot locomotion utilizing the developed algorithm are shown to be dynamically stable.

Words of thanks

This thesis would not have been possible without the support of many people over the years. Many thanks for my professor Henk Nijmeijer, from whom I feel honoured to receive guidance and Marijke Creusen for all the administration support.

There are a number of people I would like to give a more special thanks. To Dragan, for all of the great coaching, support and the friendship. My parents and brother for all of their support. Most of all, to my wife, Boa Wang Kooijman, who is so perfect, that she always inspires me to be better. Thank for the inspiration, thanks for putting up with all those study days and thanks for completing my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The humanoid robots (humanoids), are made such as to mimic human-like characteristics. Humanoids that use two legs for their locomotion are known as bipeds. Bipeds have inspired many science fiction authors but as of yet have shown little practical application. This raises the question, why study bipeds at all? The answer lies in the huge potential that they have, both directly and indirectly, to advance mankind. An example of the direct potential can be seen in the field of health care. An aging society is putting an ever increasing burden on the health care industry and one possible solution to this problem is to design robots which are able to assist the elderly. These robots would have to be working and interacting with humans in their normal environment and ideally be using the tools designed for humans. For these reasons, humanoid robots are a natural choice for the task. The indirect potential of humanoids refers to the technology spin offs which they will undoubtedly inspire. Although robotics is already a challenging field, stretching across numerous technological fields such as electronics, mechanics and software, humanoids are right at the cutting edge of this technology. Technological advances made here have a high potential for impacting the industrial robot industry, where the robots are known for being bulky, inflexible and dangerous by comparison. Technological advances in humanoids are also expected to impact health care indirectly, due to the fact that a better understanding of humanoids is synonymous to better understanding of the human body.

There is a lot of research done on forward walking however relatively much less on sideways stepping. This particular form of locomotion is important, for example, for moving in narrow spaces like a kitchen. Another example where robot movement in narrow spaces can be important is for preforming rescue tasks, especially where the environment is too dangerous for humans to go into. The promotion of the development of robotics to preform complex tasks in dangerous, degraded, human-engineered environments, was the purpose of the DARPA Robotics Challenge, [2], held for 2012 to 2015. The winning robot, SHAFT, of the 2013 challenge can be seen in figure 1.1, opening a door after preforming side stepping locomotion.

In order to develop side stepping locomotion the following problem statement is formulated:

> *For a given target position and time, develop an algorithm which calculated motions in the robot joints such that the biped preforms stable side stepping locomotion to intersect that target position before the target time.*

A humanoid robot, named TUlip,[18], was developed by the three technical universities in the Netherlands (TU Delft, TU Eindhoven, UTwente) and Philips, to take part in a similar challenge as the DARPA Robotics Challenge, namely the RoboCup, adult soccer league. This robot shown in figure 1.2, provides a test bed for the study. TUlip has 12 actuated degrees of freedom (DoF), provided by four 90 Watt, and eight 60 Watt, DC motors each with its own planetary gearbox.

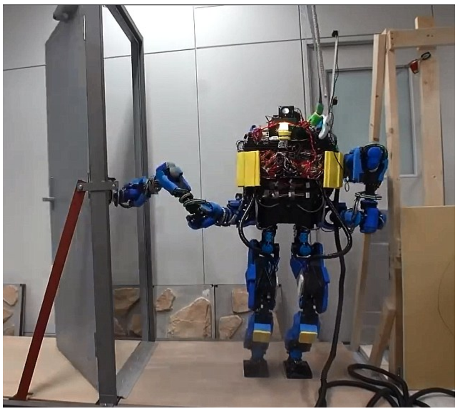


Figure 1.1: Photo taken humanoid SHAFT opening a door after preforming sidestepping locomotion, taken during the DARPA Robotics Challenge

To evaluate whether the joint trajectories indeed lead to stable locomotion, a multi-body dynamics model of TUlip is required. Modeling of the TUlip dynamics is discussed in Chapter 3. Calculation of sound and feasible joint trajectories that solve the given problem requires a solid understanding of the biped dynamics and the concepts of biped balance that are discussed in Chapter 4. In the same chapter, a side stepping gate, particular type of biped locomotion. The algorithm to compute the joint trajectories is also explained in Chapter 4. The conclusions to this project and recommendations for future research are given in Chapter 5.

Figure 1.2: Humanoid named TUlip developed by TU Delft, TU Eindhoven, UTwente and Philips.

Chapter 2

Literature study on biped locomotion

2.1 Introduction

The literature study was preformed to investigates various modelling and trajectory generation techniques that are currently being used. At the end of the chapter, a technique to generate the robot reference trajectories is chosen. The selected behavior and the trajectory generation technique are going to be worked out in the following chapters.

2.2 Modelling

2.2.1 Biped modelling

Van Zutven et al in [75] performed a comprehensive study on biped modelling approaches. Based on thorough experimental analysis, four major conclusions are drawn. Firstly, that point feet could successfully be imitated on humanoid robots by turning off the ankle control. Next, that impacts of the feet with the ground can be modeled with discontinuous velocities and energy loss. It is also found that there are significant couplings between bodies in a humanoid robot and finally, that dynamics in the coronal plane influences dynamics in the sagittal plane and vice versa. These conclusions suggest that a 2D model is not sufficient for modelling the biped. Consequently, in this project 3D modeling of TUlip is performed.

2.2.2 Ground contact model

In [11], Ehsaniseresht takes closer look at the modeling contacts between bipeds and the ground. He finds out that while many contact models used for the biped locomotion concentrate on a particular gate, such as walking, these are less suited for the other gaits such as running or jumping. A smoothing function is used to turn a discontinuous friction model of, for example, the coulomb friction around zero velocity, into a continuous one. This discontinuity, according to Ehsaniseresht, may result in modelling errors. This statement is not supported and goes against Zutven et als finding mentioned earlier. A reason for using the continuous contact model, however, is that a discontinues model results in a so called stiff equation of motion that are difficult to solve without relatively sufficiently small integration time-step.

2.3 Trajectory generation

There are different strategies to generate trajectories for biped locomotion, and these are difficult to classify due to the overlapping principles involved. One method, proposed by Vanderborght [63], calssifies these as natural dynamics-based control, soft computing, and model-based trajectory generation. To begin with, stability and side stepping locomotion are researched.

2.3.1 Stability

Zero moment point (ZMP) is the most popular criterion for evaluating the stability of a biped robot [66]. This criterion evaluates the reaction forces acting inside an area defined by contact points between a biped and walking surface. During running or jumping, the humanoid loses complete contact with the floor and thereby dynamic stability can no longer be guaranteed using this criterion. If these locomotion phases are periodically repeated, their stability can be evaluated by computing the spectral radius of the Jacobian of the Poincar map associated with the cyclic motion, as described in [32]. This criterion describes the robustness of the periodic solution against small perturbations. Stability is therefore not guaranteed for larger perturbations or even during the initial transition phase the biped needs to go through to get up to the desired locomotion speed. To ensure that the biped motion remains stable, in this project it is therefore chosen that sole of one or both feet of the biped always remain in contact with the walking surface. Jumping behaviors are consequently not considered.

2.3.2 Side stepping locomotion

The side stepping gait has not been researched as much as the forward walking gait [15]. Alitavoli [4] is one of a few researchers concentrating on the lateral biped motions. To achieve a control lateral motions, he proposes a novel sliding-mode tracking and control algorithm. However, he evaluates that algorithm only in simulations using a two link, 2D biped model. Although not much research is focused on lateral motions, the stability considerations are similar to the forward walking. Hence, both ZMP and the Poincar stability map [56] can be used to evaluate stability of the side stepping.

A possible difference between the sidestepping and forward walking gaits lies in the kinematic redundancy available in the direction of motion. In the field of robotics, the so called sagittal and coronal plains are used to describe the motion plains. The sagittal pain is a longitudinal plane that divides the body into right and left sections. The coronal plain is a plane that divides the body into front and back parts, perpendicular to the sagittal and horizontal planes.

Rotation redundancy of the torso is often used to provide corrective action to maintain balance [17, 35] during the walking. As for TUlip, this rotational redundancy however only exists in the direction normal to the direction of motion (in the so called coronal plain), which can only be used for the corrective balance in the orthogonal direction to the sidestepping.

2.3.3 Natural dynamics-based control

This first group is characterized by the fact that design of the trajectories is intrinsic to the mechanical design of the robot. Passive dynamic walkers give good example of such walking gaits. The passive dynamic walkers, pioneered by McGeer [30], exploit the natural ballistic dynamics of the biped walking system to preform stable rhythmic cyclic gaits without actuation or external energy sources other than gravity. Stability is quantified by examining the eigenvalues of the linearized step-to-step return map, taken around a point in the period either immediately preceding or immediately following the time of contact between foot and the ground. Tedrake [60] extended this approach to actuated dynamic walkers, showing that simple controllers could be used allowing the robot to walk stably on even terrain and even up a small slope.

These approaches are suited for bipeds designed to preform a particular gaits only. For more bipeds such as the TUlip, that are aimed to perform much wider range of tasks, Virtual Model

Control (VMC)[43] , could be used instead of ballistic gaits. VMC can also be considered natural dynamics-based control where the natural dynamics are artificially created though the use of control loops. For example where a natural dynamics model would have physical spring, the VMC would have proportional controller.

Virtual components are placed at strategic locations within the biped or between biped and it's environment. A virtual spring and damper element could, for example, be placed between the robot hip and ground. The virtual forces produced by these elements are then mapped into physical torques at each of the robot joints. VMC has been successfully implemented in the Spring

Flamingo planar biped [41]. Virtual Gravity Compensation, VGC, is introduced by Keehong et al. [47]. The VGC combines VMC and capture points [42], and it is successfully implemented on a biped showing balance even when moving on uneven terrain. A major disadvantage of the VMC related methods is that parameters need to be tuned by hand and rely heavily on the experts experience and knowledge [63].

2.3.4 Soft computing methods

Soft computing such as neural networks and fuzzy systems are formal part of the computer science. These methods are characterized by the use of inexact solutions to computationally- complex tasks [74]. As such, the soft computing methods are particularly interesting for synthesis annd control of biped locomotion since the bipeds feature complex and non-linear dynamics of high order with inherently large model and parameter uncertainties due to, for example, changing friction and impacts between the feet and the ground.

Choi et al. [6] studied and verified with experiments that a fuzzy logic posture control for biped walking using ZMP feedback from force sensors on the feet could be used for robust disturbance rejection. Park et al showed in [37] that using the leg reference trajectories as input to the fuzzy-logic generator is a way to to generate stable locomotion for a seven link robot model in simulation.

Wiklendt et al showed using simulations in [68, 69] that a 3D biped gains ability to perform dynamic walking using evolution-strategies after two thousand learning cycles, whereby each learning cycle ends when the biped falls. This kind of neural network strategy results in walking gaits that closely resemble human ones [31]. This learning algorithm however relies on recreating the initial condi- tions for each trial cycle which is very difficult in real life. Another major issue with this method is that many humanoids would break down and need repairing each time they fall. Finally, the time span required to run such experiment in real life would also be infeasible due to wear of the mechanical parts and the general costs involved.

Reinforcement Learning (RL) on the other hand, is shown to be practically more feasi- ble. Similar to evolution strategies, this generic framework learns new behaviors though rewards for good and bad behaviors. Tedrake et al [59] used RL algorithm on a simple biped which started walking within a minute and showed that the learning converges after 20 minutes. Schuitema et al [46] showed that using RL on a 2D walking robot named LEO, it is possible to make over 43000 footsteps during an 8 hour experiment. It should be noted that LEO was designed and built to be able to walk, fall and stand up without human interaction and in [46] it is men-tioned that the most bipeds are not designed robustly enough to withstand the large number of learning trials required.

Experimental evidence has shown that pattern generators found in spinal cords of mammals are at least partly responsible for their locomotion [61]. This discovery inspired the development of Central Pattern Generators (CPGs). Matsubara et al in [28] successfully applied a CPG-based biped locomotion controller using a neural oscillator model proposed by Matsuoka [29], to preform stable walking gait using a 6 link 2D biped.

2.3.5 Model-based approach

The vast majority of biped gaits are designed to be so called Zero Moment Point (ZMP) stable [63]. This involves generating joint trajectories that make sure that the biped remains dy- namically stable at all times. Due to the high computational cost for calculating multi-body dynamics required for ZMP computation, the most popular methods do this off-line [39]. This subgroup, which Sugihara et al [55] call trajectory replaying divides the problem into two sub-problems, namely, locomotion planning and control. The other subgroup, termed "real-time generation" does the planning and control in a more generic way.

An advantage of the trajectory replaying is absence of concerns about computational demands. A good example of a computationally expensive technique which can only be performed offline is the full posture goal method [23, 22]. Given the initial and final goal postures and using kinematic and dynamic models of the biped as well as geometric model of the environment, this method performs a heuristic search though the entire configuration space to find all statically stable and collision avoiding configurations. Next, a smoothing function determines the kinematic trajectory and finally a dynamic filtering method [72], constraining the ZMP within the support polygon, gives the joint velocities. Kuffner et al. used in [23] the full posture goal method to simulate the H6 humanoid model (33-DOF) placing the right foot above the surface of an obstacle while balancing on the left leg. This was also verified in an actual robot experiment.

Offline trajectory planning is not performed exclusively for computationally expensive methods. Kajita et al [21] proposed modelling the biped as a Linear Inverted Pendulum Model (LIPM) witah a linearized relationship between Center Of Mass (CM) and ZMP. Takanishi et al. proposed in [58] a method to solve the ZMP equations by transforming the equations of motion into the frequency domain. Kagami et al. later on expanded this notion in [19], proposing a method to solve the problem in the discrete time-domain.

Optimization criterion using space-time constraints is another offline technique. Wang et al used in [67] a wide range of object functions including terms for power minimization and angular momentum minimization, for generating a trajectory for a 30 DOF model. The resulting trajectory interestingly posses numerous human-like features including active toe-off, near-passive knee swing, and leg extension during the swing phase. An optimization criterion can also be used online, as Dekker demonstrated in simulations for a 11 link biped in [1]. A possible limitation, however, is only local optimality of the resulting solution trajectory (for each time step) and not for the com- plete locomotion period. Furthermore, retuning gains of the cost function might be necessary for different locomotion phases in order to come up with feasible robot gaits [45].

Online techniques offer great advantages with respect to mobility and disturbance rejection. Based on the LIPM model, Kajita et al formulated in [20] the ZMP control as a servo problem and proposed use of preview control. They also showed that the preview controller also compensates for the ZMP error caused by the difference between the simplified single pendulum model and the more precise multi-body model.

Ha et al elaborate in [16] on the LIPM by coming up with the virtual height inverted pendulum model. This model takes into account all link masses and depending on a given trajectory makes a better approximation/simplification of multi-body dynamics of the biped. Ha demonstrates that online stabilization action can be achieved by varying the height of the bipeds CM.

While the LIPM model based walking control strategies aim to constantly maintain balance, the Foot Placement Estimator (FPE) method determines where the foot should be placed in order to restore balance. FPE is introduced by Dwight et al in [8], showing how a biped could restore balance by controlling swing foot position during the gait cycle. The theory does make the assumption that the mass in the legs of a biped can be neglected, which was shown to be untrue by Zutven et al in [75]. An extension to the FPE approach is introduced by Zutven et al in [62], called the Foot Placement Indicator FPI method. This method does not assume massless legs and has shown in simulation on a 5 link planar biped to work better than the FPE method.

Boer et al presented in [7] a foot placement framework in which a biped was capable of transitioning to a stable stationary posture given any plausible and common states of bipedal locomotion.

2.4 Conclusion

This chapter gives an overview of the literature on several topics that are relevant for the gait generation problem for lateral locomotion task for humanoid robots. The need for 3D modeling is highlighted by Van Zutven et al in [75] in contrast to more standard 2D modeling approach found in the literature.

After critical evaluation of the state-of-the-art found in the literature, it is decided to generate locomotion gait for the side stepping task using the ZMP method. To facilitate online gait generation, a computationally plausible LIMP method is found as the most appealing one for this project, since it is already used on many physical bipeds.

Chapter 3

Dynamic modeling of TUlip

3.1 Introduction

A model is a simplification of a real life system. Consequently, the modeling objectives have to be clear in order to capture physical phenomena one wishes to simulate. Concerning TUlips locomotion, there are 2 main reasons for modeling, namely for the purpose of trajectory generation and for verification that the generated trajectories are feasible for the robot and meet the side stepping objectives.

Trajectory generation deals with computing the joint trajectories that lead to stable gaits. For multibody robotic systems, this is a complex nonlinear problem. An elegant solution to this problem is to use of a simple ZMP model, such as the linear inverted pendulum model (LIMP) [21]. To evaluate the correctness of the resulting gait, a more realistic / complex dynamical robot model is required. Such a model can be used to verify that the gait is in fact stable in simulations before being implemented on the actual robot. The model can also be important for tuning the controller parameters.

In sections 3.1 and 3.2, we give a general description of the robot system, discussing the input/output variables along with modeling assumptions. In section 3.2, 3.3 and 3.4, a model decomposition into subsystems is proposed, namely the actuator, rigid-body, and contact models. The implementation of these subsystem models in Matlab/Simulink/SimMechanics is discussed in section 3.5. An experimental verification of the model is given in section 3.6.

3.1.1 Architecture of a model of TUlip multi-body dynamics

Inputs and outputs of the robot multi-body dynamics are given in figure 3.1.

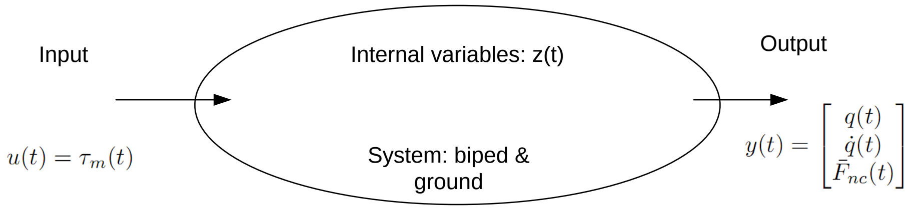


Figure 3.1: Inputs and outputs of the robot multi-body dynamics

When modeling a dynamical system, we need to define its inputs and outputs and states. The inputs are the control torques acting on the robot joints. These torques are generated by the robot motion controllers. In models of robot multi-body dynamics, dynamics of electrical motors are often neglected as bandwidths of these motors in series with the corresponding power amplifiers are generally much higher (2 times at least) than of the dynamics of the robot mechanics. In the particular case of TUlip, significantly higher bandwidths of the motor/amplifier combinations are not yet achieved in comparison with bandwidths of the robot mechanics. It is therefore decided to include the motor dynamics in the TUlip dynamical model.

The inputs to the model are the reference torques,$\tau_{m_{ref}}$ (i is $\{1, 2, , 12\}$). These torques are computed by the robot motion controllers While walking, TUlip experiences reaction forces with the ground. These forces can be considered as internal forces of the robot dynamics since their values are directly influenced by the actual robot kinematic configuration and dynamic behaviour.

TUlip is equipped with angular encoders and pressure sensors. Readings from these sensors can be considered as the system outputs and that is why the dynamical model of TUlips has the same outputs. Each robot joint has two encoder sensors, one at the motor side and another one at the load side (after the gearbox), measuring the motor and load angles,$q_{m,i}$, $q_{l,i}$ (i is $\{1, 2, , 12\}$), respectively.

In the control software of TUlip, the joint velocities are determined by differentiating and filtering the load angle,$q_{l,i}$ only. This is because the motor shaft rotates in the order of 2 faster than that of the load and a velocity estimation using $q_{m,i}$ is thus expected to be too inaccurate and noisy. Consequently $q_{m,i}$ is not used for any feedback purposes and is therefore also excluded from the model output. In order to reduce computation costs, the model outputs $q_{l,i}$ directly.

TUlips contact pressure sensors are placed at the corners of the sole of each feet. The pressure measurements are divided by the sensor area to determine the normal contact forces, f_{cni}, at each contact point $\bar{f}_{cn} = \begin{bmatrix} f_{cn1} & f_{cn2} & .. & .. & f_{cn7} & f_{cn8} \end{bmatrix}^T$. These contact forces can then be used to compute the actual ZMP location.

3.1.2 Model decomposition into subsystems

To simplify the modeling problem, the system is divided into 3 subsystems: the motor, robot mechanics, and contact dynamics with the ground. These describe the electro-mechanical motor/gearbox dynamics, robot multi-body dynamics, and the foot-floor contact dynamics, respectively. The two port-Hamiltonian representation of these sub- systems can be seen in figure 3.2, showing the relationship between potential and flow sources, $p(t)$ and $f(t)$ respectively. In the figure $p(t)$ and $f(t)$ with subscript l denote the power transfore at biped motor - joint boundary. $p(t)$ and $f(t)$ with subscript g represent the power transfer at the foot/floor boundary. When these sub-systems interact with each other, there is usually a bilateral coupling whereby one of the variables crossing the system boundary is determined by the system and the other by the environment. As the product of $p(t)$ and $f(t)$ is a measure of the power transfer, the direction of the potential and flow sources need to be consistent throughout the system to prevent any causality errors.

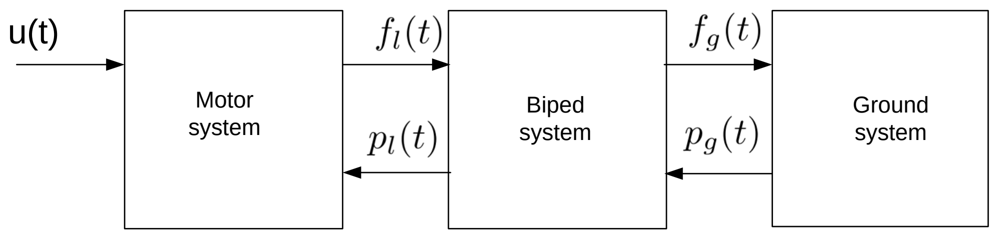


Figure 3.2: 2-port representation of the model subsystems. Here, $p(t)$ and $f(t)$ are the generalized forces and velocities, respectively, crossing the subsystem boundaries

The implication of the coupling relationship between the subsystems is demonstrated by considering two common methods for computing the ground forces, $f_g(t)$:

- The penalizing method. [25]

- The constraint method. Non penetrating constraint equations are added to the system of equations at the time of contact, t^*, and using The Lagrange multiplier theorem, the constraint force is computed, as discipled by Nathan van de Wouw in [71].

Since a particular type of the contact model has influence on the modelling accuracy and simulation time, it needs to be chosen carefully.

3.2 Model of the biped multi-body dynamics

TUlip is made up of 17 individual bodies (links) connected by 16 actuated joints 6 in each leg, 1 in each arm and 2 connecting the neck to the head and torso. As the bodies can be considered as stiff, so we assume that these can be modeled as rigid bodies. The dynamical contributions of the head, neck and arms to the overall system dynamics is negligible. This reduces the modeled system to a 13 link articulated chain of rigid bodies. The articulated rigid body chain has 18 DoF, of which 12 are actuated and the additional 6 describe the connection of the chain to the inertial frame. The first step when describing the robot dynamics is to assign a coordinate frame to each link in the chain. The Denavit-Hartenberg convention described in [49], is chosen for this purpose.

3.2.1 Assignment of the link coordinate frames

Jaques Denavit and Richard S. Hartenberg, [49], proposed a matrix method of systematically assigning a coordinate frame $x_iy_iz_i$ to each robot link i. The convention relies on 4 parameters, known as the DH-parameters, that define the geometric position of one link w.r.t the previous one:

- link length a_i is the offset distance between the z_{i-1} and z_i axes along the x_i axis;
- link twist α_i is the angle from the z_{i-1} axis to the z_i axis about the x_i axis;
- link offset d_i is the distance from the origin of frame $i-1$ to the x_i axis along the z_{i-1} axis;
- joint angle θ_i is the angle between the x_{i-1} and x_i axes about the z_{i-1} axis.

The convention is illustrated in figure 3.3.

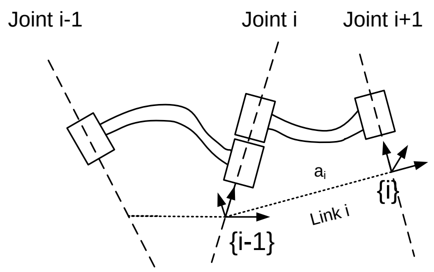


Figure 3.3: joint numbering in the standard DH convention.

For every link/joint pair the homogeneous coordinate transformation from the previous coordinate frame to the next one is described as

$$A_n^{n-1} = \mathrm{Trans}_{z_{n-1}}(d_n) \cdot \mathrm{Rot}_{z_{n-1}}(\theta_n) \cdot \mathrm{Trans}_{x_n}(a_n) \cdot \mathrm{Rot}_{x_n}(\alpha_n) \tag{3.1}$$

where

$$\text{Trans}_{z_{n-1}}(d_n) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_n \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{3.2}$$

$$\text{Rot}_{z_{n-1}}(\theta_n) = \left[\begin{array}{ccc|c} \cos\theta_n & -\sin\theta_n & 0 & 0 \\ \sin\theta_n & \cos\theta_n & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{3.3}$$

$$\text{Trans}_{x_n}(a_n) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & a_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{3.4}$$

$$\text{Rot}_{x_n}(\alpha_n) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_n & -\sin\alpha_n & 0 \\ 0 & \sin\alpha_n & \cos\alpha_n & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{3.5}$$

Thus:

$$A_n^{n-1} = \left[\begin{array}{ccc|c} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & r_n\cos\theta_n \\ \sin\theta_n & \cos\theta_n\cos\alpha_n & -\cos\theta_n\sin\alpha_n & r_n\sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|c} & R_n^{n-1} & & o_n^{n-1} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{3.6}$$

where R and o_n^{n-1} are 3×3 matrix and 3×1 vector, respectively, describing the position (orientation and translational displacement) of frame i wrt frame i-1.

Following the standard DH convention, we assign the coordinate frames to the links of TUlip as shown in figure 3.4. The corresponding DH parameters are given in Table 3.1. Coordinates of the contact points are shown in Table 3.2 and Inertial parameters of each robot link are given in Table 3.3. Note that these are described in the link coordinate frame as described by the DH convention.

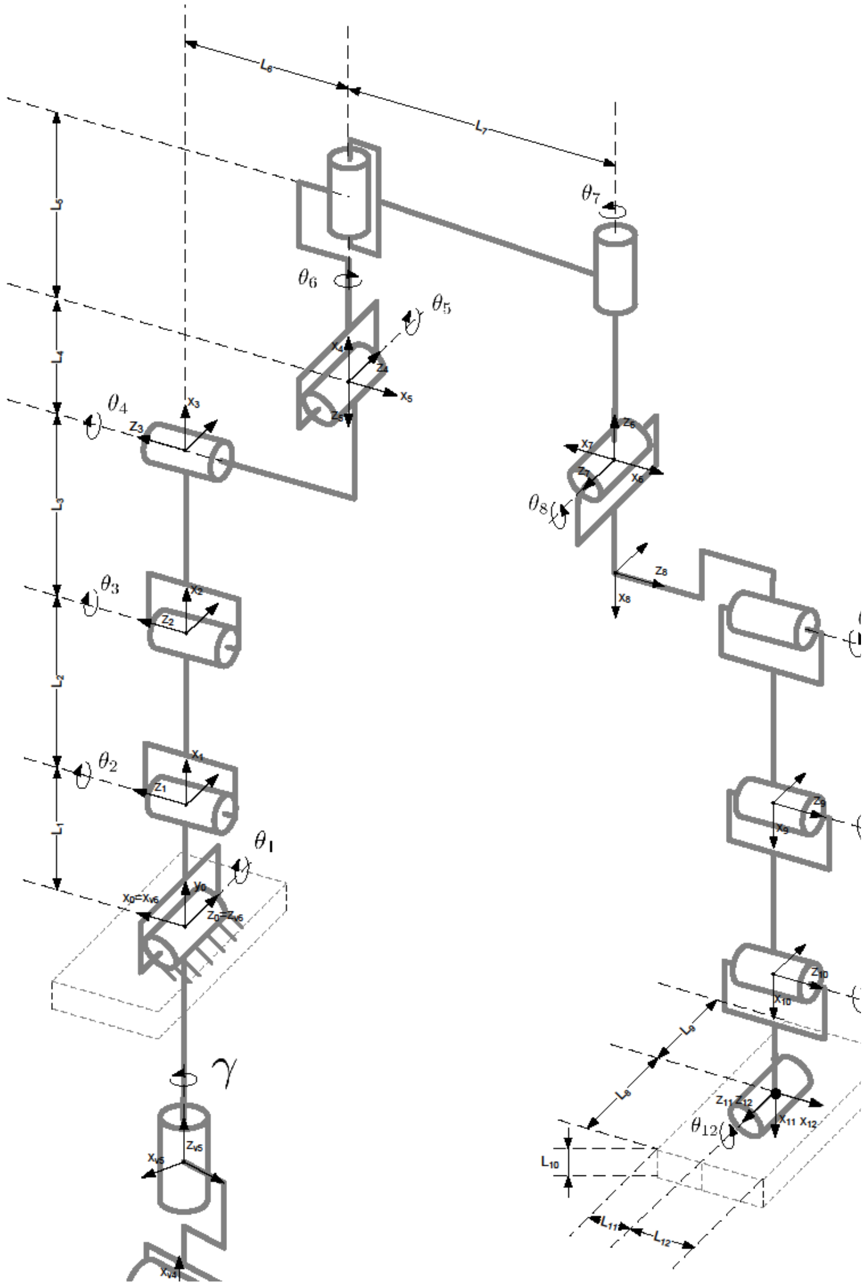


Figure 3.4: Schematic representation of the kinematic model of TUlip, from [75], with the link coordinate frames assigned according to the DH convention.

3.2.2 (Parameters)

The DH parameters are found in table [?]. Contact Points, CP, in link coordinate frame can be found in table [?] The link inertial properties, mass, inertia, and center of mass of link,L_i, in link coordinate frame, {i}, can be found in table [?].

3.2.3 Equations of motion

The kinematics of TUlip is described in terms of n=18 generalized coordinates $q = \begin{bmatrix} q_1 & q_2 \ldots & q_{17} & q_{18} \end{bmatrix}^T$.

The robot equations of motion, as described in [49], can be expressed as follows :

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) \tag{3.7}$$

where

- q is the vector of generalized joint coordinates.
- $\dot{q}$ is the vector of joint velocities.
- $\ddot{q}$ is the vector of joint accelerations.
- M is the symmetric inertia matrix.
- $C\dot{q}$ describes Coriolis and centripetal effects. Centripetal torques are proportional to $\dot{q}_i^2$, while the Coriolis torques are proportional to $q_i q_j$.
- F describes viscous and Coulomb friction.
- G is the gravity vector.
- τ is the vector of generalized actuation torques at the robot joints.

3.3 Contact Model

Unilateral contact describes a mechanical constraint which prevents two bodies from penetrating. Contact modeling describes how the contact force, λ, relates to a gap, h, (a measure of the distance between colliding bodies), as illustrated in figure 3.5.

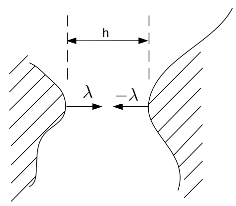


Figure 3.5: Unilateral contact, contact force λ, and gap h, a measure of the signed distance between the bodies.

L_i	Link Description	a_i [m]	$d_i[m]$	α [rad]	θ_i [rad]	q_{offset}
1	virtual link	0	q_1	$\pi/2$	0	0
2	virtual link	0	q_2	$\pi/2$	$\pi/2$	0
3	virtual link	0	q_3	$\pi/2$	$\pi/2$	0
4	virtual link	0	0	$\pi/2$	q_4	$\pi/2$
5	virtual link	0	0	$\pi/2$	q_5	$\pi/2$
6	Right foot	0	0	$\pi/2$	q_6	$-\pi/2$
7	right ankle	L_1	0	$\pi/2$	q_7	$\pi/2$
8	right lower leg	L_2	0	0	q_8	0
9	right upper leg	L_3	0	0	q_9	0
10	right lower hip	L_4	$-L_6$	$-\pi/2$	q_{10}	0
11	right upper hip	0	0	$-\pi/2$	q_{11}	$\pi/2$
12	torso	L_7	0	π	q_{12}	0
13	left upper hip	0	0	$-\pi/2$	q_{13}	π
14	left lower hip	L_4	0	$-\pi/2$	q_{14}	$\pi/2$
15	left upper leg	L_3	L_6	0	q_{15}	0
16	left lower leg	L_2	0	0	q_{16}	0
17	left ankle	L_1	0	$\pi/2$	q_{17}	0
18	left foot	0	0	0	q_{18}	0

Table 3.1: DH parameters

CP_i	Description	Right Foot, $\underline{r}^6_{RCP_i}$	Left Foot, $\underline{r}^{18}_{LCP_i}$
1	Outside toe	$[\ L_{12};\ -L_{10};\ -L_8\ ]$	$[\ L_{10};\ L_{12};\ L_8\ ]$
2	Inside toe	$[\ -L_{11};\ -L_{10};\ -L_8\ ]$	$[\ L_{10};\ -L_{11};\ L_8\ ]$
3	Inside heel	$[\ -L_{11};\ -L_{10};\ L_9\ ]$	$[\ L_{10};\ -L_{11};\ -L_9\ ]$
4	Outside heel	$[\ L_{12};\ -L_{10};\ L_9\ ]$	$[\ L_{10};\ L_{12};\ -L_9\ ]$

Table 3.2: Contact Points

L_i	Description	M_i	I^i_i	$\underline{r}^i_{CM}$
6	right foot	0.366	$(R^{w6})^T I^w_6 R^{w6}$	$[\ -L_{10}/2;\ 0;\ L_8/2\ ]$
7	right ankle	0.614	$(R^{w7})^T I^w_7 R^{w7}$	$[\ -L_1/2;\ 0;\ 0\ ]$
8	right lower leg	0.315	$(R^{w8})^T I^w_8 R^{w8}$	$[\ -L_2/2;\ 0;\ 0\ ]$
9	right upper leg	2.141	$(R^{w9})^T I^w_9 R^{w9}$	$[\ -L_3/2;\ 0;\ 0\ ]$
10	right lower hip	0.614	$(R^{w10})^T I^w_{10} R^{w10}$	$[\ -L_4;\ L_6;\ 0\ ]$
11	right upper hip	0.614	$(R^{w11})^T I^w_{11} R^{w11}$	$[\ 0;\ 0;\ -L_5/2\ ]$
12	torso	8.594	$(R^{w12})^T I^w_{12} R^{w12}$	$[\ -L_7/2;\ 0;\ L_5+0.17\ ]$
13	left upper hip	M_{11}	$(R^{w13})^T I^w_{11} R^{w13}$	$[\ 0;\ -L_5/2;\ 0\ ]$
14	left lower hip	M_{10}	$(R^{w14})^T I^w_{10} R^{w14}$	$[\ 0;\ 0;\ 0\ ]$
15	left upper leg	M_9	$(R^{w15})^T I^w_9 R^{w15}$	$[\ -L_3/2;\ 0;\ 0\ ]$
16	left lower leg	M_8	$(R^{w16})^T I^w_8 R^{w16}$	$[\ -L_2/2;\ 0;\ 0\ ]$
17	left ankle	M_7	$(R^{w17})^T I^w_7 R^{w17}$	$[\ -L_1/2;\ 0;\ 0\ ]$
18	left foot	M_6	$(R^{w18})^T I^w_6 R^{w18}$	$[\ L_{10}/2;\ 0;\ L_8/2\ ]$

Table 3.3: Link inertial properties

Contact forces are functions of the contacting area which,is on TUlip, constrained to the four small rubber pressure sensors extruding at the outer ends of both feet. As a result the contact area is modelled using 8 possible contact points. Placing the origin of the inertia frame in the plane on the floor (a plane with normal $\vec{n}$) h_i can be expressed simply as the dot product of the normal and position vector to contact point i, $\vec{r}_{cp_i}$:

$$h_i = \vec{r}_{cp_i} \cdot \vec{n} \tag{3.8}$$

Forward kinematics can be used to express the position vector $\vec{r}_{cp_i}$, and therefore the gap, h, is also a function of only the generalized coordinates: $h(q)$.

Strategies for modeling the contact dynamics fall generally into 2 categories: the constraint modelling and the penalizing method. The first one describes non-smooth behaviours of contact forces that obey non-penetrating and rigidity conditions associated with the rigid body dynamics. The second one allows some interbody penetration which simulates compliant bodies by means of virtual spring and damper elements. Modeling choices can significantly affect computation time, modeling complexity and physical resemblance. Both techniques are therefore studied in depth. The constraint method, requires solving differential algebraic equations, DAE and is described in more detail in Appendix C.2.

3.3.1 Penalizing method

A penalizing method is used to compute the normal reaction forces by attaching virtual spring and damper elements to the contacting surfaces. In contrast to the constraint method approach, the contact forces becomes a function of the state vector and can therefore can be eliminated to obtain a system of ordinary differential equations. A regularized[1] friction model approximates the stick - slip transition using the regularization parameter, $\dot{x}_{tol}$. This removes the infinite gradient, $\frac{d\dot{x}}{d\lambda}$ at $\dot{x} = 0$, which is illustrated in figure 3.6,A, and approximates this with a gradient $\frac{d\mu F_N}{d\dot{x}_{tol}}$, which is illustrated in figure 3.6,B.

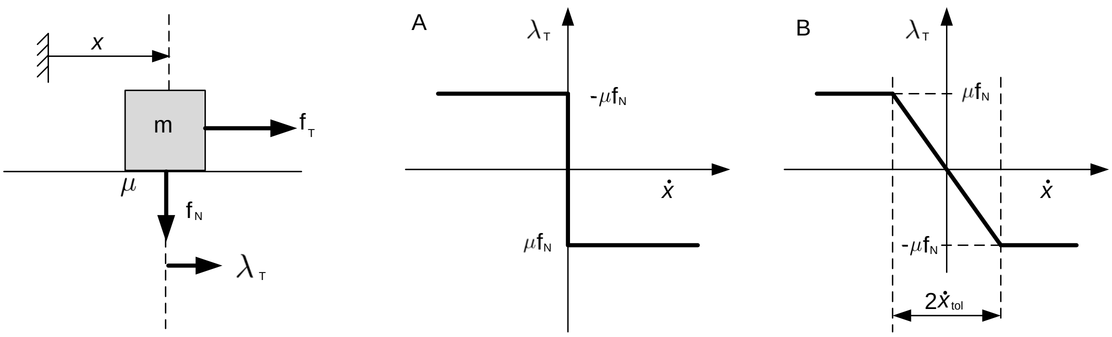


Figure 3.6: Left: Block which can slide or stick on a table. Outmost at the right-hand side: A) Dry Coulomb friction model B) Regularized friction model

The approach ensures the existence and uniqueness of a solution which is not always the case when using set valued force laws [27]. Another advantage is that contact forces are regarded as external forces making this approach easily implementable.

The penalizing method does have a number of drawbacks:

[1]Regularization is a method of dealing with infinite divergent expressions by introducing an auxiliary concept of a regulator ϵ [27]. Correct physical result is obtained in the limit in which the regulator goes away: $\epsilon \to 0$.

- It introduces very large stiffness into the system leading to stiff differential equations.
- Oscillating behaviour is often noticed, especially on the acceleration level [3].
- Sticking does not respect Coulomb's law. Any non zero tangential force always results in slipping. The biped model would, for example, therefore always slide down an inclined slope.

After careful investigation, event-driven integration, although expected to be the best choice in terms of accuracy of simulation, was rejected as the simulation time turned to be too large. Implementing the model using a time-stepping scheme, for example using OpenDE, as described in Appendix C.2, was also rejected because of limited accuracy . Use of more sophisticated implementation of the time stepping method in Matlab was rejected, too, due to algorithmic and computational complexity. Instead, SimMechanics toobox of Matlab was chosen as the modelling and simulation platform.

Normal contact force

At the beginning of the modeling phase, the Kelvin Voigt linear spring-damper model [25] was considered for the contact model. It was later on rejected as it produces non-smooth friction force behavior due to the non-zero initial velocity of the foot coming into contact with the floor. Hunt and Crossley proposed a model in [27] for which the damping term is also a function of the penetration depth, thereby solving the abovementioned issue with the Kelvin Voight model. The Hunt and Crossley model consists of nonlinear spring and damper elements, where the contact force λ_N equals:

$$
\begin{aligned}
\lambda_N &= \lambda_{N_K} + \lambda_{N_D} \quad if\ h < 0 && (3.9)\\
&= \lambda_{N_K} + e_r \lambda_{N_D} \dot{h} && (3.10)
\end{aligned}
$$

Forces λ_{N_K} and $\lambda_{N_D} = e_r \lambda_{N_D} \dot{h}$ are due to the contact stiffness and damping respectively, while e_r is the coefficient of restitution. In model (3.9), the stiffness term is calculated using Hertz theory [40], which states that for two contacting flexible spheres the contact force as a function of the stiffness of the flexible spheres is:

$$\lambda_{K_N} = \tfrac{4}{3} E^* R^{*1/2} |g|^{3/2} = K_N |h|^{3/2} \tag{3.11}$$

where $K_N = \frac{4}{3} E^* R^{*1/2}$, and h is the gap distance. A value of E^* is derived from the combined material properties of the contacting bodies:

$$\frac{1}{E^*} = \frac{1 + \mu_1^2}{E_1} + \frac{1 + \mu_2^2}{E_2} \tag{3.12}$$

where μ is the Poison ratio and E the Modulus of elasticity. Parameter R^* is a function of the curvature of the contacting bodies:

$$\frac{1}{R^*} = \frac{1}{R_1} + \frac{1}{R_2} \tag{3.13}$$

where R_1 and R_2 are defined as in figure 3.7.
Notice that for a flat surface: $R_2 \to \infty, R^* \to R_1$

The normal contact force therefore becomes:

$$\lambda_N = K_N |g|^{3/2} + e_r K_N |g|^{3/2} \dot{g} \tag{3.14}$$

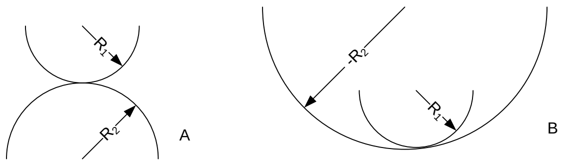


Figure 3.7: Illustration of how to apply formula for calculating curvature radius of two contacting bodies

Tangential contact force

A smooth approximations of the sign(x) function is needed to approximate dry Coulomb friction. Several approximation were considered:

- Linear approximation, $S_1 = Kx$, which inbetween $\pm x_{tol}$, approaches sign as $x_{tol} \to 0$ (or as $K \to \infty$).
- $S_2 = \tanh(Kx) = \frac{e^{2Kx}-1}{e^{2Kx}+1}$, which approaches sign(x) as $K \to \infty$.
- $S_3 = \frac{x}{\sqrt{x^2+\epsilon^2}}$. which approaches sign(x) as $\epsilon \to 0$.

The maximum function stiffness $(dS_i/dx|_{x=0})$, needs to be chosen by a trade off: it should be high enough for reasonable approximation of the sign function, but also not so high to slow down the simulation too much. For this reason it was decided to use the linear approximation, giving the smallest error, as it can be seen in figure 3.8.

This gives the relation for the tangent contact force, λ_T:

$$\lambda_T = \begin{cases} K_T \dot{x} & \text{if} |\dot{x}| < \dot{x}_{tol} \\ \pm\mu & \text{if} \mp \dot{x} > \dot{x}_{tol} \end{cases} \tag{3.15}$$

where $\dot{x}$ and λ_T are the tangent velocity of contact point and contact force respectively. see figure 3.6. The stiffness parameter $K_T = \frac{\mu}{\dot{x}_{tol}}$ is heuristically tuned using a method of trial and error for a given reference trajectory, taking into consideration model accuracies: keep K_T as small a possible, but also considering the computation speed which increases inversely proportional to K_T.

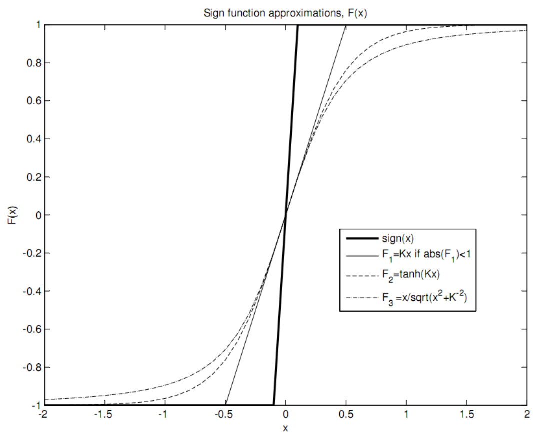


Figure 3.8: Various approximations of the sign function, with normalized maximum stiffness, $(dF_i(0)/dx = 2)$

The properties of the contacting bodies that are used in equations (3.15) and (3.14) can be found in Appendix E.

3.4 Motor Sub-system Model

The motor sub-system can be described by dynamics of the electrical and mechanical parts, as illustrated in figure 3.9. Torque delivered by the motors is a product of the motor current i and a motor constant K_m, $\tau_m = iK_m$. The reference torque, r_τ therefore needs to be transformed into a reference current, $r_i = r_{\tau_m}/K_m$. The first problem, however, is that power, supplied by the batteries is a voltage source rather than current source. Consequently a current feedback loop is needed to follow the reference current input to the motors.

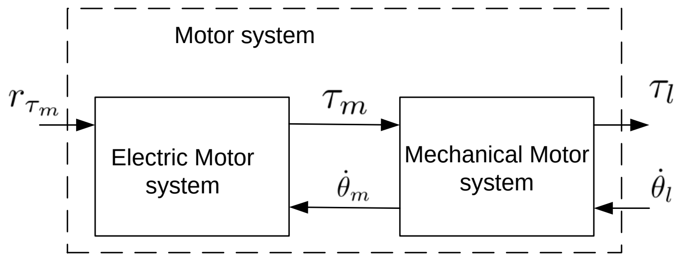


Figure 3.9: The motor sub-system split into its electric and mechanical parts, where r_{τ_m} is the reference input torque, τ_l and τ_m are the load and motor torques, respectively, and $\dot{\theta}_l$ and $\dot{\theta}_m$ are the angular velocities of load and motor shafts, respectively.

3.4.1 Electrical motor dynamics

To derive the motor dynamics, we first look at the circuit diagram for an armature controlled DC motor in figure 3.10.

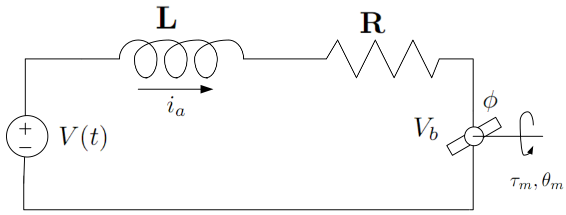


Figure 3.10: Circuit diagram for an armature controlled DC motor. Parameters R and L are the rotor winding inductance and resistance respectively. The applied voltage, V, is the control input and V_b is the back electromotive force (EMF).

Whenever a conductor moves in a magnetic field, a voltage, V_b is generated across the terminals that is proportional to the velocity of the conductor in the field. The differential equation for the armature current is therefore

$$V = L\dot{i}_a + Ri_a + V_b. \tag{3.16}$$

Voltage,V_b, known as back EMF, opposes the current flow in the conductor, according to

$$V_b = K_b\dot{\theta}_m. \tag{3.17}$$

where K_b is a motor constant called the back EMF constant. The motor torque is a function of the current, $\tau_m = iK_m$, where K_m is a motor constant. If SI units are used for parameters K_m and K_b, it can be shown that the numerical values of these parameters are the same, although there units are different. Therefore, we can use equations (3.16) and (3.17) to express the motor torque as a function of the voltage and motor velocity. This is shown in figure 3.11.

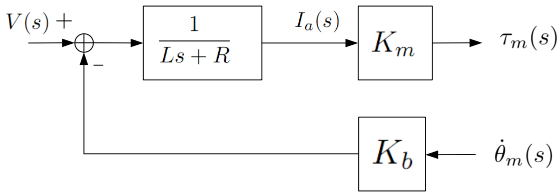


Figure 3.11: Model of an electrical motor

Power is supplied by batteries that deliver a constant voltage. In order to control the power delivered to the motors, this voltage needs to be modulated in amplitude by switching it on and off. The process responsible for converting the power delivery is known as the power processing unit, PPU, which is explained in Appendix B. In this process, output of a controller vc is amplified by a constant factor K_{PWM}, thus $V(t) = v_c(t)K_{PWM}$. As the torque is a function of current, rather than voltage, a PI controller is used to track the current reference. The electric motor model, as shown in figure 3.9 can now be presented as in figure 3.12

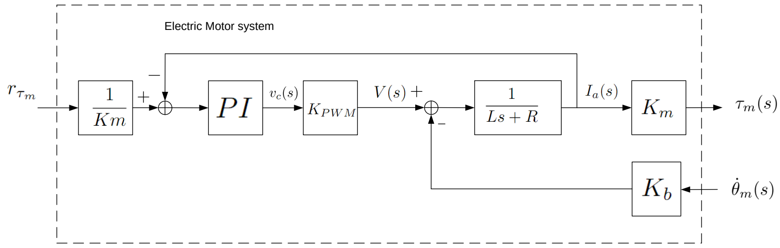


Figure 3.12: Electric motor model.

The current feedback system suggests that τ_m tracksr_{τ_m} within a particular servo control bandwidth.

3.4.2 Mechanical motor model

The dynamics is modelled to better understand the influence of the $\dot{\theta}_m$ term. On TUlip, as depicted in figure 3.13, gear reduction is used rather than a direct-dive actuation to reduce the dynamic coupling among the joints. However, the gearboxes increase friction while introducing backlash and compliance in the motor drive-trains.

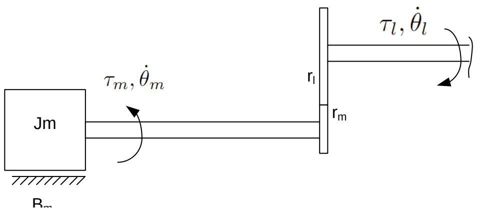


Figure 3.13: Lumped model of a single link with actuator / gearbox drive-train. Parameters J_m, B_m are the motor inertia and coefficient of motor friction, respectively. Gearbox ratio equals $N = r_l/r_m$

The dynamics is determined by splitting the mechanical sub-system into its components, as shown in figure 3.14.

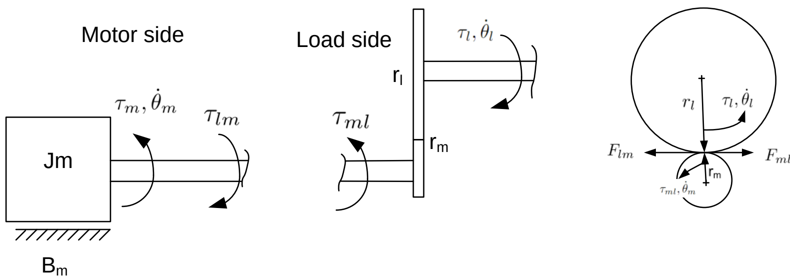


Figure 3.14: Motor model separated into subsystems. On the left-hand side, labelled motor side, the motor consists of inertia and motor damping elements. Variable τ_{lm} is the reflected torque from the load side. The middle figures, labelled load side, represents the gearbox. Variable τ_{ml} is the torque exerted by the motor onto the load side (equal and opposite to τ_{lm}). On the right-hand side, forces F_{ml} and F_{lm} are shown that are the equal to each other.

For the motor side, we can write:

$$\tau_m = J_m\ddot{\theta}_m + B_m\dot{\theta}_m + \tau_{lm} \tag{3.18}$$

where τ_{lm} is the load torque reflected onto the motor side. Neglecting frictional losses, backlash and slippage, we can derive velocity and force constraints at the point of gear contact:

$$\begin{aligned} \dot{\theta}_m r_m = \dot{\theta}_l r_l \Rightarrow \quad \dot{\theta}_m &= \frac{r_l}{r_m}\dot{\theta}_l \\ &= N\dot{\theta}_l \end{aligned} \tag{3.19}$$

$$\begin{aligned} F_{lm} = F_{ml} \Rightarrow \quad \frac{\tau_l}{r_l} &= \frac{\tau_{ml}}{r_m} \\ \tau_{ml} &= \frac{1}{N}\tau_l \end{aligned} \tag{3.20}$$

where the gear ratio equals $N = \frac{r_l}{r_m}$. Differentiating (3.19) in time gives:

$$\ddot{\theta}_m = N\ddot{\theta}_l \tag{3.21}$$

After substituting (3.20) and (3.21) into (3.18), we obtain:

$$\begin{aligned} \tau_m &= J_m N\ddot{\theta}_l + B_m N\dot{\theta}_l + \frac{1}{N}\tau_l \\ \Rightarrow \tau_l &= N\tau_m - J_m N^2\ddot{\theta}_l - B_m N^2\dot{\theta}_l \end{aligned} \tag{3.22}$$

Equation (3.22) can now be used to express the mechanical motor system as in figure 3.15

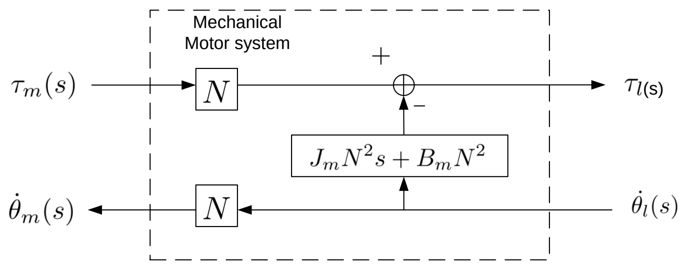


Figure 3.15: Mechanical motor model

3.4.3 Torque control

Using equation (3.22), the current and velocity feedbacks can be expressed as in figure 3.16A. As for TUlip N = 100, τ_L can be neglected and the velocity feedback term can be expressed as in figure 3.16B. Since the velocity feedback is opposite proportional to N^2, it can also be considered as negligible, leading to further simplification of the torque control loop which is illustrated in figure 3.16C.

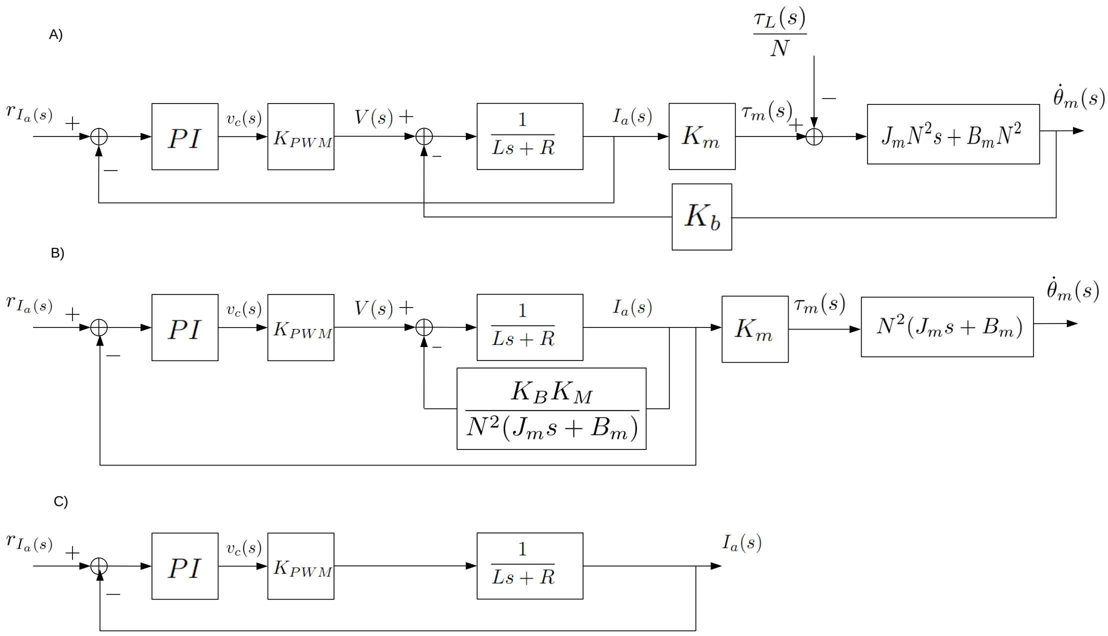


Figure 3.16: torque control loop

The open-loop current transfer function, $G_{I,OL}$, can now be expressed as

$$G_{I,OL} = PI \cdot K_{PWM} \frac{1}{Ls+R} \tag{3.23}$$

$$= \frac{k_I}{s}(1 + \frac{s}{k_I/k_p})K_{PWM} \frac{1/R}{1+s\tau_e} \tag{3.24}$$

where the PI- control law is, $PI = \frac{k_I}{s}(1+\frac{s}{k_I/k_p})$ and the motor time constant equals $\tau_e = L/R$. On TUlip, for stability robustness, safe control gains are set as kp = 1, kI = 0, giving the closed-loop frequency response shown in figure 3.17.

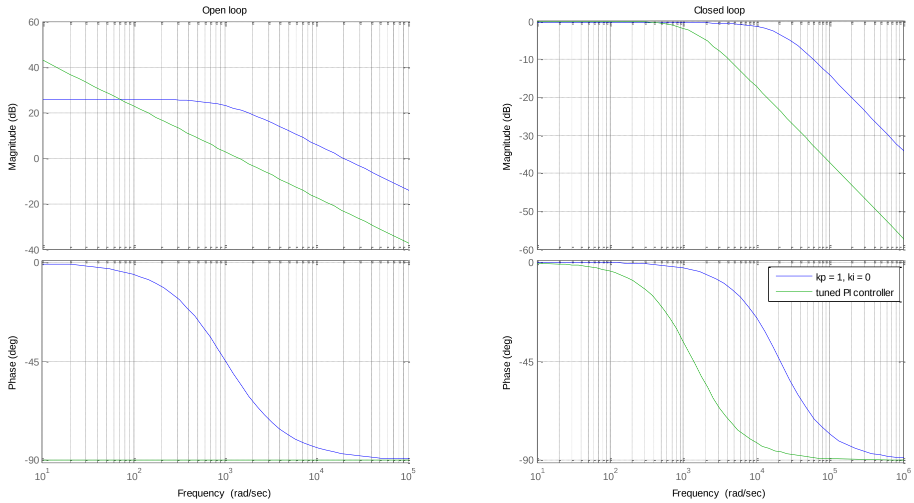


Figure 3.17: Calculated frequency response of the current control loop before and after tuning

To reduce computational costs, it is decided to neglect the current control-loop from the overall model, thereby considering that the reference torque is equal to the actual one:

$$r_{\tau_m} = \tau_m \tag{3.25}$$

This reduces the motor model shown in figure 3.9 to the one depicted in figure 3.18.

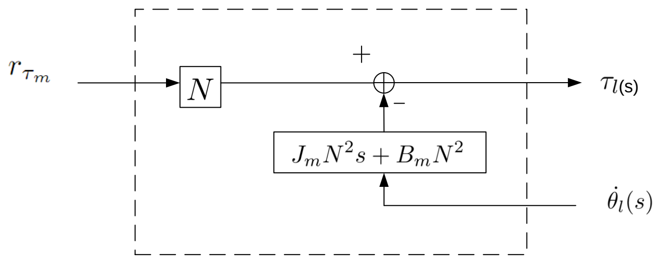


Figure 3.18: Reduced motor / gearbox model

3.4.4 Summary and conclusions on the motor dynamics

By closely looking at the motor sub-system, it can be concluded that the control bandwidth of the current control-loop is high enough to be neglected from the model. This reduces the motor model only to the static gain term from the gear box and damping and inertial terms of the motor. The resulting simplified model is shown in figure 3.18.

3.5 Software package implementation

After having considered advantages and disadvantages of contact modelling and integration methods, a critical look is given to the types of simulation environments that are available. Matlab/simulink/SimMechanics was chosen because of its friendliness, the feasibly of knowledge transfer and the flexibility it proved for maintaining and adapting the model.

3.5.1 Implementation in SimMechanics

The SimMechanics model is built up from 10 different objects illustrated in figure 3.19.

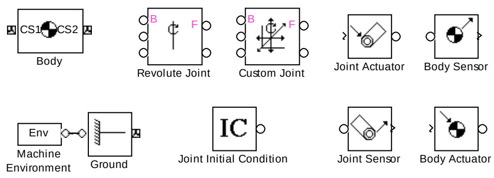


Figure 3.19: SimMechanics object blocks used to build model

Topology

Implementation in SimMechanics poses no restriction on choosing which robot link should represent the robot base. Featherstone showed in [12] that the inertia matrix of a branched kinematic tree in general has a lower condition number than that of a single kinematic tree structure. This is significant as it is shown in Appendix C.1) that the integration error is directly proportional to the condition number of the inertia matrix. For this reason the torso is chosen to be the floating base link leading to the topology shown in figure 3.20

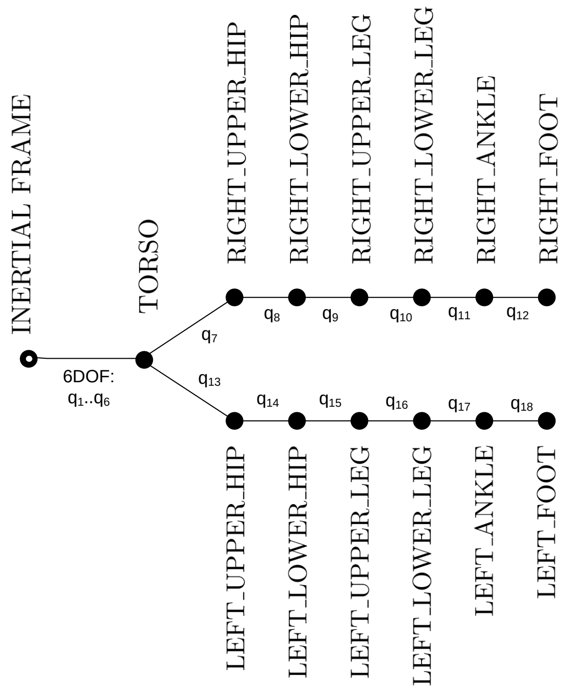


Figure 3.20: Connectivity graph of the TUlipmodel implemented in SimMechanics

The implementation of the SimMechanics model is consists of 3 parts, namely, the floating base, leg and feet subsystems.

Floating base link

The machine environment block is used to define X-Y-Z world frame and gravity vector. Next, a ground block is needed to define a fixed point in the space for connectio to a joint object. A custom joint is used whose 3 mutually orthogonal axes of rotation and 3 axes of translation are defined with respect to the world coordinate frame. Variableq_1, q_2 and q_3 denote X, Y and Z displacements and variables q_4, q_5 and q_6, the rotations about X,Y and Z axes respectively. As Particular rotational variables, the role, pitch and yaw angles are chosen. The B and F on the joint object, see figure 3.19 , stand for bass and follow-er, indicating the direction of the joint in accordance with the right hand rule. The direction is set in accordance with the direction of the kinematic chain. The final step is adding a body object representing the torso, but before this could be done, coordinate frames needed to be assigned to the body. The origins of thse frames are located at the connection ports, labeled CS_i,, where i is the number of the coordinate frame defined. Per body, at least 3 coordinate frames have to be assigned:

- Ψ_{in},the point which connects to the follower side of the joint. This is defined as having an origin at the same place as the coordinate fame attached to the base side of the joint, with orientation equal to that of Ψ_{COM}.
- Ψ_{COM} has the origin at the center of mass. Inertial properties of the body are expressed in this frame. The initial orientation is by default equal to the orientation of the world coordinate frame
- Ψ_{out} has the origin at the connecting point with the next joint in the kinematic chain. The orientation of Ψ_{out} is also equal to the orientation of Ψ_{COM}.

For the torso body, Ψ_{out_L} and Ψ_{out_R} are defined, that connect to the left- and right legs, respectively.

Leg

The leg subsystem consists of 4 repeated joints - mass combinations that describe how the the kinematic chain from the upper hip down to the ankle is attached to the preceding body.

Foot

A joint - mass combination is added to the end of the chain. The body rep- resenting the feet has four extra coordinate frames at the vertices of the feet that identify the points of contact to the ground.

The initial conditions blocks are added to all the joints. Also, the sensor and actuator blocks are added to all the actuated joints and also the body force and body sensor blocks are added at the contact points.

3.5.2 Conclusions on implementation of the equations of motion

Because of its user friendless and accuracy, the software package SimMechanics is chosen for modeling and simulations of the robot multi-body dynamics. Equations (3.22) and (3.25) describing the motor dynamics and equations (3.14) and (3.15) describing the contact model are also implemented in Matlab/Simulink/SimMechanics. The resulting robot model can be found in Appendix F.

3.6 Model validation

To evaluate quality of the robot model, the simulated joint angles and joint forces are compared to the actual ones that are measured on physical TUlip robot for the same reference trajectory and controller parameters. Two test configurations are used. In the first test, the biped was placed with its feet on the floor and it executed a forward walking gait designed by dr. D. Kostic [9]. To prevent it from falling over, the biped was manually supported by several gentle touches. In the second test, TUlip was hanging on a stative with no contacts with the ground. While being suspended by cables, the biped was performing the same gait as before. The method used for measuring the motor torques is described in Appendix H. The joint rotations in these experiments were defined in accordance with the kinematic model shown in figure 3.21.

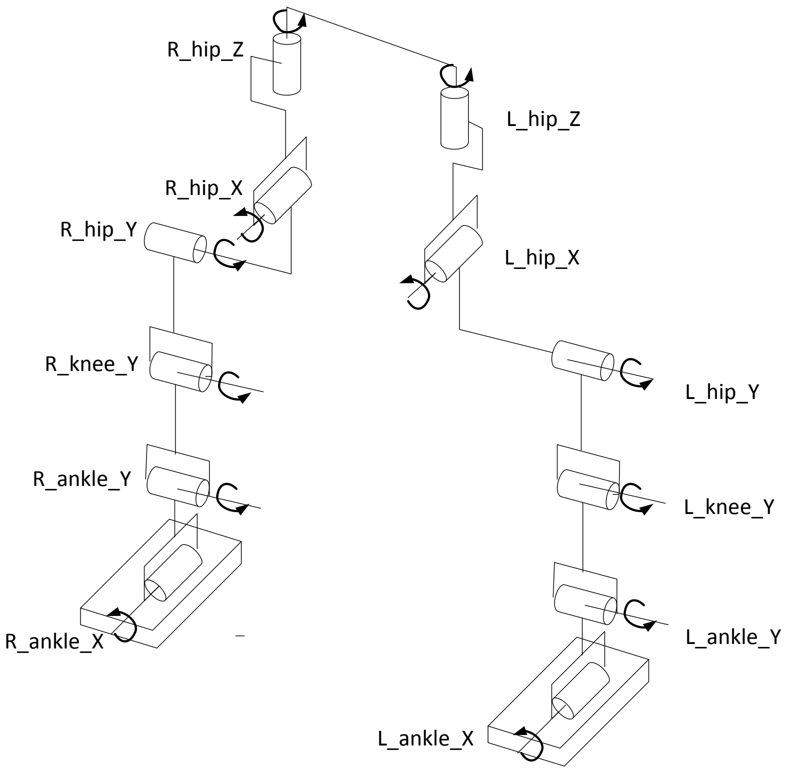


Figure 3.21: The adopted definition of the joint angles

3.6.1 Balancing configuration

The reference joint trajectories and the achieved motor and joint rotations (after gearboxes) were measured. These are shown in figure 3.22. Note that the motor rotations were scaled using the joint gear box ratios. Note that the motor rotations are mapped to the joint side by dividing them by the gear box ratios.

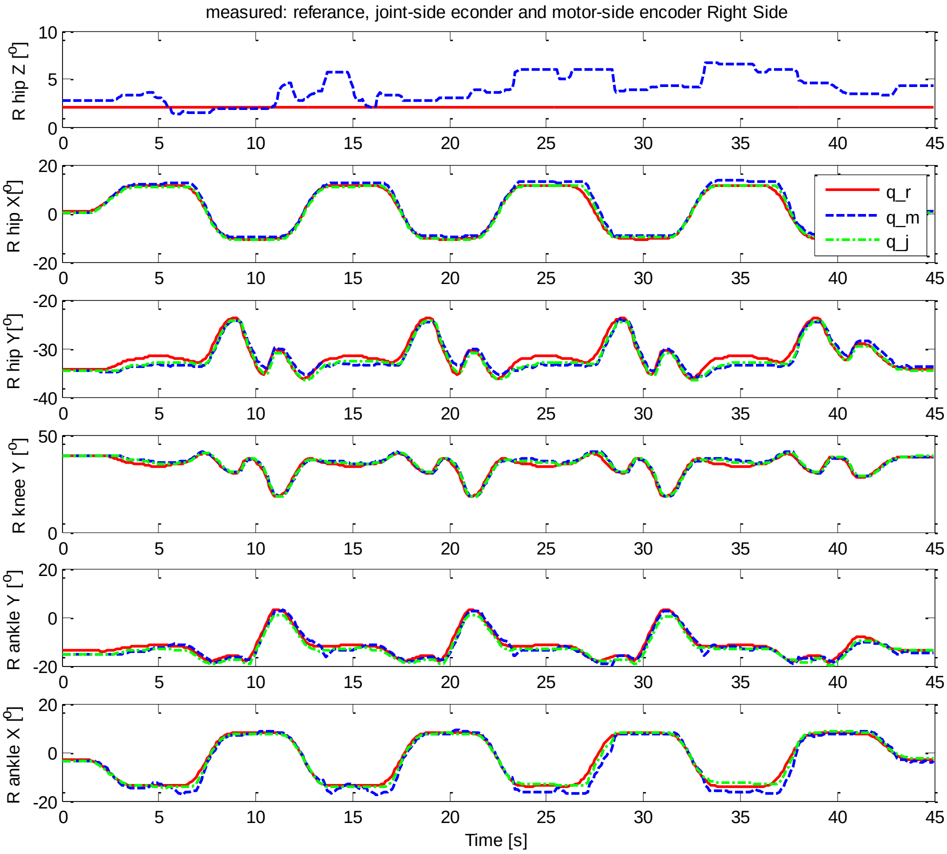


Figure 3.22: Recorded reference, joint and motor angles of the right leg, when the biped preforms a straight forward walking gait. Assistance was given to prevent the biped from falling down. The motor angles are reflected to the joint side.

When performing the forward walking test, manual assistance was needed to prevent the biped from falling. At the time of the testing, TUlip suffered some mechanical issues that prevented it to remain balanced. Before these issues arose, TUlip was able to perform the same walking gait without any human intervention. The issues were caused by wear and tear due to extensive robot usage at the art and technology festival STRP 2010. By inspection of figure 3.22, one can notice differences between the actual motions in the R ankle X joint at the motor and joint sides. These differences can be attributed to a backlash phenomenon in the drive train of this joint. The same backlash phenomenon is also the cause of large jumps of the torque measured in this joint that can be observed in figure 3.23.

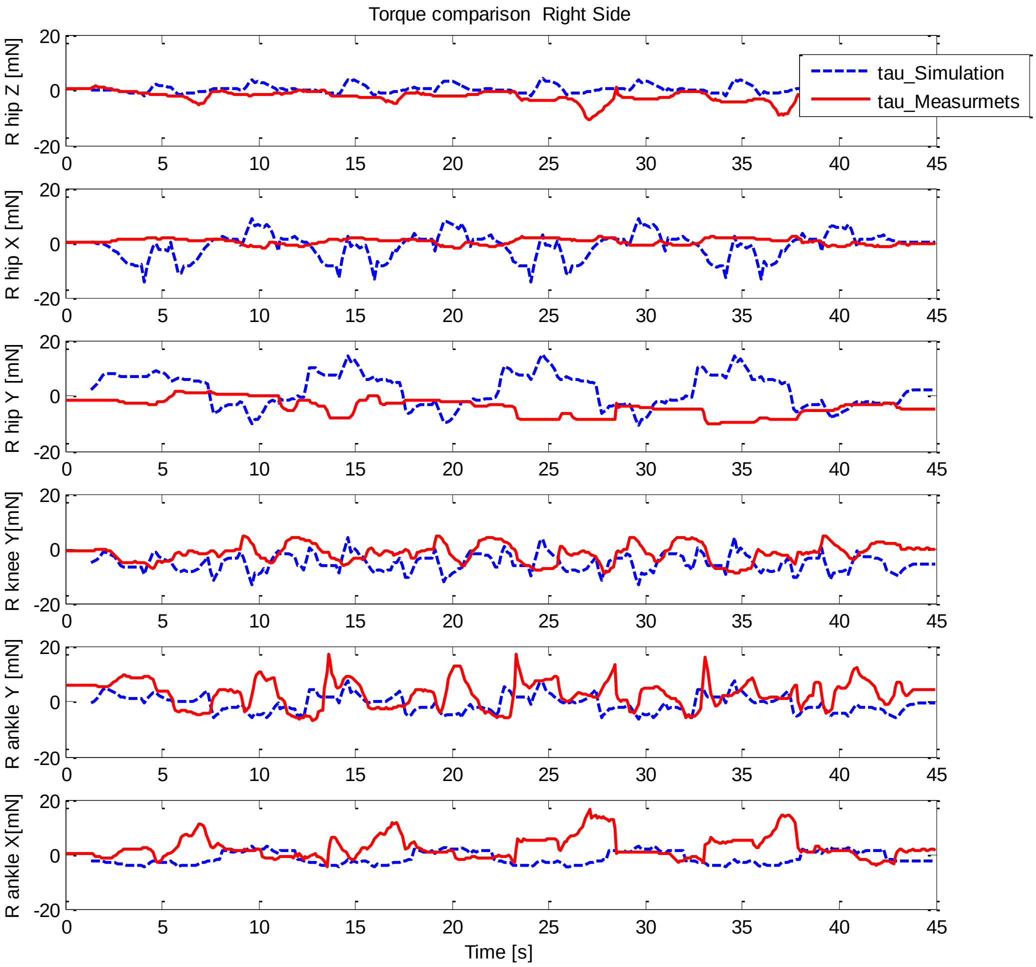


Figure 3.23: The measured and simulated joint-side torques when the biped preforms a straight forward walking gait.

The largest tracking error appears in the R hip Z joint. Figure 3.24 shows the torque measured in the R hip Z joint together with the measured position error in this joint (difference between the reference and actual joint motions), By closer inspection, one can notice a correlation between the measured torque and the position error. That correlation can be explained by the fact that the controller gains were simply too low. In particular, the controller gains for the R hip Z joint were an order of 10 lower than the controller gains of the other robot joints. This was realized only after the experiments have been finished, after which the biped was unavailable for further testing due to a longer repair and maintenance period.

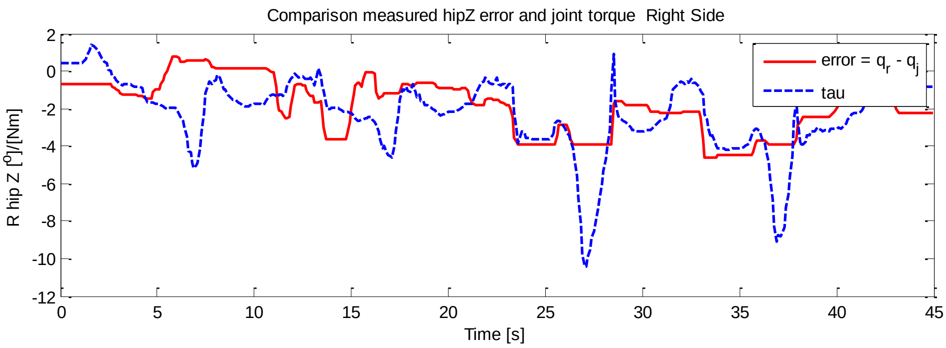


Figure 3.24: The torque and position error in the hip Z joint on the right-hand side of the biped. These values were measured as the biped preformed a straight forward walking gait.

By comparing the simulated and measured torques, one can notice very different profiles although these profiles have the similar order of magnitude. The differences could be caused by the manual assistance and by friction in the robot joints which is not captured by the SimMechanics model. Repeating the walking gait while TUlip is hanging on the cables eliminates the need for assistance. Also, the gravity loading on the robot joints is much lower which reduces the frictional forces in the joints.

3.6.2 Hanging configuration

Due to the mechanical and balance issues mentioned above, the measurements were repeated while TUlip hung on a stand with two chains attached to the torso. This hanging situation was modeled by replacing the 6DOF joint connecting the torso to the inertial frame (see figure F.4 in Appendix G) with a hinge joint.

The measured and simulated joint angles for the hanging robot configuration are shown in figure 3.26. The reference trajectory is not visible as it coincides with the simulated and measured joint trajectories. Hence, the position errors are much lower for the hanging configuration.

The simulated and measured joint torques are plotted together in figure 3.27 and then in figure 3.28 the simulated torques are shown separately. for convenience. By inspection of figure 3.27, still large differences between the measurement and simulation results can be noticed. Same as for the walking configuration, these differences were likely caused by the friction forces.

The joint torques for the simulated and measured system were plotted first together (figure 3.27) and then separately (figure 3.28), for clarity. The figures show that there is still a large difference in joint torques values between those computed in simulation and those measured on the biped. This again is thought to most likely caused by frictional forces.

3.6.3 Conclusions

The simulation results achieved with the developed Matlab/Simulink/SimMechanics model of TUlip show deviations with respect to data measured on the physical robot. It is expected that friction in the robot joints which is not included in the model is the main cause of the observed difference. Hence, it can be concluded that the model cannot directly be used for tuning parameters of the robot motion controllers. That is why it is recommended to be include the joint friction in the robot model and identify the fiction parameters in direct robot experiments. The other forces influencing the robot multi-body dynamics, namely the inertial, Coriolis/centripetal and gravity forces, are captured by the model and their parameter have been identified and validated by Pieter van Zutven in [44]. Consequently, despite absence of the joint friction, the developed Matlab/Simulink/SimMechanics model of TUlip can be considered good enough for the development and analysis of the sidestepping gait.

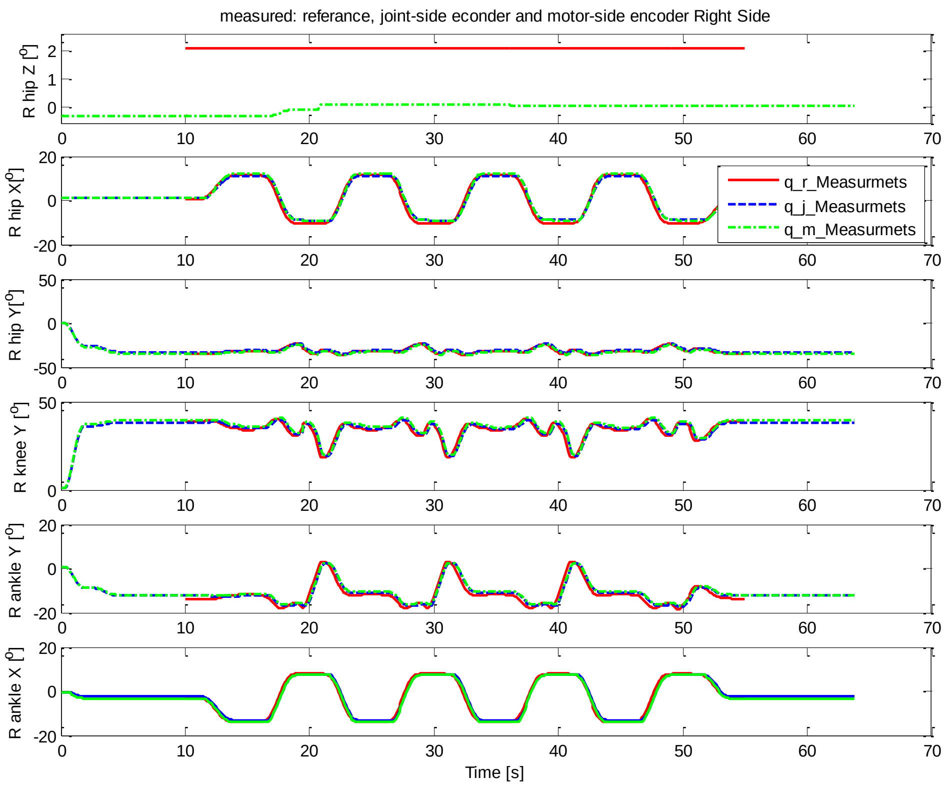


Figure 3.25: Reference, motor and joint encoder trajectories for the hanging robot configuration

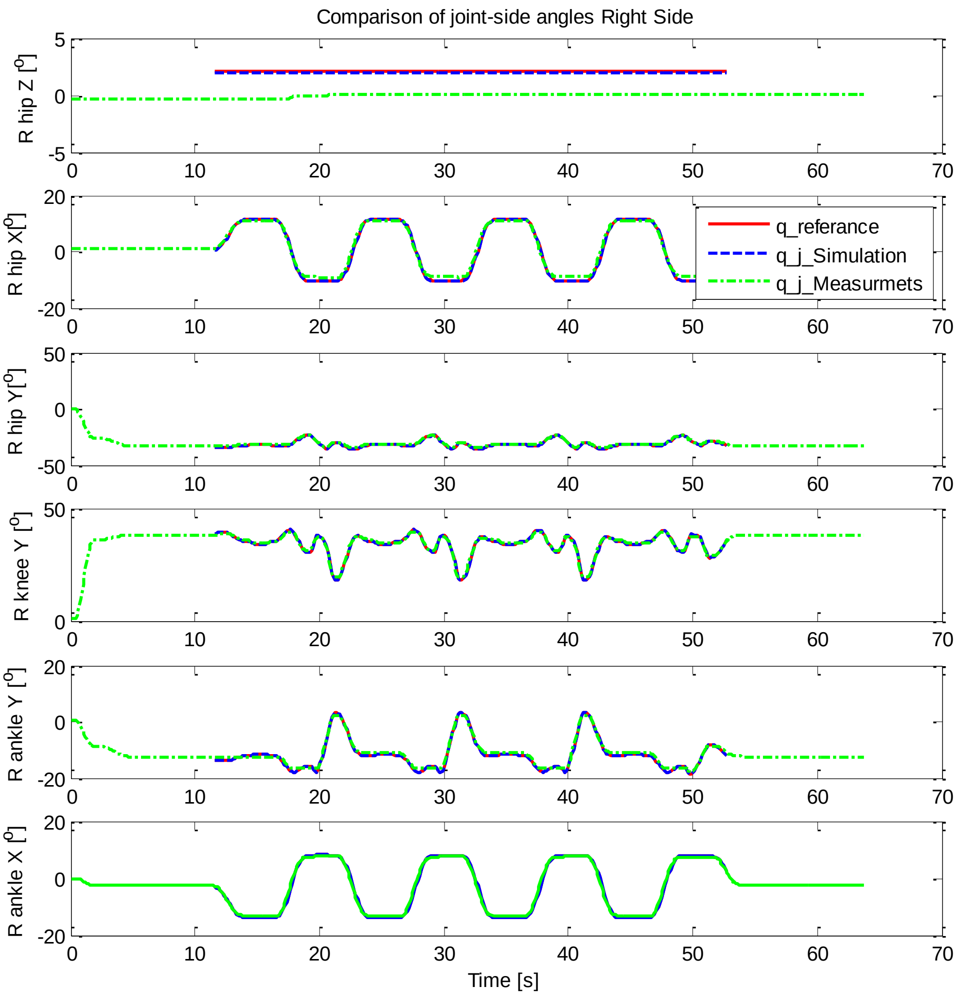


Figure 3.26: The measured and simulated angles in the joints on the right hand side of the biped for the hanging robot configuration.

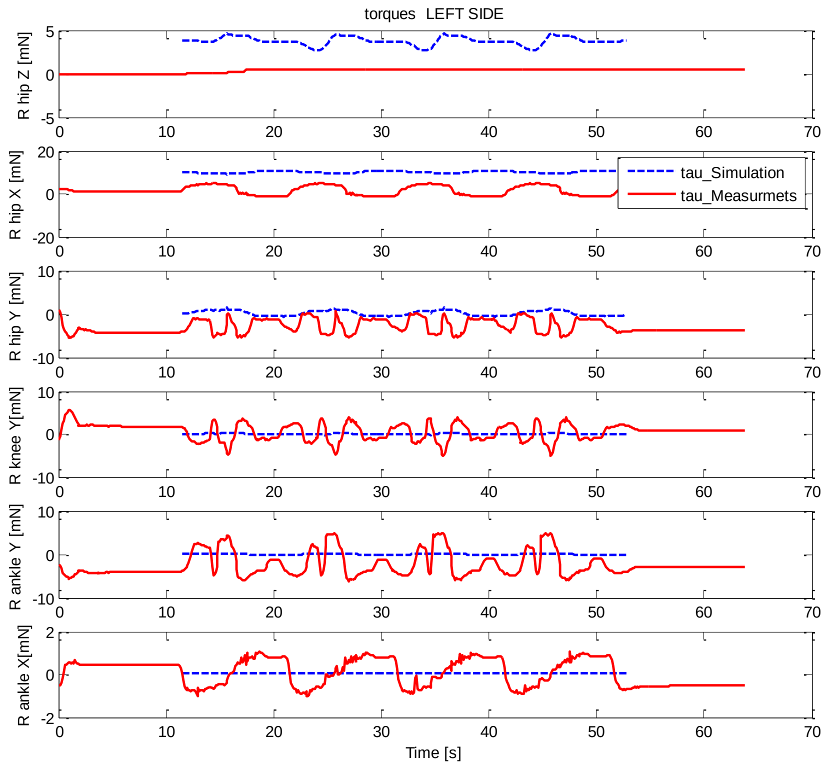


Figure 3.27: The measured and simulated joint-side torques in the joints on the right hand side of the biped for the hanging robot configuration.

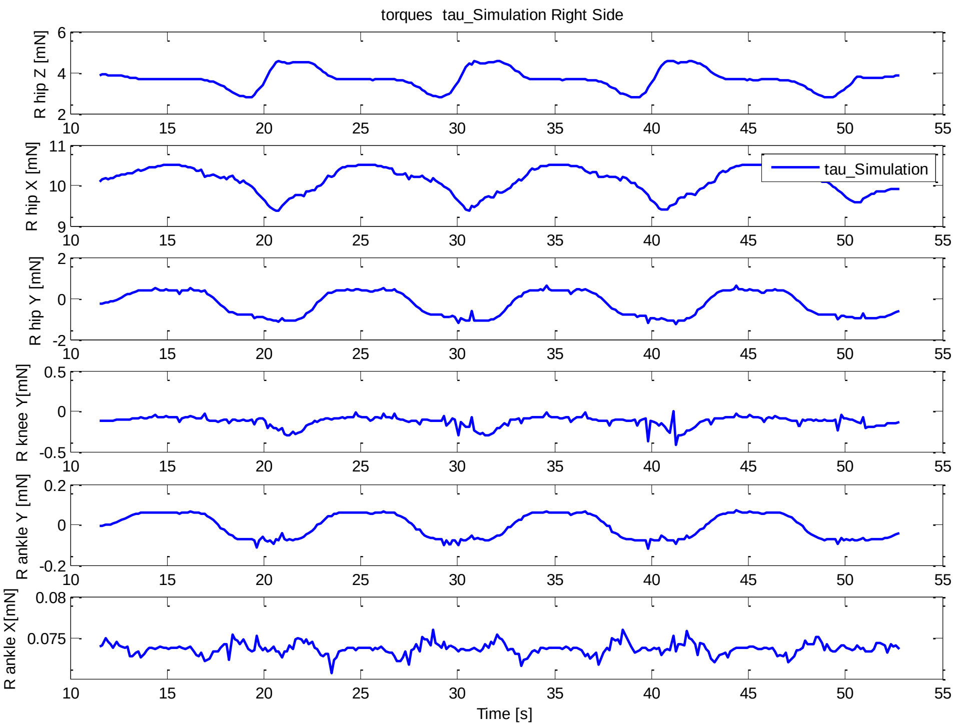


Figure 3.28: Simulated joint-side torques in the joints on the right hand side of the biped for the hanging robot configuration for the hanging robot

Chapter 4

Sidestep planning and control

4.1 Introduction

The main objective of this work and the topic of this chaotor is to design a timely sidestepping biped gait. That can be achieved if the robot reaches a location before a certain time. An assumption is made that all inertial and kinematic parameters of the biped are known. The following two requirements have therefore been formulated:

- Requirement 1: Generate the gait that brings the robot to the required position earlier than a given time.
- Requirement 2: The gait must not compromise balance of the biped. Even stricter, a high margin on the balance is desired.

The balance margin mentioned in the second requirement refers to gait stability, a measure of the ability to sustain a gait without falling to the ground. This notion of stability notably differs from the classical definition used in control theory, namely that a system is said to be stable if its output remains a bounded function of its inputs for a given working range. A sufficient, although not necessary condition for a gait to be stable is that it remains dynamically balanced at all time. To account for this requirement the linear inverted pendulum strategy proposed by Ka- jita et al [21], for the gate computation is chosen because it allows formal synthesis of a dynamically stable gait.

4.2 Maintaining dynamic balance

4.2.1 Introduction

M. Vukobratovi first introduced the concept of using the so called zero movement point, ZMP, to control stability of humanoids in [26], 16 years prior to the first practical application of the dynamically balanced biped gait was realized in the WL-10RD robot [57]. ZMP has since then become famous for its role in the synthesis of stable gates for the bipeds [66]. Vukobratovi defined ZMP in 1972 as follows:

As the load has the same sign all over the surface, it can be reduced to the resultant force, F_p, the point of attack of which will be in the boundaries of the foot. Let the point on the surface of the foot, where the resultant F_p passed, be denoted as the Zero-Moment point[26]

This section describes the derivation of the ZMP for a general 3D multi-body dynamics model of a biped. Since these dynamics are computationally demanding and require knowledge of all robot inertial parameters, there is a need for a simplified computation of the ZMP. Under certain motion constraints ,the ZMP computation can be simplified, leading to the so called linear inverted pendulum and the table-cart model. These are often used in practice to derive a motion pattern for the center of mass which, together with a foot placement pattern, ensure a stable gait. In this chapter, the linear inverted pendulum method is going to be used to generate stable side stepping gait for a walking robot.

4.2.2 ZMP computation

To explain the notion of ZMP, let us consider the mechanism in single-support phase, with the whole foot in contact with the ground. This is illustrated in figure 4.1, where the influence of the bodies above the ankle are replaced by a force and moment F_A and M_A, respectively, acting on the ankle.

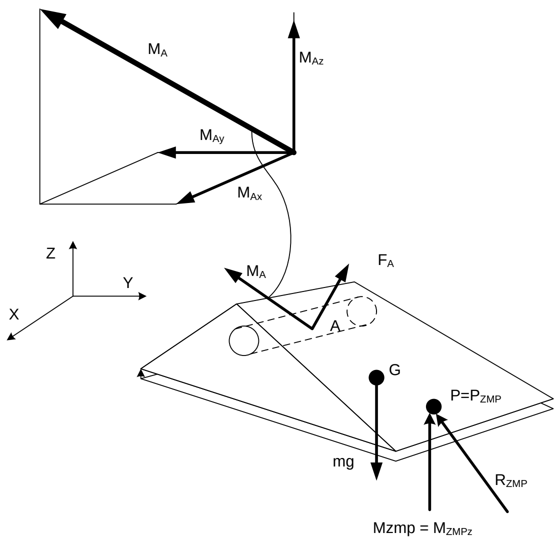


Figure 4.1: Representation of the ZMP

We can express the resultant of the ground reaction forces and moments acting on the foot at a general point, P_i, which is keeping the mechanism in equilibrium with force and moment R_i, M_i. The horizontal components of the reaction force R_{ix}, R_{iy}, are static friction forces that balance the horizontal forces of F_A, while the vertical moment of friction, M_{iz}, balance M_{Az} and the vertical moment induced by F_A. Force R_{iz} balances the vertical resultant forces F_A and gravitational force of the foot, F_G. Due to the unidirectional nature of the foot contact, all the reaction forces points towards the foot and consequently can not induce a moment. This means that the horizontal components of M_A can only be balanced by changing the position of the reaction force, P_i, to P_*, such that the moment induced by R_{iz} completely balance the applied horizontal moments M_{Ax}, M_{Ay} and those induced by F_{Az} and F_G. Since we have $P_* = P$, we can derive the following expression:

$$R + F_A + m_f g = 0 \tag{4.1}$$

$$\vec{OP} \times R + \vec{OG} \times m_f g + M_A + M_p + \vec{OA} \times F_A = 0 \tag{4.2}$$

where $\vec{OP}$,$\vec{OG}$ and $\vec{OA}$, are radius vectors from the origin of a coordinate frame to points P, G and A. Parameter m_f is the mass of the foot and g is the gravitation acceleration. Notice that if the acting moments M_A increase, it would be compensated by the term $\vec{OP} \times R$. As R stays the same, this means that the zero moment point would shift outwards with it until reaching the edge of the foot. The resultant reaction force obviously can not act outside the support region and thus any further increase in the acting moment would create a resultant moment causing rotation of the foot. The point outside the support polygon which still satisfies equation (4.2) is known as the fictional zero moment point, FZMP. Both terms, the ZMP and FZMP are often referred together as the *computed ZMP* (see [49] and [66]). The robot joint trajectories therefore need to be designed such that the computed ZMP does not leave the support polygon. If we attach a coordinate frame to point P, we can express the horizontal moments about P as:

$$\sum \begin{bmatrix} M_{Ax} \\ M_{Ay} \end{bmatrix} + \sum \begin{bmatrix} M_{Fx} \\ M_{Fy} \end{bmatrix} + \sum \begin{bmatrix} M_{Gx} \\ M_{Gy} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.3}$$

where M_{Fi}, M_{Gi} are the i component of moments about about P^* induced by forces F_A and G respectively. For an n-link biped, see figure 4.2, we can express equation (4.3) as

$$\sum_{i}^{n} \{ r_i - r_p) \times m_i \ddot{r}_i + I_i \omega_i - (r_i - r_p) \times m_i g \} = \begin{bmatrix} 0 & 0 & * \end{bmatrix} \tag{4.4}$$

where r_i,r_p are the position vectors of link i and the zmp respectively. Parameters m_i and I_i are the mass and inertia of link I, respectively.

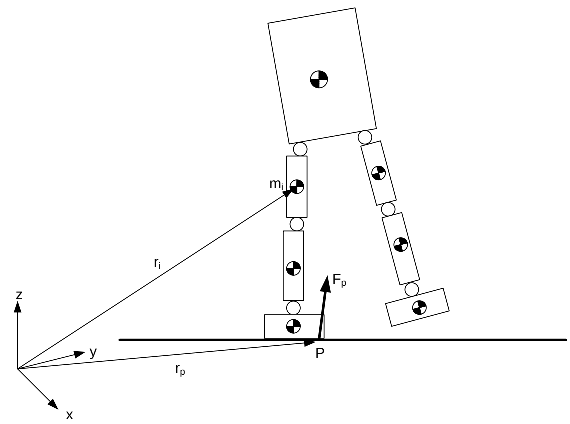


Figure 4.2: Description of link positions used to compute the ZMP, p, with equation (4.4).

Given a set of joint trajectories $q, \dot{q}$ of a particular gait and the inertial properties of the biped, dynamic balance can easily be verified. This can be done by first assuming dynamic balance is indeed sustained, in which case there is no unknown additional DoF to consider. We can differentiate the trajectory and use forward kinematics to solve (4.4) for r_p, giving either the ZMP or FZMP. Dynamic balance can then be verified by checking if r_p remains inside the support polygon. The goal for stable sidestepping is to find a particular set of joint trajectories which satisfy equation (4.4) given a particular ZMP trajectories. This is not a trivial task as it is a highly non-linear differential problem which does not give a unique solution.

4.2.3 Simplification of the ZMP and CM relation

Two single mass models, namely the so called cart-table model and the linear inverted pendulum model are investigated to help understand and derive the *ZMP equations*, [63], that give the centre of mass, CM as a function of the ZMP. The linear inverted pendulum model gives a more comprehensive insight into how the CM can be manipulated to constrain the ZMP to a certain point which coincides with the origin of the support polygon. The cart-table model is even more simplified model which relies on a constraint that the CM has one degree of motion only. The simplicity of these equations are used derived the trajectory of the CM. For completeness, the linear inverted pendulum model is described in appendix A.

4.2.4 Cart-table model

The cart-table model, illustrated in figure 4.3, is possibly the simplest model of the ZMP motion.

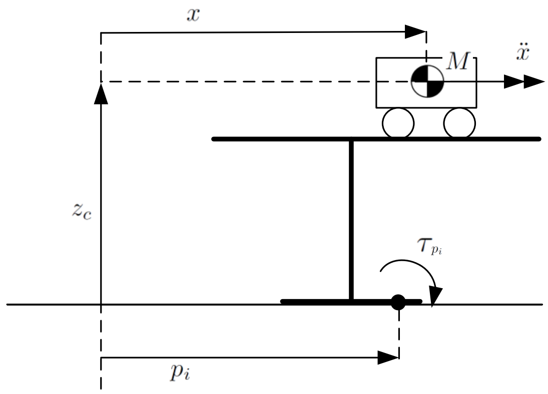


Figure 4.3: The table-kart model, a very simple model of a humanoid, used to compute the ZMP equations. M represents the total mass of the biped and z_c the hight of the center of mass. τ_{p_i} is the resultant of the ground reaction moment exerted by the ground on the 'biped' at point point p_i.

The model considers a mass, M, and position, x, of the CM of the biped, which moves on a massless table. Also, the support polygon of the table coincides with the support polygon of the biped. The resultant reaction torque,τ, at a random point p equals

$$\tau_p = -Mg(x - p) + M\ddot{x}z_c \tag{4.5}$$

where z_c is the height of the center of mass, and x is the horizontal displacement. The computed ZMP,p, where $\tau = 0$ equals

$$p = x - \frac{z_c}{g}\ddot{x} \tag{4.6}$$

Equation (4.6) can be used derive a CM motion pattern which ensures that the ZMP remains within the support polygon.

4.2.5 CM motion pattern generation

In order to determine how the CM should move to ensure the gait stability, it is necessary to consider what is the desired position of the ZMP. Designing the trajectories such that the ZMP remains at the origin of the support polygon, which coincides with the mid-point of this polygon, has three main advantages:

- Motion is energy efficient as ankle X and Y motion occur passively. This is because the actuated torques, u_x and u_y, in equations (A.14) and (A.15) in Appendix A, are equal to zero in the corresponding directions.

- The highest margin in terms of the balance. As the ankle joint is positioned more or less at the center of the foot, if the ZMP leaves the ankle position (or more accurately, the floor projection of the ankle position) the allowable error bound, $e = (zmp_{ref} - zmp)$ is maximized.
- For a fixed CM height, this will potentially maximize the side stepping speed. To explain this first notice that for side stepping locomotion it is required to accelerate and then decelerate the CM. From equation (4.6), it can be seen that the acceleration is a function of the horizontal distance the CM is from the ZMP, therefore to maximise the acceleration and deceleration the CM needs to be as far as possible from the ZMP. While taking into account the mentioned stability bound, this means that the ZMP will optimally switch instantaneously form one supporting foot to the other.

Placing the origin of the coordinate system at the mid-point of the support foot (coincides with the mid-point of the ankle joint) and by integrating equation (4.6) twice, we get

$$\begin{aligned} p = x - \frac{z_c}{g}\ddot{x} &= 0 \\ \Rightarrow \ddot{x} - \frac{g}{z_c}x &= 0 \end{aligned} \tag{4.7}$$

To solve equation (4.7), we first compute the roots of the auxiliary equation:

$$\begin{aligned} m^2 - \frac{g}{z_c}x &= 0 \\ \Rightarrow m &= \pm\sqrt{\frac{g}{z_c}} \end{aligned} \tag{4.8}$$

Therefore

$$x = C_1 e^{at} + C_2 e^{-at} \tag{4.9}$$

where $a = \sqrt{g/z_c}$. C_1 and C_2 are the constants of integration. Differentiating equation (4.9) and by applying initial conditions at $x(t_i) = x_i, \dot{x}(t_i) = \dot{x}_i$, we get:

$$\begin{aligned} \begin{bmatrix} x_i \\ \dot{x}_i \end{bmatrix} &= \begin{bmatrix} C_1 e^{at_i} + C_2 e^{-at_i} \\ aC_1 e^{at_i} - aC_2 e^{-at_i} \end{bmatrix} = \begin{bmatrix} e^{at_i} & e^{-at_i} \\ ae^{at_i} & -ae^{-at_i} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \\ \Rightarrow \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} &= \begin{bmatrix} \frac{1}{2e^{at_i}} & \frac{1}{2ae^{at_i}} \\ \frac{1}{2e^{-at_i}} & -\frac{1}{2ae^{-at_i}} \end{bmatrix} \begin{bmatrix} x_i \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} \frac{1}{2e^{at_i}} x_i + \frac{1}{2ae^{at_i}} \dot{x}_i \\ \frac{1}{2e^{-at_i}} x_i - \frac{1}{2ae^{-at_i}} \dot{x}_i \end{bmatrix} \end{aligned} \tag{4.10}$$

By substituting values of (4.10) back into equation (4.9), we obtain

$$\begin{aligned} x(t) &= (\frac{1}{2e^{at_i}} x_i + \frac{1}{2ae^{at_i}} \dot{x}_i) e^{at} + (\frac{1}{2e^{-at_i}} x_i - \frac{1}{2ae^{-at_i}} \dot{x}_i) e^{-at} \\ &= (\frac{e^{at}}{2e^{at_i}} + \frac{e^{-at}}{2e^{-at_i}}) x_i + (\frac{e^{at}}{2ae^{at_i}} - \frac{e^{-at}}{2ae^{-at_i}}) \dot{x}_i \\ &= x_i (\frac{e^{a(t-t_i)} + e^{-a(t-t_i)}}{2}) + \frac{\dot{x}_0}{a} (\frac{e^{a(t-t_i)} - e^{-a(t-t_i)}}{2}) \\ &= x_i \cosh(a(t-t_i)) + \frac{\dot{x}_i}{a} \sinh(a(t-t_i)) \\ &= x_i \cosh(\frac{t-t_i}{T_c}) + T_c \dot{x}_i \sinh(\frac{t-t_i}{T_c}) \end{aligned} \tag{4.11}$$

where $T_c = 1/a = \sqrt{z_c/g}$, is a motion time constant. By increasing z_c, which is he height of the CM, a slower motion, $x(t)$, is achieved. Equation (4.11) can be easily differentiated to find the velocity profile of the CM:

$$\dot{x}(t) \quad = \quad \frac{x_i}{T_c}\sinh(\frac{t-t_i}{T_c}) + \dot{x}_0\cosh(\frac{t-t_i}{T_c}) \tag{4.12}$$

By using (4.11), we are able to describe the relative motion of the CM with respect to the ZMP, given initial position and velocity of the CM, $x_i, \dot{x}_i$. Next, we will show how the initial conditions are derived such that the CM moves from the initial to the finish position during a single side stepping gait cycle, whiles keeping the ZMP in the ideal position, underneath the ankle of the supporting foot (the midpoint of the support polygon).

4.3 Side stepping motion phases

The side stepping motion consists of motion phases that incorporate 5 distinct postures (states) shown in 4.4. Figure 4.5 shows the state transitions the biped makes when executing a left-hand side stepping walk. For a right stepping walk the arrows indicating a state transition point to the other direction. Notice that after the transition from postures P1 to P2 to P3 to P4 and back to P1, the biped has made 1 full side step which consists of 2 half steps. These 5 state transitions can be repeated for any given number of sidesteps. The initial and final state transitions, illustrated with the dotted lines, move the biped to and from its initial posture, P0.

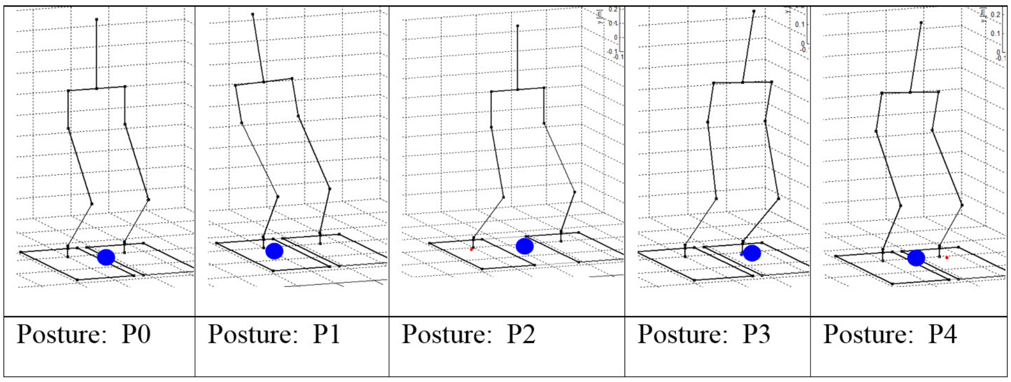


Figure 4.4: Distinct postures where the biped transitions to and from during the side stepping gait. The position of the CM projected onto the floor plain in indicated by the blue ball

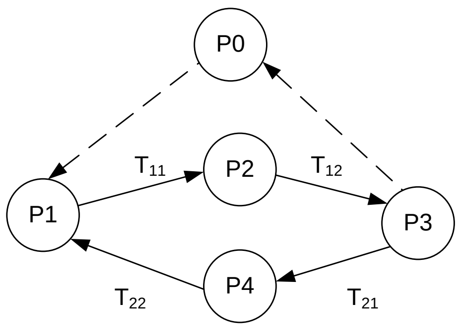


Figure 4.5: State transition diagram shown the phases the biped moves though during the left direction sidestepping. The initial and final transitions are illustrated by dotted lines

The initial and final movements are clearly less critical with respect to keeping the balance and therefore a motion trajectory with a cosine velocity profile for the CM is implemented for these phases. Equations (4.11) and (4.12) are used to describe motion of the CM which ensures that the position of the ZMP remains underneath the ankle while repositioning between the side stepping postures, P1 to P4. The following sections will show how, giving a particular side stepping step size, the position of the CM in P1 can be manipulated to produce a side stepping gaite with the required movement speed. It will also be shown that this speed is bounded and an estimate of the maximum speed will be computed.

4.3.1 Time period per motion phase

As mentioned in 4.3, the side stepping gait consists of the 4 repeating state transitions and 2 initiating state transitions. Due to symmetry in the motion, only the left to right-hand side side stepping motions are considered. Specifically the following transitions and phase times are first defined:

- Phase 01, phase period: T_{start}. The transition from posture P0 to P1. CM moves to an initial start position. At this position, there is a certain offset between CM and the supporting ankle (which is the desired position of the ZMP). If the initial offset increases, the CM needs accelerate more in order to remain balanced. During this phase, the CM moves relatively slowly according to a cosine velocity profile. A time period of 1 second is chosen which is in simulation verified as long enough for stable motion.
- Phase 20, phase period: T_{end}. The transition from posture P2 to P0. CM moves to from the final back to the start posture. A time period of $T_{end} = T_{start} = 1$ second is chosen.
- Phase 11, phase period: $T_{11} : t_0 \leq t < t_1$. The transition from posture P1 to P2. CM accelerates in such a way that the ZMP remains under the supporting ankle. The final position is half way between the feet. While the CM accelerates, the left foot makes a step.
- Phase12, phase period: $T_{12} : t_1 \leq t < t_2$. The transition from posture P2 to P3: CM decelerates back to zero velocity. The following leg makes a step. $T = T_{11} = t_0 \leq t < t_1$.
- Phase 21, phase period:$T_{21} : t_2 \leq t < t_3$. The transition from posture P3 to P4: CM accelerates back to the mid-point between legs, $T = T_{21} = t_2 \leq t < t_3$
- Phase22, phase period:$T_{22} : t_3 \leq t < t_4$. The transition from posture P4 to P1: CM decelerates back to it's initial stepping posture P1, $T = T_{22} = t_3 \leq t < t_4$.

Transitions T_{11} and T_{12} move the CM from the above the left (following)foot to the right (leading) foot. This motion is defined as the first body swing motion, which is followed by T_{21} and T_{22}, moving the CM back to the left foot which is defined as the second body swing motion, as illustrated in figure 4.7.

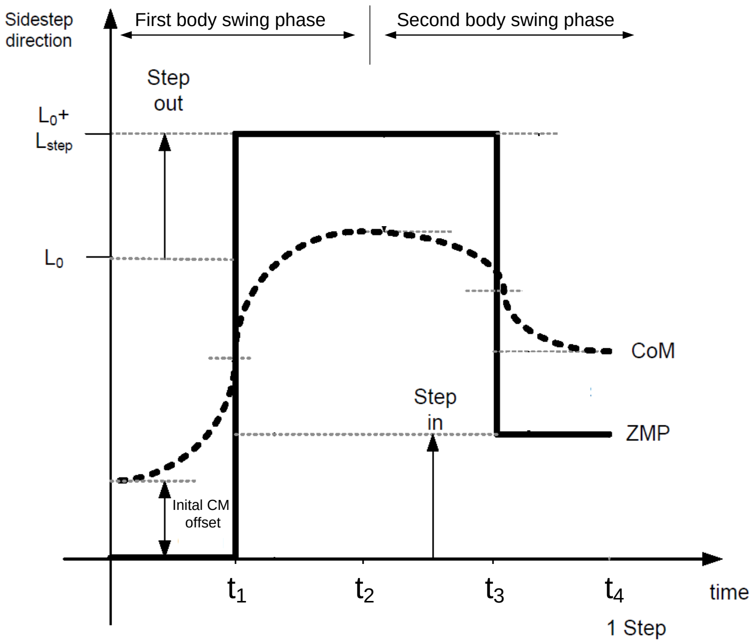


Figure 4.6: Illustration of the desired trajectory of the CM and subsequent ZMP for a single side step. L_o is the initial distance between the supporting legs, L_{step} is the stepping distance.

The motion trajectory of CM and phase times of these motions are derived next.

First swing phase

The initial CM velocity at the start of the gait is equal to zero, $\dot{x}_0 = 0$. Notice that the solution to equation (4.11), given an initial position also equals to zero: $x(t)|_{x_0=\dot{x}_0=0} = 0$. As such, the CM needs to start with an initial offset with respect to the ZMP, $x_0 \neq 0$. This initial displacement is expressed as a fraction, α_i, of the total displacement, S_i, of the CM at the end of the phase, at time $t =^- t_1$, just before switching support. Therefore:

$$x(t_0) = \alpha_1 S_1 \quad (4.13)$$
$$x(t_1) = S_1 \quad (4.14)$$

Due to symmetry in the motion, the CM at a moment of support transition,$t = t_1$, should be half way between the two extrema in the ZMP reference positions. The step in the ZMP can be expressed as a distance between the ankle before the step (the initial stance distance), L_0, plus the stepping distance, L_S:

$$\begin{aligned} x(^-t_1) = S_1 &= \frac{1}{2}\text{step}(ZMP) \\ &= \frac{1}{2}(L_0 + L_S) \end{aligned} \quad (4.15)$$

For the first part of the initial swing phase, the motion of the CM can be described as:

$$x(t) = \alpha_1 S_1 \cosh(\frac{t - t_0}{T_c}) \quad t_0 \leq t < t_1 \quad (4.16)$$
$$\dot{x}(t) = \frac{\alpha_1 S_1}{T_c}\sinh(\frac{t - t_0}{T_c}) \quad (4.17)$$

To find the phase time, we fill in $t = t_1$ into (4.16) to get :

$$\begin{aligned} x(t_1) = S_1 &= \alpha_1 S_1 \cosh(\frac{T_{11}}{T_c}) && (4.18) \\ \Rightarrow T_{11} &= T_c \ln(\frac{1 + \sqrt{1-\alpha_1^2}}{\alpha_1}) && (4.19) \end{aligned}$$

where $T_{11} = t_1 - t_0$, is the phase time period for this first part of the first single stance phase. Equation (4.19) shows that as the body swing magnitude becomes small, ($\alpha \to 1$) the period becomes shorter ($T_{11} \to 0$), and visa versa. Also, as $T_c = \sqrt{z_c/g}$, the period is directly proportional the square root of the height of the center of mass ($T_{11} \propto \sqrt{z_c}$).

At the moment of support transition, the ZMP jumps $2S_1$ in the direction of x, as defined earlier. Therefore, just after switching the support, $t =^+ t_1$, the position of the CM wrt ZMP becomes:

$$x_{+t_1} = -S_1 \tag{4.20}$$

It is assumed that no energy is lost during the impact and the initial velocity at $t =^+ t_1$ is thus

$$\dot{x}_{+t_1} = \dot{x}_{-t_1} = \frac{\alpha_1 S_1}{T_c} \sinh(\frac{T_{11}}{T_c}) \tag{4.21}$$

To find an expression for the phase time, we can fill in $t = t_2$ into (4.12), and make the resulting expression equal to zero, giving:

$$\begin{aligned} \dot{x}(t_2) \equiv 0 &= \frac{x_i}{T_c} \sinh(\frac{t_2 - t_1}{T_c}) + \dot{x}_0 \cosh(\frac{t_2 - t_1}{T_c}) \\ 0 &= \tanh(\frac{T_{12}}{T_c}) + \frac{T_c \dot{x}_i}{x_i} \\ \Rightarrow \frac{T_{12}}{T_c} &= \operatorname{arctanh}(\frac{-T_c \dot{x}_i}{x_i}) && (4.22) \end{aligned}$$

where $T_{12} = t_2 - t_1$, the phase time period. By filling in the new initial conditions,x_i and $\dot{x}_i$ into (4.20) and (4.21), we get

$$\frac{T_{12}}{T_c} = \operatorname{arctanh}(\alpha_1 \sinh(\frac{T_{11}}{T_c})) \tag{4.23}$$

Finally by expressing α_1 in terms of T_{11}, using (4.19), we get

$$\begin{aligned} \frac{T_{12}}{T_c} &= \operatorname{arctanh}(\operatorname{sech}(\frac{T_{11}}{T_c}) \sinh(\frac{T_{11}}{T_c})) = \frac{T_{11}}{T_c} \\ \Rightarrow T_{12} &= T_{11} && (4.24) \end{aligned}$$

The result (4.24) seems trivial as the motion is obviously symmetric. From now on we use $T_{11} = T_{12} \equiv 1/2T_1$, where T_1 is the time period of the first swing phase. Inserting the initial conditions into 4.11 we get the desired CM trajectory for this phase:

$$x(t) = -S_1 \cosh(\frac{t - t_1}{T_c}) + S_1 \alpha_1 \sinh(\frac{T_1}{2T_c}) \sinh(\frac{t - t_1}{T_c}) \quad \text{for } t_1 \le t < t_2 \tag{4.25}$$

Second swing phase

For the second swing phase, the initial velocity condition is, as defined, equal to zero. As for the initial position condition, x_i, can be found by filling in $t = t_2$, into (4.25) to give

$$\begin{aligned} x(t_2) &= -S_1\cosh(\frac{1/2T_1}{T_c}) + S_1\alpha_1\sinh(\frac{1/2T_1}{T_c})\sinh(\frac{1/2T_1}{T_c}) \\ &= -(\cosh(\frac{1/2T_1}{T_c}) - \alpha_1\sinh^2(\frac{1/2T_1}{T_c}))S_1 \\ &= -(\alpha_1^{-1} - \alpha_1(\alpha_1^{-2} - 1))S_1 = -\alpha_1 S_1 \end{aligned} \tag{4.26}$$

where T_c and T_1 are expressed in terms of α_1 using (4.18). Although the motion direction is reversed, the motion characteristics are similar to that of the first swing phase. Before the support switches back to the previous leg, the biped needs to take another step in the same direction and length as in the previous step. This means that the the ZMP makes a jump:

$$L_{StepSizeZMP} = L_0 + L_S - L_S = L_0 \tag{4.27}$$

The required end position of the CM, S_2, is therefore:

$$S_2 = \frac{1}{2}L_0 \tag{4.28}$$

As before, we can express the start position of the CM with respect to the ZMP as a fraction of its total displacement, α_i, which means that:

$$\begin{aligned} x(t_3) = -\alpha_2 S_2 &= -\alpha_1 S_1 \\ \Rightarrow \alpha_2 &= \alpha_1\frac{S_1}{S_2} \end{aligned} \tag{4.29}$$

The phase time $T_{21} = T_{22} \equiv 1/2T_2$, where T_2 is the total phase time of the second swing phase, can then easily be found by replacing α_1 with α_2 in equation (4.19):

$$T_2 = T_c \ln(\frac{1 + \sqrt{1 - \alpha_2^2}}{\alpha_2}) \tag{4.30}$$

Given a particular stance size, L_0 and step size L_S and the initial CM offset fraction, α_1, the complete CM trajectory for the sided step gate is now defined. This is illustrated in figure 4.7.

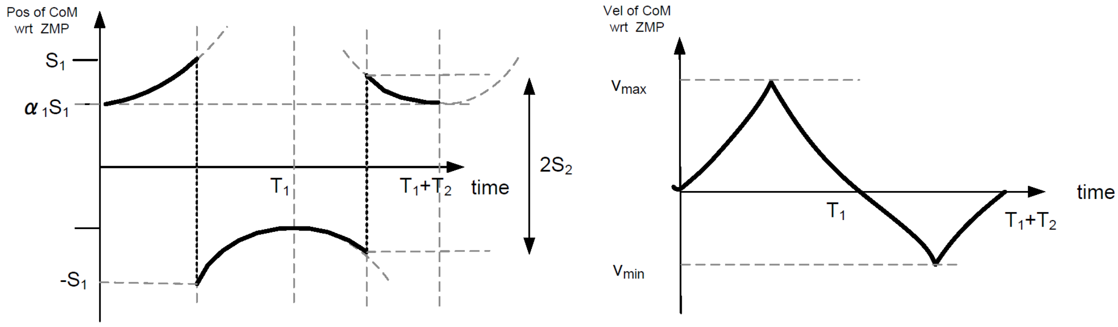


Figure 4.7: The position and velocity of the robot's CM with respect to the ZMP, when preforming an ideal side stepping motion.

4.3.2 Sidestep gait as function of time

In the previous section it has been shown how to compute the total time for a side step, given the stance and step size and initial CM offset for the supporting foot. From equation 4.19, it can be seen that α and z_c affect the phase time. It is required to arrive at a certain location within a certain time. Given the target time and distance, T_t, and distance, D_t, and maximum step size, $L_{maxstep}$, the biped can easily compute the minimum number of equidistant steps, N_g :

$$\begin{aligned} N_g &= \lceil \frac{D_t}{L_{maxstep}} \rceil \\ \Rightarrow L_S &= \frac{D_t}{N_g} \end{aligned} \tag{4.31}$$

Next, the end time available could be expressed as a function of the CM motion phase times:

$$T_t = T_{start} + N_g(T_1 + T_2) - T_2 + T_{end} \tag{4.32}$$

where $T_1$and T_2 are durations of the first and second body swing motions, T_{start} and T_{end} is the time taken for the biped to move from and back to the initial posture to the initial start posture. After the final step the biped does not need to go back to its initial stepping posture, P1, but can go straight to P0, as illustrated in figure 4.5, therefore T_2 is subtracted from the total time.

Equations (4.19) and (4.35) and (4.29) are used to express $T_1$and T_2 in terms of α_1,an initial stance, L_0, stepping size L_{step} and CM height, Z_{cm}. After taking T_t to the right hand side of the equation (4.32), this equation could be solved for α_1 using a root finding algorithm, for a given L_0, L_{step}, N_g, Z_{cm} and T_t. The stepping speed could therefore be controlled by controlling α_1.

Maximum step size and sidestepping velocity

Previous section described how to compute the gait parameters for a certain side stepping velocity, given the maximum step size. This section proceeds further by estimating of the maximum step size and velocity, that effectively determine the performance boundary on the side stepping gait. To simplify the analysis, the stepping velocity is defined as:

$$v_g = \frac{L_S}{T_{step}} \tag{4.33}$$

Assuming L_S is independent of T_g, equation (4.33) suggests that the velocity is maximized when the stepping distance, L_S is maximized and T_g is minimized. Length L_S is geometrically bounded by the lengths of the legs, L_{leg}, width of the pelvis, L_{plevis}, the length spanned by the inside of the feet L_{feet_inside}, and finally the maximum reach of the ankle joint, θ_{max}, as illustrated in figure 4.8.

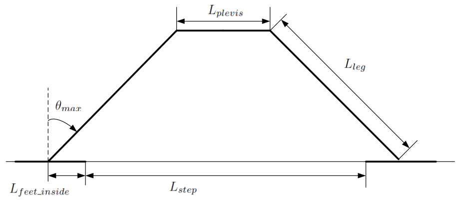


Figure 4.8: Geometric parameters that bound the maximum step size. Parameters L_{foot_inside} and L_S represent the length stretching from the ankle to the inside of the foot and the step size respectively. Parameters L_{plevis} and L_{leg} represent the width of the pelvis and the length of the stretched out leg. Parameter θ_{max} is the maximum rotation available in the ankle X joints.

Using straightforward trigonometry we can determine:

$$\begin{aligned} L_{S_{max}} &= 2L_{leg}\sin(\theta_{max}) + L_{plevis} - 2L_{foot\_inside}) \\ &= 2 * 0.65\sin(25) + 0.155 - 2 * 0.065 \\ &= 0.45\ [m] \end{aligned} \tag{4.34}$$

Note that $L_S = L_{S_{max}}$, which implies $L_0 = L_{0_{min}} = 2L_{foot\_inside}$. The configuration indicated in figure 4.8 also fixes the height of the center of mass, $zc = 0.59$, computed using forward kinematics. To minimize T_g we express it first as a function of α_i and T_c:

$$\begin{aligned} \min(T_g) &= \min(T_1 + T_2) \\ &= \min(2T_c \ln(\frac{1+\sqrt{1-\alpha_1^2}}{\alpha_1}) + 2T_c \ln(\frac{1+\sqrt{1-\alpha_2^2}}{\alpha_2}) \end{aligned} \tag{4.35}$$

Notice that as α_2 is a fraction which we can use in equation (4.29) to bound α_1:

$$\alpha_1 < \frac{S_2}{S_1} \tag{4.36}$$

Using (4.15) and (4.28), we get

$$\alpha_1 < \frac{L_0}{L_0 + L_S} \tag{4.37}$$

Figure 4.9 shows the respective swing phase times when varying α_1 from 0.001 to $\frac{L_0}{L_0+L_S}$. The figure shows that phase time is the most sensitive when α approaches zero. This is expected as $\alpha = 0$ is a statically stable state and thus $T(\alpha = 0) = \inf$.

Applying the upper bound, $\alpha_2 = 1 \Rightarrow \alpha_1 = \frac{L_0}{L_0+L_S}$, implies that second swing phase switches instantly back to the first, thus $T_2 = 0$. Such a gait is illustrated in figure 4.10

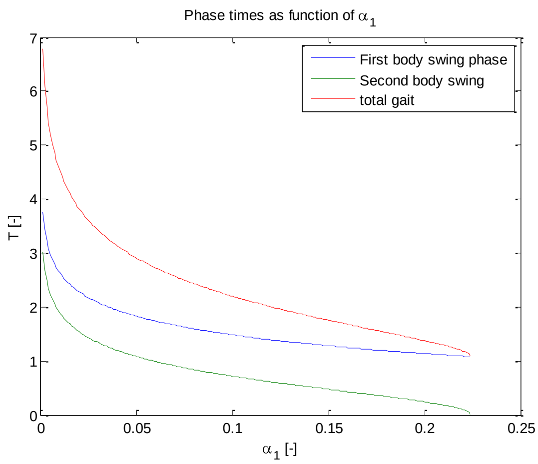


Figure 4.9: Sensitivity of the swing phase times and overall gait cycle time as function of α_1, given a maximum step size, $L_S = 0.45$.

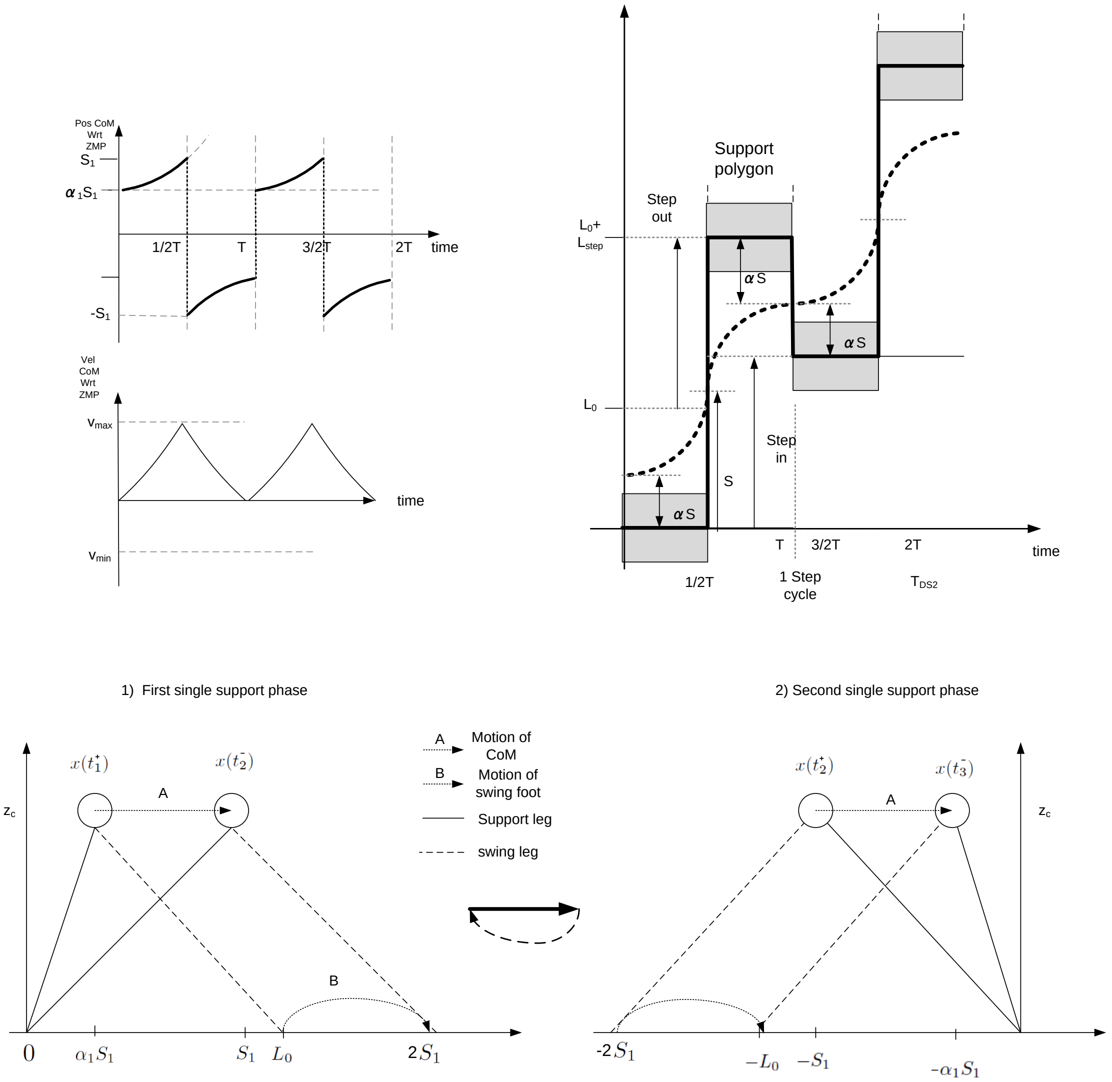


Figure 4.10: CM trajectories for max *alpha* ($\alpha_1 = \frac{L_0}{L_0+L_S}, \alpha_2 = 1$). Top left: CM with respect to the ZMP. Top right: global CM / ZMP trajectory.
Bottom: Pendulum model illustrating switch of the support.

For a given L_0 and L_S, we can now compute the minimum gait time:

$$\begin{aligned} T_g|_{\max\alpha} = T_1|_{max\alpha_1} &= 2T_c \ln(\frac{1+\sqrt{1-\frac{L_0}{L_0+L_S}}^2}{\frac{L_0}{L_0+L_S}}) \\ &= T_c \ln(1+\frac{L_S+\sqrt{L_S^2+2L_0L_S}}{L_0}) \end{aligned} \quad (4.38)$$

The equation can be simplified by expressing L_S as $L_S = K_sL_0$:

$$\begin{aligned} T_g|_{max\alpha} &= 2T_c \ln(\frac{L_0+K_sL_0+\sqrt{K_s^2L_0^2+2K_sL_0^2}}{L_0}) \\ &= 2T_c \ln(1+K_s+\sqrt{K_s^2+2K_s}) \end{aligned} \quad (4.39)$$

Now we can also express the maximum gait velocity (given L_0 and L_S), as a function of K_s:

$$v_g|_{max\alpha} = \frac{L_0K_s}{2T_c \ln(1+K_s+\sqrt{K_s^2+2K_s})} \quad (4.40)$$

The minimum phase time / maximum gait velocity as a function of the step size, (see equations (4.39) and (4.40) are plotted in figure 4.11. The figure shows that the gate velocity is almost a linear function of the step size for $K_s > 1$ which can also be seen from the phase time attending towards a constant value. We can now see that theoretically the maximum side stepping velocity equals $v_{\max} = 0.42\text{ms}^{-1}$, for $L_0 = L_{0_{min}}$, $L_S = L_{S_{max}}$, $\alpha_1 = \frac{L_0}{L_0+L_S}$).

Correctness and feasibility of the result was checked and verified by extrapolating the best fit line of measured data for human normal walking speed, versus step size of a normal person in [5].

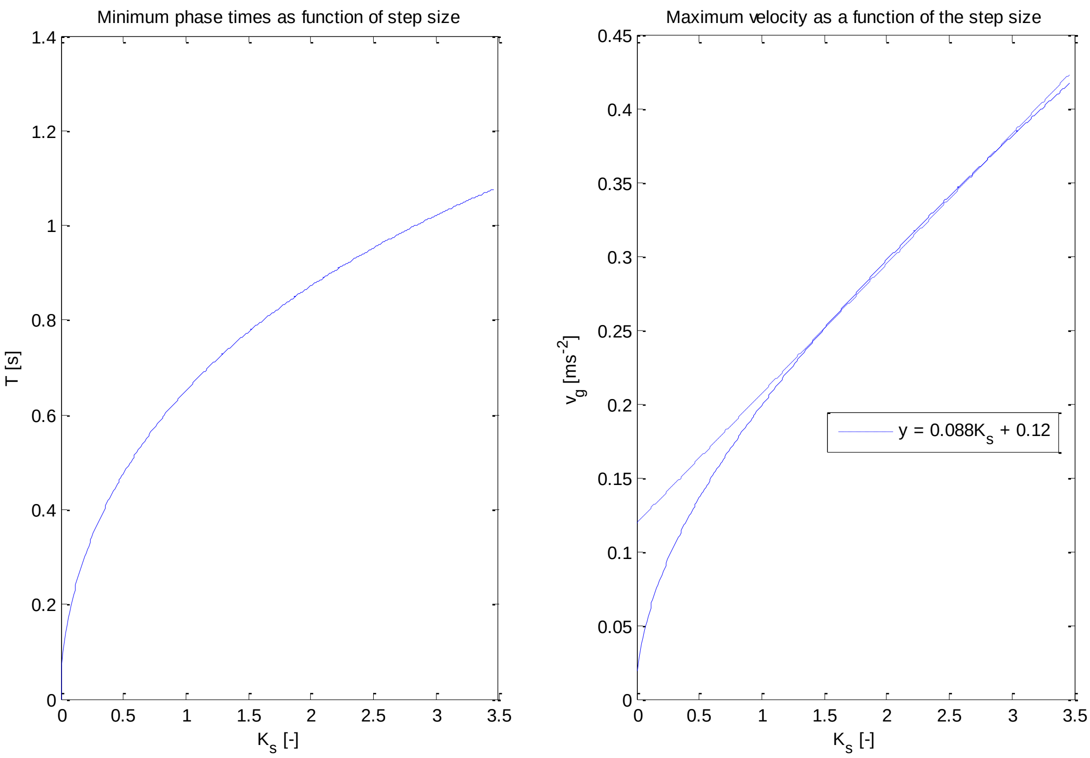


Figure 4.11: Plots showing the minimum gait cycle time (left) and maximum gait velocity (right) (according to gait characteristic with $\alpha_2 = 1$) as function of the step size, $L_S = K_s L_0$. Here $L_0 = L_{0_{min}} = 0.13$ is the initial leg separation distance.

By analysis performed so far, the maximum stepping speed is determined that guarantee balance based on the cart-table model of the ZMP. However, the maximum stepping speed does also depend on the actuation capabilities of the biped. The maximum joint torque which can be applied by the biped actuates are limited by motor type and the available power from the biped's power supply. This will limit the achievable joint and subsequent CM acceleration therefore could lead to instability. Another consideration is that there will always be modelling error between the modelled inertial properties of the biped and the true inertial properties of the biped. As the side step speed increases, the accelerations will increase and therefore also the significance of this modelling error will increase.

4.4 Computation of joint trajectories

4.4.1 Introduction

Based on the knowledge developed in the previous section, we are able to derive a number of trajectories in the task space of the robot,namely:

- The CM moves at a certain height, in order to simplify the ZMP equations, constraining Z-DOF.
- The CM move in the X -Y plain in such a way that the ZMP remains at the ankle joint, constraining X-, and Y-DOFs.

- The position and orientation of the feet are known at the start / end of each motion phase. Interpolating between these constrains 6 DOF.

There is no unique mapping from the task space to the joint space motion variables due to their nonlinear relationship. In this section, the mapping problem is solved using the differential kinematics relationship between the task space and the joint space robot variables, as described in [52].

Inverse differential kinematics

Derivation of the inverse differential kinematic solution follows the following main steps:

Applying forward kinematics to express 12 by 1 column vector x_c containing the task space motion variables of the robot and expressing it a a function of the 12 by 1 column vector q containing the robot joint space variables:

$$x_c = f(q) \tag{4.41}$$

where f is a 12 by 1 column vector function. Differentiating equation (4.41) with respect to time gives:

$$\dot{x}_c = \frac{\partial x_c}{\partial q}\dot{q} = J(q)\dot{q} \tag{4.42}$$

where $J(q) = \frac{\partial x_c}{\partial q}$ is known as the analytical Jacobian. By inverting the Jacobian, we can compute the joint velocities:

$$\dot{q} = J(q)^{-1}\dot{x}_c \tag{4.43}$$

Finally, given an initial robot configuration, $q(0)$, the joint motion trajectories can be determined by time-integration of equation (4.43):

$$q(t) = \int_0^t \dot{q(\zeta)} d\zeta + q(0) \tag{4.44}$$

In practice, due to digital implementation of the robot control software, numerical integration is carried out instead of the continuous one as given by (4.44). Euler integration method is the simplest form of the numerical integration:

$$\begin{aligned} q(t_{k+1}) &= \dot{q}(t_k)\Delta t + q(t_k) + & (4.45)\\ &= J(t_k)^{-1}\dot{x}(t_k)_c \Delta t + q(t_k) & (4.46) \end{aligned}$$

Euler integration methods may suffer from numerical drift and therefore a feedback compensation for this drift.

Notice that that the considered biped contains 12 actuated joints, which is more 9 considered DOFs in the robot task space trajectories, as described by equation (4.4.1). This gives us a convenience to specify 3 more DOFs in the robot task space, in particular, we consider to specify the desired orientation of the robot torso. The reason for this specific choice is that the cart-table model of the ZMP derived in Appendix A neglects rotational moments. The orientation of the body, having the largest mass moment of inertia, should therefore be kept constant along the sidestepping trajectory to prevent generation of the moments that can cause differences between the actual ZMP location and ZMP calculated using the car-table model. The forward kinematic expressions (4.41) are derived in the next section.

4.4.2 Forward kinematics expressions

As discussed in section (4.2.2) and Appendix A, the task space reference robot trajectory x_c, consists of the desired CM trajectory, the position and orientation trajectories of the swing foot, and the orientation trajectory of the torso. This section explains how the so-called forward kinematic expressions can be derived, expressing x_c in terms of the joint space coordinates q as in equation (4.41).

The inertial coordinate frame relative to which x_c is expressed switches back and forth between the feet. For a given base, the first steps for deriving the forward kinematics expression involve assigning coordinate frames to the bodies:

- label the links L_0 to L_{12}, from the robot ground (base link) to the swing foot
- label the joints (from the base one nearest to the ground outwards): J_1 to J_{12}. Also label equivalent points on the sole of the base and swing feet as J_0 and J_{13} respectively.
- Add body fixed coordinate frames $\{\Psi_i\}$ to L_i at position of J_i for i = 1:12. Also add $\{\Psi_0\}$ to L_0
- label the joint angular displacements $q_{BS} = \begin{bmatrix} q_1 & q_2 & ... & q_{12} \end{bmatrix}^T$ Note that subscript "BS" stands for "Base to Swing feet".
- for each link, L_i, we express the position of it's CM and the origin of joint, J_{i+1}, in coordinates frame: $r^i_{CM_i}$, $r^i_{J_{i+1}}$ of the joint J_i.

Each time the base and swing points switch, parameters and variables used to describe the new situation relate with the previous situation as follows:

- The new vector $\tilde{q}$ of the joint coordinates is related to the previous one, q as $\tilde{q}_{BS} = \begin{bmatrix} -q_{12} & -q_{11} & ... & -q_1 \end{bmatrix}^T$
- New body fixed vectors become: $\tilde{r}^i_{J_{i+1}} = -r^i_{J_{i+1}}$ and $\tilde{r}^i_{CM_i} = r^i_{CM_i} - r^i_{J_{i+1}}$,

The position and orientation of each link i, can now be described using a homogenous transformation matrix:

$$A_i^{i-1} = \begin{bmatrix} R(q_i)_i^{i-1} & o_i^{i-1} \\ \underline{0} & 1 \end{bmatrix} \tag{4.47}$$

where o_i^{i-1} denotes position of the origin of the body fixed frame $\{\Psi_i\}$ with respect to $\{\Psi_{i-1}\}$, given by vector $r^i_{J_{i+1}}$. $R(q_i)_i^{i-1}$ is the 3 by 3 rotation matrix discribing the orientation of body fixed frame $\{\Psi_i\}$ with respect to $\{\Psi_{i-1}\}$ is the Concatenating the homogeneous transformation matrixes, we can express the position and orientation of any coordinate frame,$\{\Psi_n\}$ relative to the inertial one $\{\Psi_0\}$:

$$A_n^0(q_1, q_2..q_n) = A_1^0(q_1)A_2^1(q_2)...A_n^{n-1}(q_n) \tag{4.48}$$

It then follows that the homogeneous position of each body fixed vector $P^i = \begin{bmatrix} p_x^i & p_y^i & p_z^i & 1 \end{bmatrix}^T$, can directly be expressed relative to the inertial frame as:

$$P^0 = A_n^0 P^i \tag{4.49}$$

Relation (4.49) is then be used to express the position, of the CM and the swing foot relative to the inertial frame as a function of q:

$$x_p = \begin{bmatrix} p_{CM}(q) \\ p_{swing}(q) \end{bmatrix} = \begin{bmatrix} Q(\sum_{i=0}^{12} A_i^0(q_1..q_i)P_{CM_i}^i m_i)/M \\ QA_{12}^0(q)P_{J_{13}}^{12} \end{bmatrix} \tag{4.50}$$

where $M = \sum_{i=0}^{12} m_i$ and Q is a selector matrix, selecting the first 3 elements from the homogeneous position vector:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.51}$$

Next, the orientation of the torso and the swing foot are found by extracting the rotation matrix R from the homogenous transformation matrix A_i^0 for i = 6 and 12 respectively. Since it is not intuitive to interpret the orientation directly from a rotation matrix, roll pitch yaw angles, $\phi_{torso} = \begin{bmatrix} \alpha_t & \beta_t & \gamma_t \end{bmatrix}^T$ and $\phi_{swing} = \begin{bmatrix} \alpha_s & \beta_s & \gamma_t \end{bmatrix}^T$, are determined from the rotation matrix, as described in [52].

We now have expressions for all reference task space variables x_o, as

$$x_o = \begin{bmatrix} p_{CM}(q) \\ p_{swing}(q) \\ \phi_{torso}(q) \\ \phi_{swing}(q) \end{bmatrix} = \begin{bmatrix} x_p \\ x_\phi \end{bmatrix} \tag{4.52}$$

Where $x_p = \begin{bmatrix} p_{CM}(q) & p_{swing}(q) \end{bmatrix}^T$ is a 6 by 1vector of the translational reference motions of the CM and swing foot, and $x_\phi = \begin{bmatrix} \phi_{torso}(q) & \phi_{swing}(q) \end{bmatrix}^T$ is a 6 by 1 vector of the reference angular motions, expressed in terms of roll, pitch and yaw angles, of the torso and swing feet.

4.4.3 Inverse kinematics algorithm

To compute the analytical Jacobian, we need to find the time derivative for of the kinematic relationship (4.41). The time derivative of the translational variables, x_p, is easily found by taking the partial derivative wrt q:

$$\dot{x}_p = \frac{\partial x_p}{\partial q}\dot{q} = J_p(q)\dot{q} \tag{4.53}$$

In the similar way we determine the time derivative for the orientation variable $\dot{x}_\phi$:

$$\dot{x}_\phi = \frac{\partial x_\phi}{\partial q}\dot{q} = J_\phi(q)\dot{q} \tag{4.54}$$

Putting equations (4.54) and (4.53) together we get:

$$\dot{x}_o = \begin{bmatrix} \dot{x}_p \\ \dot{x}_\phi \end{bmatrix} = \begin{bmatrix} J_p(q) \\ J_\phi(q) \end{bmatrix} \dot{q} = J_A(q)\dot{q} \tag{4.55}$$

where $J_A = \begin{bmatrix} J_p(q) & J_\phi(q) \end{bmatrix}^T$, is known as the analytical Jacobian [48]. It is used in equation (4.46) to compute $q(t_{k+1})$:

$$q(t_{k+1}) = q(t_k) + J_A^{-1}(q(t_k))\dot{x}_o(t_k)\Delta t \tag{4.56}$$

Computed joint velocity $\dot{q}_c = J_A^{-1}(q(t_k))\dot{x}_o(t_k)$ may not coincide with the true value $\dot{q}$ which satisfies equation (4.46) due to discrete-time Euler computation and numerical drift that may therefore arise in the reconstruction of q. To compensate for the drift we consider a task space error between the desired and the actual task space variables:

$$e = x_d - x_o \tag{4.57}$$

Taking the time derivative of (4.57) we get:

$$\begin{aligned} \dot{e} &= \dot{x}_d - \dot{x}_o \\ &= \dot{x}_d - J_A(q)\dot{q} \end{aligned} \tag{4.58}$$

Using a positive definite gain matrix K in the error feedback the error,e, we get

$$\dot{q} = J_A^{-1}(\dot{x}_d + Ke) \tag{4.59}$$

$$\Rightarrow (\dot{x}_d - J_A\dot{q}) + Ke = 0 \tag{4.60}$$

Hence, thanks to the feedback error mechanism applied in (4.58), we achieve asymptotically stable linear error dynamics:

$$\dot{e} + Ke = 0 \tag{4.61}$$

Using equation (4.59) and (4.56) the kinematic control algorithm was created, illustrated in figure 4.12.

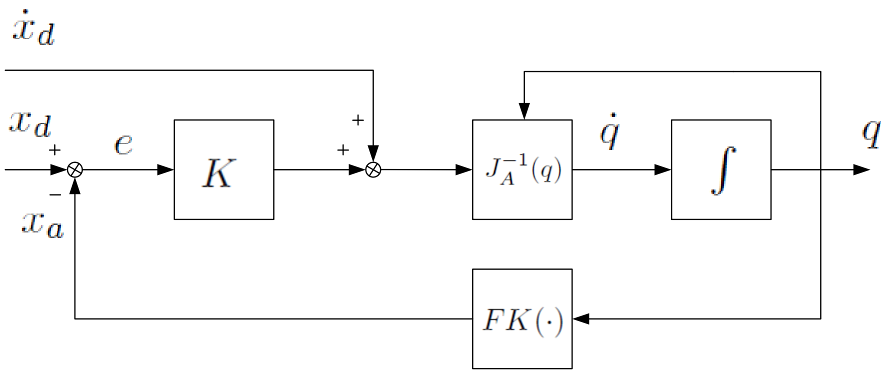


Figure 4.12: Inverse kinematics algorithm. x_d is the desired vector of operational space reference trajectories: position of CM and swing foot, and the orientation of the torso and swing foot (expressed in roll, pitch and yaw angles). $FK(\cdot)$ are forward kinematic expressions which compute the actual operational space reference trajectories, x_a, given the joint angles q. J_A is the analytical jacobian, $J_A = \frac{\partial x_a}{\partial q}$

4.5 Evaluation of side stepping trajectory

4.5.1 Introduction

The inverse kinematics algorithm is examined by finding the joint trajectories for the following arbitrarily chosen side stepping gait parameters:

Symbol	Description	Value
T_{gait}	Total stepping time	8 [s]
D	Total stepping distance	0.2[m]
Z_{CM}	CM height	0.65[m]
T_0	Start and stop time	1[s]
L_{step}	Step size	0.1 [m]

Table 4.1: Side stepping gait configuration parameters

Furthermore, the inverse kinematics algorithm is tested in a simulation of the multi-body dynamics model of the walking robot in Matlab/Simulink/SimMechanics.

4.5.2 Inverse kinematic results

For the gait configurations described in table 4.1, the inverse kinematics algorithm computes the joint trajectories shown in figures 4.13, 4.14, 4.15 and 4.16. These trajectories are then used to compute trajectories of the CM and ZMP. As for ZMP, two models are considered: a simplified cart-table one and the one based on the full inertial properties of the biped consisting of 12 leg joints and the torso. The resulting CM and ZMP trajectories are shown in figure 4.17.

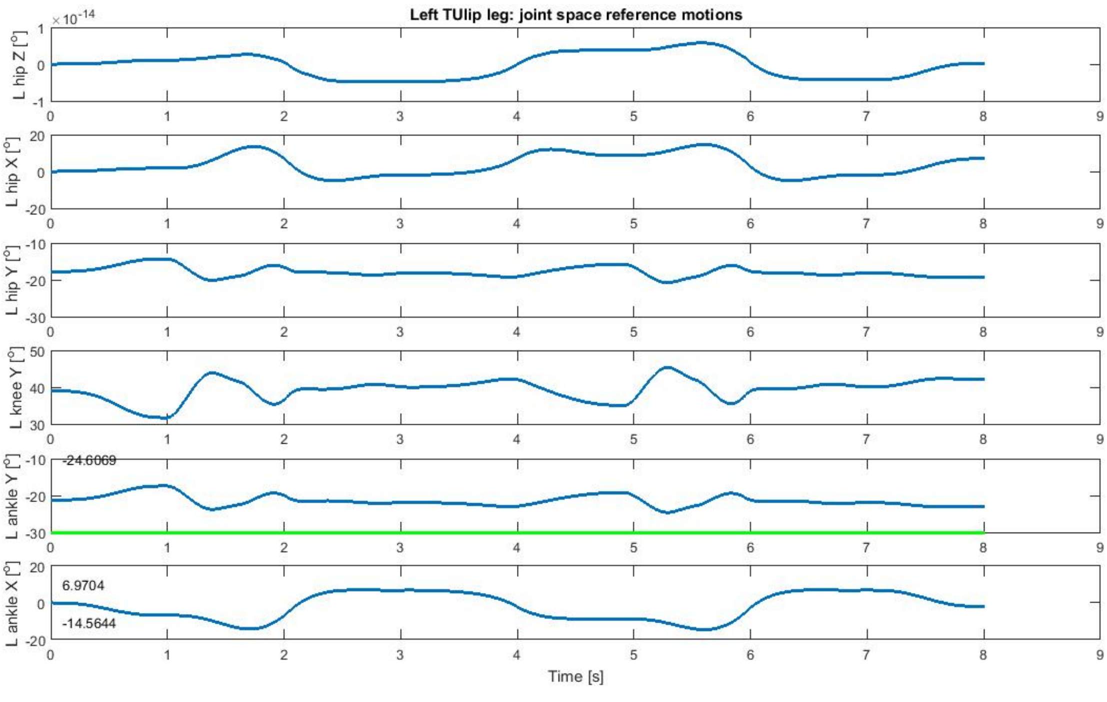


Figure 4.13: Joint motions in the left leg for a 2 step gait.

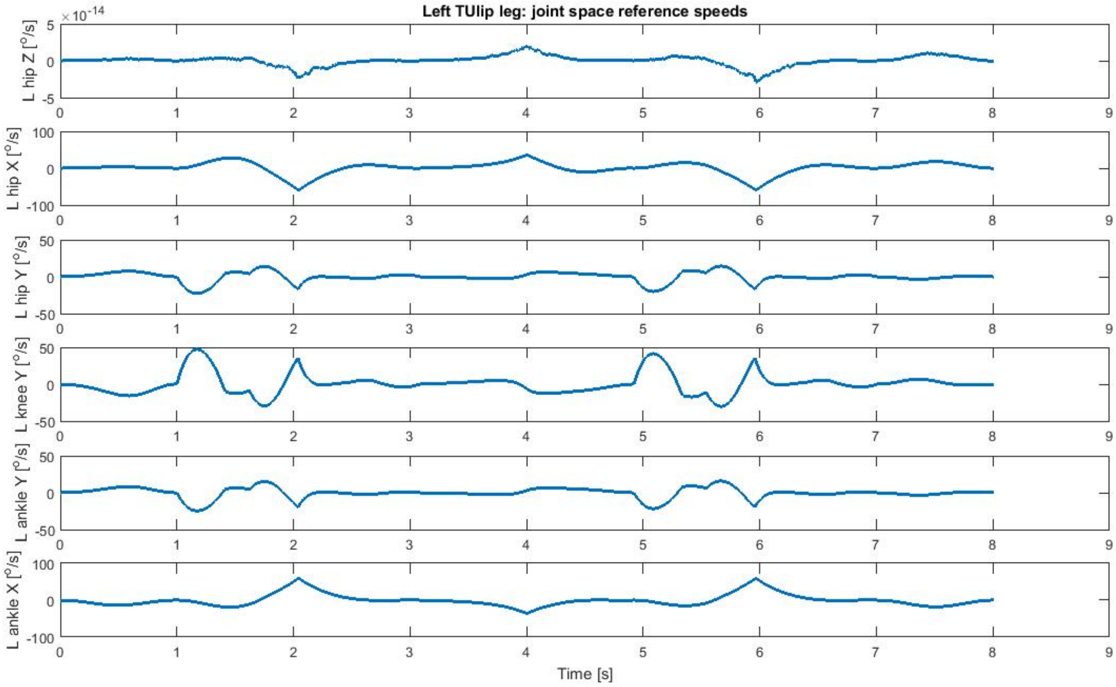


Figure 4.14: Joint speeds in the left leg for a 2 step gait.

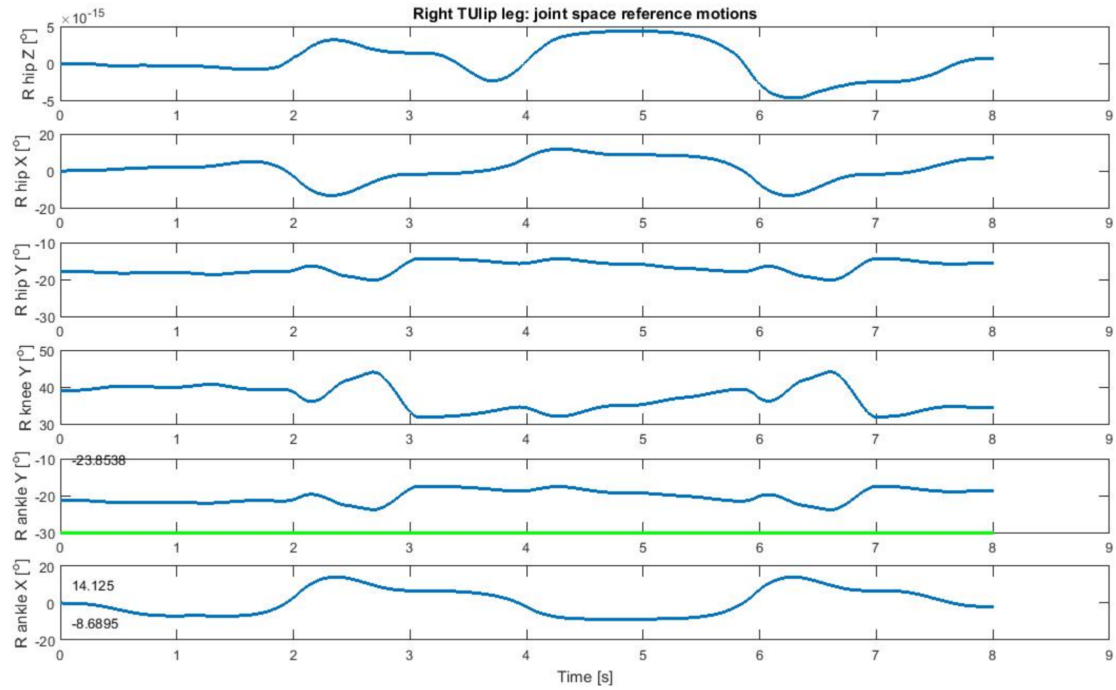


Figure 4.15: Joint motions in the right leg for a 2 step gait.

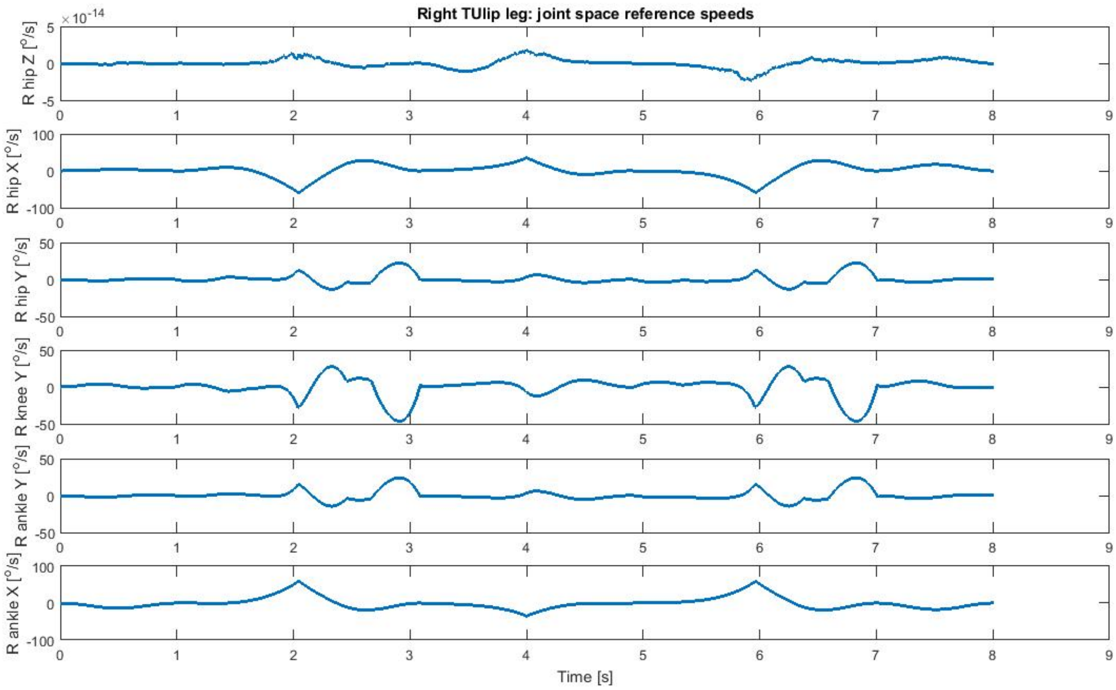


Figure 4.16: Joint speeds in the right leg for a 2 step gait.

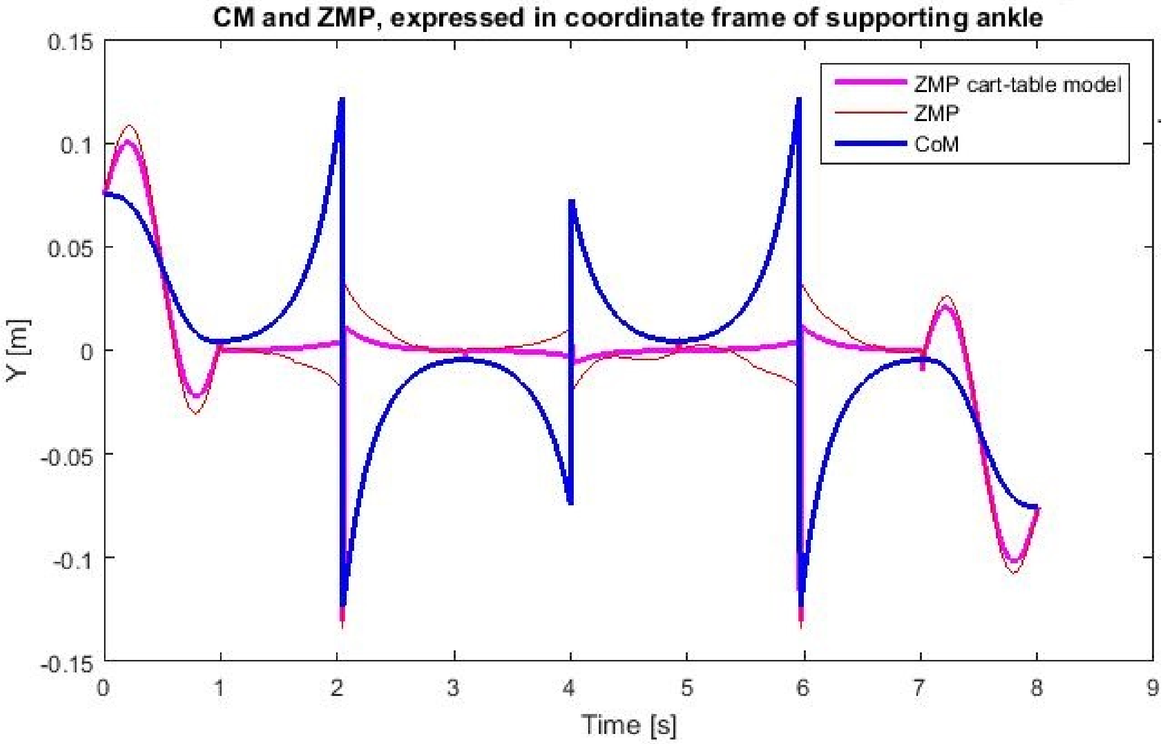


Figure 4.17: CM and ZMP trajectories for a 2 step gait.

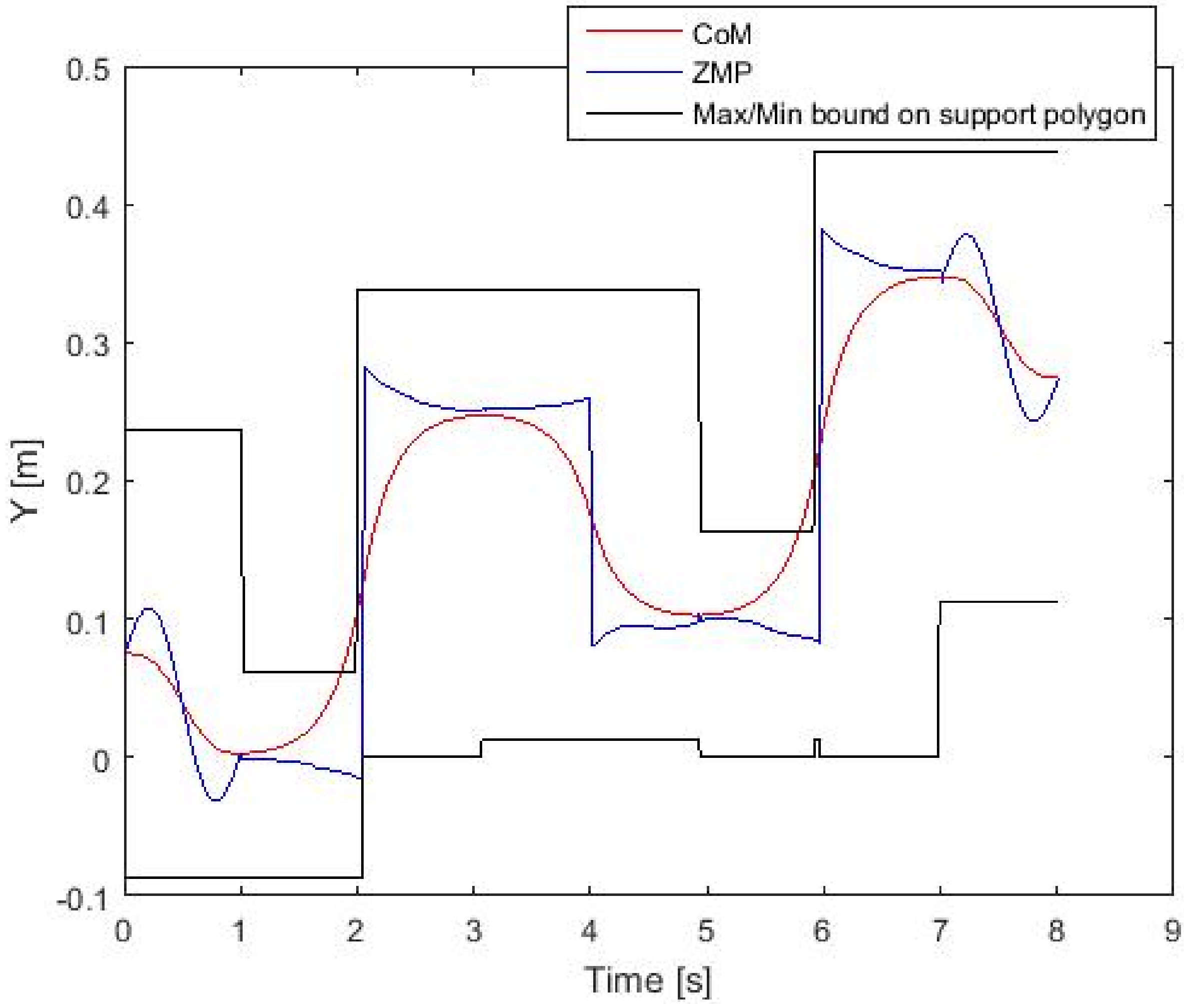


Figure 4.18: CM and ZMP trajectories and limit values of the support polygon expressed in world coordinates, for a 2 step gait. The figure shows that although the CM moves outside the support polygon, the ZMP remains inside at all time

The joint positions and speeds are continuous functions of time, which is necessary to consider the joint trajectories as feasible for the robot. Figure 4.17 shows that the CM accelerates and decelerates as was expected. The figure also show that the first and last second the CM and ZMP make motions with a sin - cosine velocity profiles,which is also demanded for the starting and the stopping phase of the side stepping gait. It shows that in-between these phases, the ZMP remains close to the origin (physically at the supporting ankle), which is as required and expected. By comparing the true ZMP with its simplified cart-table model version, it can be seen that they are both very close to each other. This suggests that it was possible to neglect the rotational moments form the ZMP equations.

4.5.3 Dynamic simulation result

The 2 step gait trajectories computed using the inverse differential kinematic algorithm are then loaded into the Matlanb/Simulink/SimMechanics dynamical model of the biped, which is illustrated in figure 4.19. In the dynamic simulation of the multi-body dynamics of the biped including contact constraints with the ground, it is verified that the biped is able to perform the side steps without falling. The motor toques achieved in this simulation can be seen in figure 4.20. They are an order on magnitude lower than the joint torques which were measured on the biped, which indicates that the trajectory is feasible. The large difference is an indication that there might be a lot of joint friction which is not included in the model. Another explanation is that the measurement were taken while executing forward walking motion, a completely different trajectory. Large torque spikes can be seen at the start of the simulation which are attributed to a slight miss alignment with respect to the required steady state start position of the stiff spring damper contact model ground contact model.

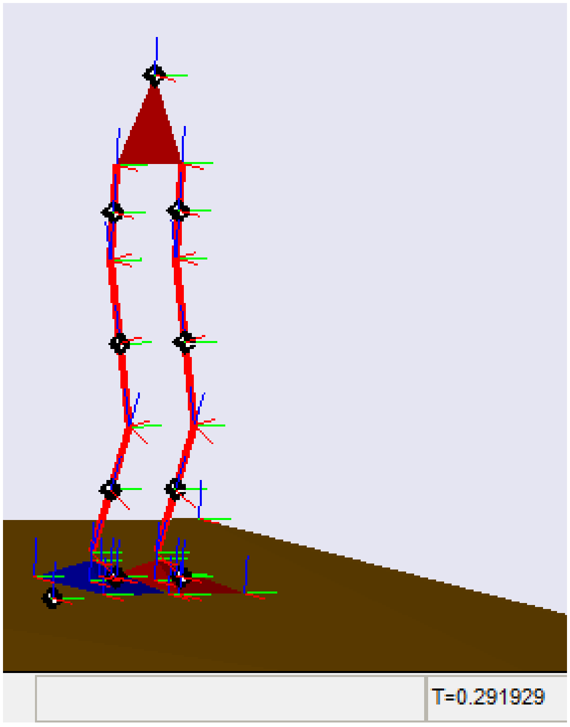


Figure 4.19: Screen shot of the animation generated by the SimMechanics model of the biped.

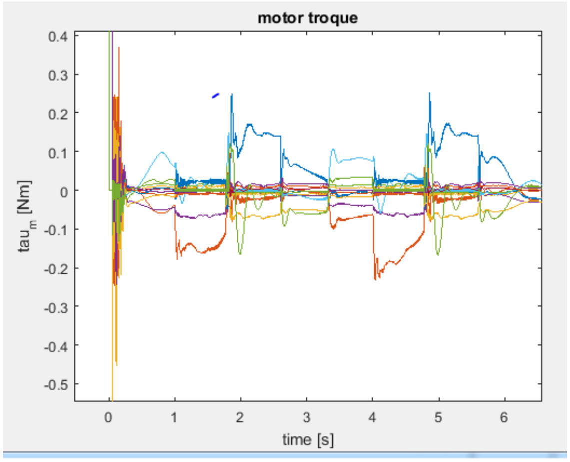


Figure 4.20: Motor torques generated by the SimMechanics dynamical simulation of the biped preforming a 2 step gait. The figure shows that the torques are bounded within a reasonable rage and therefore feasible.

4.5.4 Conclusion

This chapter described an the requirements for the gait requirements for an optimal sidestepping gait, namely that the ZMP remains under the a supporting foot at all times. Analyses of sidestepping speed concluded that the speed was proportional to stepping length and for Tulp's dimensions the speed is theoretically limited 0.42m/s. An algorithm for computing the trajectories was presented the trajectories for a 2 step sidestep were evaluated. Evaluation showed that the ZMP remained inside the supporting polygon indication a stable locomotion. This was result was supported by running a dynamical simulation using the model described in chapter 2, showing that the biped model did not fall over while executing the motion. The order of magnitude of the joint torques were less than these found by measuring the torques in Tulip, which could indicate that the trajectories are feasible. This also might indicate that there is something missing from the model, for example dominant joint friction.

Chapter 5

Conclusions and recommendations

5.1 Conclusions

In this thesis, the bipedal locomotion is studied for the purpose of preforming stable side stepping motion. The biped TUlip is considered as an experimental test-bed which is required to perform stable side steps in a timely manner. From the literature study, it is found that despite a lot of research on the bipeds, the main focus of the studies so far is on straightforward walking in the robot sagittal plain. The side stepping in the coronal robot plain of motion is rarely addressed. Furthermore, following the conclusions of Van Zutven et al. in [75] about limitations of 2D biped modeling, it is decided to consider 3D modeling of the biped dynamics and design of the 3D side stepping gait.

After carefully considering a number of simulation platforms, a 3D dynamical model is created using Matlab. During the modelling process, it is found that modeling of the motor dynamics can be avoided due to the high servo control bandwidth of the electrical motors with the corresponding power amplifiers. An important aspect of the humanoid model is the ground contact model, which is also considered in detail. Constraint and penalizing methods for contact models are considered of which the penalizing method is chosen since it guarantees an unique solution for the foot contact force as a function of time. The contact parameters, such as the contact stiness, are tuned such as to achieve realistic contact forces within a reasonable simulation time. Finally, quality of the model is evaluated by comparing the simulated joint torques and velocities with the corresponding signals measured in experiments on TUlip, showing that the computed and measured torques are of the same order of magnitude. The values, however, were different. This is possibly due to modelled joint friction.

General biped locomotion stability is studied together with sidestepping step phases. It is found that the dynamic balance could be maintained by ensuring that the zero moment point, ZMP, remains inside the support polygon. For robustness of the balance, it is important to monitor how far this ZMP is from the outer edges of the support polygon. To maximize the robustness of the balance during the side stepping, the ZMP is required to remain at the position of the ankle of the stand foot. This gait requirement has an additional advantage in that the ankle joint moves passively, preserving electrical energy. Furthermore, it is found that the sidestepping speed could be regulated by controlling the initial position of the center of mass at the start of the sidestepping gait.

A framework for stable and timely sidesteps is presented. It consists of state-machines that control the biped motion between 5 general posture states, in order to perform the sidestepping walk. Depending on the desired stepping distance and available time, the gait parameters are computed. These parameters are used to compute the joint trajectories required to achieve the 5 general postures by executing the side stepping gait.

Analyses of the sidestepping velocity, according to the proposed motion pattern, shows that the velocity would increase linearly with a step size. This means that the maximum side-stepping velocity is therefore bounded by the maximum side step length of the biped. In that case, the maximum sidestepping speed for TUlip is estimated to 0.42[m/s]. This conclusion is valuable, as it can be used to quickly evaluate if it is feasible to reach a certain target position within a target time.

To evaluate the algorithm for calculation of the sidestepping gait, the joint trajectories are computed for a 2 side step gait. The resulting joint rotations and velocities are found to be continuous in time. The achieved ZMP motion is simulated using a 3D dynamical model of Tulip. In the simulations, the ZMP remains inside the supporting polygon, which is a sufficient condition for the biped to remain dynamically balanced.

5.2 Recommendations

In order to support the assumption that the large difference found between the experimentally determined and simulated joint torques are related to friction, it is recommended to measure the friction in the joints and add joint friction to the dynamical model to see if this in fact explains the difference

As the side stepping algorithm is only tested in simulation, experimental validation of these results is highly recommended. To make the gait generation algorithm more robust to modelling uncertainties, it is advised enhance it by an ZMP feedback, such as in the framework presented by Napoleon et al in [34].

The sidestepping algorithm is evaluated for one particular side stepping gait speed only. It would be interesting to investigate what would happen with the bipeds dynamic balance when the required sidestepping speed increase up to and beyond the theoretical maximum. While increasing the sidestepping speed, the biped's higher order dynamics are more excited and thus bigger deviations to the simplified balance model are expected. This investigation would give an indication of how robust the sidestepping motion pattern is with respect to the increases stepping velocity. In addition it is recommended to investigate the affect of the computed joint torques. This is an indication of whether or not the computed side setting motions are feasible.

Bibliography

[1] Zero-moment point method for stable biped walking. 2009.

[2] DARPA Robotics Challenge. April 2012.

[3] Michel Abadie. Dynamic simulation of rigid bodies: Modelling of frictional contact. In Bernard Brogliato, editor, *Impacts in Mechanical Systems*, volume 551 of *Lecture Notes in Physics*, pages 61–144. Springer Berlin / Heidelberg, 2000.

[4] M. Alitavoli, M. Taherkhorsandi, M. J. Mahmoodabadi, A. Bagheri, and B. Miripour-Fard. Pareto design of sliding-mode tracking control of a biped robot with aid of an innovative particle swarm optimization. In *Innovations in Intelligent Systems and Applications (INISTA), 2012 International Symposium on*, pages 1–5, July.

[5] T P Andriacchi, J A Ogle, and J O Galante. Walking speed as a basis for normal and abnormal gait measurements. *Journal of Biomechanics*, pages 261–268, 1977.

[6] Kyu-Cheon Choi, Hyun-Jeong Lee, and Min Cheol Lee. Fuzzy posture control for biped walking robot based on force sensor for zmp. In *SICE-ICASE, 2006. International Joint Conference*, pages 1185 –1189, 2006.

[7] T De Boer. *Foot placement in robotic bipedal locomotion.* PhD thesis, Delft University of Technology, 2012.

[8] Eric G Kubica Derek L Wight and David W L Wang. Introduction of the foot placement estimator: A dynamicmeasure of balance for bipedal robotics. *Journal of Computational and Nonlinear Dynamics*, 2008.

[9] Kostic Dragan. Humanoid robots. University Lecture, 2011.

[10] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972.

[11] A. Ehsaniseresht and M. M. Moghaddam. A new ground contact model for the simulation of bipeds' walking, running and jumping. In *Robotics and Mechatronics (ICROM), 2015 3rd RSI International Conference on*, pages 535–538, Oct 2015.

[12] Roy Featherstone. An empirical study of the joint space inertia matrix. *In International Journal of Robotics Research*, 23:859–871, 2004.

[13] Li Fu, Qi Wang, and Shimin Wang. Time-stepping for multibody dynamics with friction-affected bilateral constraints. *Progress in Natural Science*, 19(12):1799 – 1804, 2009.

[14] J.Van Der Kamp P.C.C.Ward G.J.P.Savelsbergh, M.A.Williams. Visual search, anticipation and expertise in soccer goalkeepers. *Journal of Sports Sciences*, pages 0264–0414, 2002.

[15] R. E. Goddard, H. Hemami, and F. C. Weimer. Biped side step in the frontal plane. In *Decision and Control including the Symposium on Adaptive Processes, 1981 20th IEEE Conference on*, volume 20, pages 147 –155, 1981.

[16] Taesin Ha and Chong-Ho Choi. An effective trajectory generation method for bipedal walking. *Robot. Auton. Syst.*, 55:795–810, October 2007.

[17] Hooshang Hemami, Kamran Barin, Laci Jalics, and Deborah Givens Heiss. Dynamics, stability, and control of stepping. *Annals of Biomedical Engineering*, 32:1155–1162, 2004. 10.1114/B:ABME.0000036651.26026.a3.

[18] Arjan Smorenberg Eelko van Breda Guus Liqui Lung Freerk Wilbers Corn Plooij Gijs van der Hoorn Edwin Dertien Gijs van Oort Martijn Wisse Pieter Jonker Stefano Stramigioli Henk Nijmeijer Thom Warmerdam4 hilip Heijkoop, Tomas de Boer. Dutch robotics 2008 teen-size team description. Admission Paper, Robocup 2008.

[19] Satoshi Kagami, Tomonobu Kitagawa, Koichi Nishiwaki, Tomomichi Sugihara, Masayuki Inaba, and Hirochika Inoue. A fast dynamically equilibrated walking trajectory generation method of humanoid robot. *Autonomous Robots*, 12:71–82, 2002. 10.1023/A:1013210909840.

[20] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 2, pages 1620 – 1626 vol.2, 2003.

[21] S Kajita, F Kanehiro, K Kaneko, K Yokoi, and H Hirukawa. The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. *Proceedings 2001 IEEERSJ International Conference on Intelligent Robots and Systems Expanding the Societal Role of Robotics in the the Next Millennium Cat No01CH37180*, 1(4):239–246, 2001.

[22] James Kuffner, Jr., Satoshi Kagami, Masayuki Inaba, and Hirochika Inoue. Dynamically-stable motion planning for humanoid robots, 2000.

[23] James Kuffner, Koichi Nishiwaki, Satoshi Kagami, Masayuki Inaba, and Hirochika Inoue. Motion planning for humanoid robots. In Paolo Dario and Raja Chatila, editors, *Robotics Research*, volume 15 of *Springer Tracts in Advanced Robotics*, pages 365–374. Springer Berlin / Heidelberg, 2005.

[24] Remco I. Leine and Nathan Wouw. *Stability and Convergence of Mechanical Systems with Unilateral Constraints*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[25] R. Lewandowski and B. Choryczewski. Identification of the parameters of the kelvin-voigt and the maxwell fractional models, used to modeling of viscoelastic dampers. *Comput. Struct.*, 88:1–17, January 2010.

[26] D.Juri M. Vukobratovi c. Contribution to the synthesis of biped gait. *IFAC Symp. Technical and Biological Problem on Control, Erevan, USSR*, 1968.

[27] D.W. Marhefka and D.E. Orin. A compliant contact model with nonlinear damping for simulation of robotic systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 29(6):566 –572, November 1999.

[28] Takamitsu Matsubara, Jun Morimoto, Jun Nakanishi, Masa aki Sato, and Kenji Doya. Learning cpg-based biped locomotion with a policy gradient method. *Robotics and Autonomous Systems*, 54(11):911 – 920, 2006. Planning Under Uncertainty in Robotics.

[29] Kiyotoshi Matsuoka. Mechanisms of frequency and pattern control in the neural rhythm generators. *Biological Cybernetics*, 56(5):345–353, 1987.

[30] Tad McGeer. Passive dynamic walking. *Int. J. Rob. Res.*, 9:62–82, March 1990.

[31] Melanie Mitchell and Stephanie Forrest. Genetic algorithms and artificial life. *Artificial Life*, 1(3):267–289, 1994.

[32] Katja D. Mombaur, Richard W. Longman, Hans Georg Bock, and Johannes P. Schloumlder. Open-loop stable running. *Robotica*, 23(01):21–33, 2005.

[33] J. J. Moreau. *Unilateral contact and dry friction in finite freedom dynamics*, pages 1–82. Number 302. 1988.

[34] Napoleon, S. Nakaura, and M. Sampei. Balance control analysis of humanoid robot based on zmp feedback control. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2437–2442 vol.3, 2002.

[35] Terumasa Narukawa, Masaki Takahashi, and Kazuo Yoshida. Efficient walking with optimization for a planar biped walker with a torso by hip actuators and springs. *Robotica*, FirstView:1–8, 2010.

[36] V. Pallavi, Jayanta Mukherjee, Arun K. Majumdar, and Shamik Sural. Ball detection from broadcast soccer videos using static and dynamic features. *J. Vis. Comun. Image Represent.*, 19:426–436, October 2008.

[37] Jong Hyeon Park. Fuzzy-logic zero-moment-point trajectory generation for reduced trunk motions of biped robots. *Fuzzy Sets and Systems*, 134(1):189 – 203, 2003.

[38] Bart Peerdeman. Building a world model for the tulip humanoid soccer robot. Master's thesis, University of Twente, 2009.

[39] Zhaoqin Peng, Qiang Huang, Xiaojun Zhao, Tao Xiao, and Kejie Li. Online trajectory generation based on off-line trajectory for biped humanoid. In *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on*, pages 752 –756, 2004.

[40] V. L. Popov. *Kontaktmechanik Und Reibung: Ein Lehr- Und Anwendungsbuch Von Der Nanotribologie bis zur numerischen Simulation.* Springer Publishing Company, Incorporated, 2009.

[41] Gill A. Pratt, Arthur C. Smith, Jerry E. Pratt, and Jerry E. Pratt. Exploiting inherent robustness and natural dynamics in the control of bipedal walking robots, 2000.

[42] J. Pratt, J. Carff, S. Drakunov, and A. Goswami. Capture point: A step toward humanoid push recovery. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 200–207, Dec.

[43] Jerry Pratt, Peter Dilworth, and Gill Pratt. Virtual model control of a bipedal walking robot. In *IEEE Conference on Robotics and Automation*, pages 193–198, 1997.

[44] H. Nijmeijer P.W.M. van Zutven, D. Kostic. Modelling identification and stability of humanoid robots. Master's thesis, Eindhoven University of Technology, 2009.

[45] Miti Ruchanurucks and Shinichiro Nakaoka. Offline and online trajectory generation with sequential physical constraints. In *Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics*, pages 578–583, Washington, DC, USA, 2009. IEEE Computer Society.

[46] E. Schuitema, M. Wisse, T. Ramakers, and P. Jonker. The design of leo: A 2d bipedal walking robot for online autonomous reinforcement learning. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3238 –3243, 2010.

[47] Keehong Seo, Joohyung Kim, and Kyungshik Roh. Towards natural bipedal walking: Virtual gravity compensation and capture point control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4019–4026, Oct.

[48] B. Siciliano, L. Sciavicco, and L. Villani. *Robotics: modelling, planning and control.* Advanced textbooks in control and signal processing. Springer, 2009.

[49] Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer, Berlin, Heidelberg, 2008.

[50] Ren Souchet. Restitution and friction laws in rigid body collisions. *International Journal of Engineering Science*, 32(5):863 – 876, 1994.

[51] Mark W. Spong and Gagandeep Bhatia. Further results on control of the compass gait biped, 2003.

[52] Mark W. Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot modeling and control*. John Wiley & Sons, Hoboken (N.J.), 2006.

[53] D. Stewart and J.C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with coulomb friction. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 1, pages 162 –169 vol.1, 2000.

[54] C. Studer, R. I. Leine, and Ch. Glocker. Step size adjustment and extrapolation for time-stepping schemes in non-smooth dynamics. *International Journal for Numerical Methods in Engineering*, 76(11):1747–1781, 2008.

[55] Tomomichi Sugihara, Yoshihiko Nakamura, and Hirochika Inoue. Realtime humanoid motion generation through zmp manipulation based on inverted pendulum control. pages 1404–1409, 2002.

[56] Victor Szebehely. Review of concepts of stability. *Celestial Mechanics and Dynamical Astronomy*, 34:49–64, 1984. 10.1007/BF01235791.

[57] A. Takanishi, M. Ishida, Y. Yamazaki, and I. Kato. The realization of dynamic walking by the biped walking robot WL-10RD. *Proc. Intl. Conference on Advanced Robotics*, pages 459–466, 1985.

[58] Lim H. Tsuda M. Takanishi, A. and I. Kato. Realization of dynamic biped walking stabilized by trunk motion on a sagittally uneven surface. pages 323–330. IEEE International Workshop on Intelligent Robots and Systems, 1990.

[59] R. Tedrake, T.W. Zhang, and H.S. Seung. Stochastic policy gradient reinforcement learning on a simple 3d biped. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2849 – 2854 vol.3, 2004.

[60] Russ Tedrake. Actuating a simple 3d passive dynamic walker. In *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA*, pages 4656–4661, 2004.

[61] Hw Van De Crommert, T Mulder, and J Duysens. Neural control of locomotion: sensory control of the central pattern generator and its relation to treadmill training. *Gait and Posture*, 7(3):251–263, 1998.

[62] Pieter van Zutven, Dragan Kostic, and Henk Nijmeijer. Foot placement for planar bipeds with point feet. In *ICRA*, pages 983–988. IEEE, 2012.

[63] Bram Vanderborght. *Dynamic stabilisation of the biped Lucy powered by actuators with controllable stiffness*. PhD thesis, Universitijd Brussel, 2007.

[64] I.B Vapnyarskii. *Lagrange multipliers*. Springer, 2001.

[65] M. Veloso and P. Stone. Individual and collaborative behaviors in a team of homogeneous robotic soccer agents. pages 309 –316, jul. 1998.

[66] Miomir Vukobratovic and Branislav Borovac. Zero-moment point - thirty five years of its life. *I. J. Humanoid Robotics*, 1(1):157–173, 2004.

[67] Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Optimizing walking controllers. *ACM Trans. Graph.*, 28:168:1–168:8, December 2009.

[68] Lukasz Wiklendt, Stephan K Chalup, and Maria M Seron. *Quadratic Leaky Integrate-and-Fire Neural Network Tuned with an Evolution-Strategy for a Simulated 3D Biped Walking Controller*, pages 144–149. 2008.

[69] Lukasz Wiklendt, Stephan K Chalup, and Maria M Seron. Simulated 3d biped walking with an evolution-strategy tuned spiking neural network. *Neural Network World*, 19(2):235–246, 2009.

[70] W.L. Wooten and J.K. Hodgins. Simulating leaping, tumbling, landing and balancing humans. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 1, pages 656 –662 vol.1, 2000.

[71] Nathan Van de Wouw. Multi-body dynamics. University Lecture, 2011.

[72] K. Yamane and Y. Nakamura. Dynamics filter - concept and implementation of online motion generator for human figures. *Robotics and Automation, IEEE Transactions on*, 19(3):421 – 432, 2003.

[73] Xinguo Yu, Changshen Xu, Qi Tian, and Hon Wai Leong. A ball tracking framework for broadcast soccer video. In *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, volume 2, pages II – 273–6 vol.2, 2003.

[74] Lotfi A. Zadeh. Fuzzy logic, neural networks, and soft computing. *Commun. ACM*, 37:77–84, March 1994.

[75] P.W.M. van Zutven. *Control and Identification of Bipedal Humanoid Robots*. PhD thesis, Eindhoven University of Technology, 2014.

Appendix A

Linear Inverted pendulum model

The linear inverted pendulum model, [20], simulates a biped in SS phase. The biped is modeled as having point mass at the end of a massless rod which is attached to the origin resembling the ankle position. The massless rod is actuated with a prismatic joint allowing elongation r and two rotational joints allowing rotations, θ_r and θ_p, about the x an y axis, as shown in figure A.1.

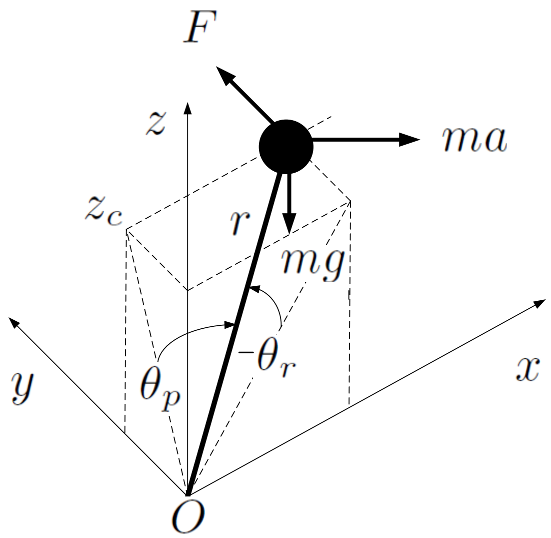


Figure A.1: LIPM

$\tau = \begin{bmatrix}\tau_r & \tau_p & r\end{bmatrix}^T$ is the column are the generalized forces associated with the generalized coordinates $q = \begin{bmatrix}\theta_r & \theta_p & r\end{bmatrix}^T$. The cartesian position of the mass, $p = \begin{bmatrix}x & y & z\end{bmatrix}^T$ can be expressed in terms of q:

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\sin\theta_p \\ -r\sin\theta_r \\ rD \end{bmatrix} \tag{A.1}$$

Where $D = \sqrt{1 - \sin^2\theta_p - \sin^2\theta_r}$. The inertial and gravity forces $m\vec{a}$ and $m\vec{g}$ are acting on the point mass. Applying Newton's formulation of the Equation of motion, we get

$$\vec{F} = m\vec{a} + m\vec{g} \tag{A.2}$$

Where $\vec{F}$ is the reaction force on the mechanism. Taking the partial deferential of p wrt q, $J = \frac{\partial p}{\partial q}$, we can compute the joint torques required to keep the system in equilibrium:

$$\tau = J^T \vec{F} \tag{A.3}$$

Where

$$J = \frac{\partial p}{\partial q} = \begin{bmatrix} \frac{\partial x}{\partial \theta_r} & \frac{\partial x}{\partial \theta_p} & \frac{\partial x}{\partial r} \\ \frac{\partial y}{\partial \theta_r} & \frac{\partial y}{\partial \theta_p} & \frac{\partial y}{\partial r} \\ \frac{\partial z}{\partial \theta_r} & \frac{\partial z}{\partial \theta_p} & \frac{\partial z}{\partial r} \end{bmatrix} = \begin{bmatrix} 0 & r\text{cos}\theta_p & \text{sin}\theta_p \\ -r\text{cos}\theta_r & 0 & -\text{sin}\theta_r \\ \frac{-r\text{sin}\theta_r\text{cos}\theta_r}{D} & \frac{-r\text{sin}\theta_p\text{cos}\theta_p}{D} & D \end{bmatrix} \tag{A.4}$$

Rearranging and inserting A.2, we can write

$$\begin{aligned} \tau &= J^T \vec{F} && \text{(A.5)} \\ &= J^T(m\vec{a} + m\vec{g}) && \text{(A.6)} \\ \Rightarrow mJ^T\vec{a} &= \tau - mJ^T\vec{g} && \text{(A.7)} \\ mJ^T \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} &= \begin{bmatrix} \tau_r \\ \tau_p \\ f \end{bmatrix} - mJ^T \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} && \text{(A.8)} \\ m \begin{bmatrix} 0 & -r\text{cos}\theta_r & \frac{-r\text{sin}\theta_r\text{cos}\theta_r}{D} \\ r\text{cos}\theta_p & 0 & \frac{-r\text{sin}\theta_p\text{cos}\theta_p}{D} \\ \text{sin}\theta_p & -\text{sin}\theta_r & D \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} &= \begin{bmatrix} \tau_r \\ \tau_p \\ f \end{bmatrix} - mg \begin{bmatrix} \frac{-r\text{sin}\theta_r\text{cos}\theta_r}{D} \\ \frac{-r\text{sin}\theta_p\text{cos}\theta_p}{D} \\ D \end{bmatrix} && \text{(A.9)} \end{aligned}$$

Multiplying the first and second rows though by $\frac{D}{\text{cos}\theta_r}$ and $\frac{D}{\text{cos}\theta_p}$ and using the kinematic relations A.1, we get

$$\begin{aligned} m \begin{bmatrix} 0 & -rD & -r\text{sin}\theta_r \\ rD & 0 & -r\text{sin}\theta_p \\ \text{sin}\theta_p & -\text{sin}\theta_r & D \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} &= \begin{bmatrix} \frac{D}{\text{cos}\theta_r}\tau_r \\ \frac{D}{\text{cos}\theta_p}\tau_p \\ f \end{bmatrix} - mg \begin{bmatrix} -r\text{sin}\theta_r \\ -r\text{sin}\theta_p \\ D \end{bmatrix} && \text{(A.10)} \\ m \begin{bmatrix} 0 & -z & y \\ z & 0 & x \\ \text{sin}\theta_p & -\text{sin}\theta_r & D \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} &= \begin{bmatrix} \frac{D}{\text{cos}\theta_r}\tau_r \\ \frac{D}{\text{cos}\theta_p}\tau_p \\ f \end{bmatrix} - mg \begin{bmatrix} y \\ -x \\ D \end{bmatrix} && \text{(A.11)} \end{aligned}$$

The dynamical equations describing the inverted pendulum in the x-y plain are therefore

$$\begin{aligned} my\ddot{z} - mz\ddot{y} &= \frac{D}{\text{cos}\theta_r}\tau_r + mgy && \text{(A.12)} \\ mx\ddot{z} + mz\ddot{x} &= \frac{D}{\text{cos}\theta_p}\tau_p - mgx && \text{(A.13)} \end{aligned}$$

In order to simply equations A.12 and A.13, the CM motion is constrained to the plain with normal vector $n = \begin{bmatrix} k_x & k_y & z_c \end{bmatrix}^T$, where k_x and k_y are chosen equal to zero. Note that the equations have also been shown to simplify in the case where walking on a slope of stairs is requred ($k_x \neq 0$,$k_y \neq 0$) [21]. Also input liberalization is applied, placing input torques τ_r and τ_p virtual input $u_x = \frac{D}{\text{cos}\theta_r}\tau_r$ and $u_y = \frac{D}{\text{cos}\theta_p}\tau_p$, such that

$$\begin{aligned} \ddot{y} &= \frac{g}{z_c}y - \frac{1}{mz_c}u_x && \text{(A.14)} \\ \ddot{x} &= \frac{g}{z_c}y + \frac{1}{mz_c}u_y && \text{(A.15)} \end{aligned}$$

In conclusion, input liberalization and constraining the motion in x-y plain were used to derive equations A.14 and A.14, independent linear equations which describe the motion of the inverted pendulum.

CM - ZMP relation

Considering another point, P, on the support polygon, the contact of the pendulum with the ground will produces a reaction force $\vec{R}$ and moment M_p at point P. The moment, $M_O = \begin{bmatrix}\tau_x & \tau_y \tau_z\end{bmatrix}^T$ produced by the ground reaction force, $\vec{R}$, will be equal to:

$$M_O = M_P + \vec{OP} \times \vec{R} \tag{A.16}$$

If point P is the ZMP of the system, then $\vec{OP} = \begin{bmatrix}x_{zmp} & y_{zmp} & z_{zmp}\end{bmatrix}^T$, $M_p = 0$, therefore

$$M_O = \begin{bmatrix}\tau_x \\ \tau_y \\ \tau_z\end{bmatrix} = \begin{bmatrix}x_{zmp} \\ y_{zmp} \\ z_{zmp}\end{bmatrix} \times \vec{R} \tag{A.17}$$

The reaction force $\vec{R}$ must be equal to

$$\begin{aligned}\vec{R} &= m\vec{a} + m\vec{g} && \text{(A.18)}\\ &= m\begin{bmatrix}\ddot{x} \\ \ddot{y} \\ \ddot{z} + g\end{bmatrix} && \text{(A.19)}\end{aligned}$$

Substituting into A.17 and setting $z_{zmp} = 0$, because the ZMP lies on the ground plane, we get

$$\begin{aligned}\begin{bmatrix}\tau_x \\ \tau_y \\ \tau_z\end{bmatrix} = m\begin{bmatrix}x_{zmp} \\ y_{zmp} \\ 0\end{bmatrix} \times \begin{bmatrix}\ddot{x} \\ \ddot{y} \\ \ddot{z} + g\end{bmatrix} &= m\begin{bmatrix}y_{zmp}(\ddot{z} + g) \\ -x_{zmp}(\ddot{z} + g) \\ x_{zmp}\ddot{y} - y_{zmp}\ddot{x}\end{bmatrix} && \text{(A.20)}\\ \Rightarrow x_{zmp} &= -\frac{\tau_y}{m(\ddot{z} + g)} && \text{(A.21)}\\ y_{zmp} &= \frac{\tau_x}{m(\ddot{z} + g)} && \text{(A.22)}\end{aligned}$$

As the motion of the CM will be constrained to the horizontal plain intersecting the $z = z_c$ (a requirement for simplifying the A.12 and A.13 equations), equations A.23 and A.24 become simply:

$$\begin{aligned}x_{zmp} &= -\frac{\tau_y}{mg} && \text{(A.23)}\\ y_{zmp} &= \frac{\tau_x}{mg} && \text{(A.24)}\end{aligned}$$

It is interesting to note that in the absence of lateral accelerations, ($\ddot{x} = \ddot{y} = 0$), equations A.14 and A.15 reduce to A.23 and A.24, proving that in such a case the ZMP is equal the the floor projection of the CM.

Substituting the A.23 and A.24 into the equations of CM motion, A.14 and A.15 we finally get the ZMP equations:

$$\begin{aligned}x_{zmp} &= x - \frac{z_c}{g}\ddot{x} && \text{(A.25)}\\ y_{zmp} &= y - \frac{z_c}{g}\ddot{y} && \text{(A.26)}\end{aligned}$$

Appendix B

Power Processing Unit

B.1 Overview

Batteries which provide power to the motors do so in fixed form, namely with constant voltage. To modify the power to the motor, connections to the batteries (at ports A and B in figure B.1) are switched on and off at high speeds. Due to induction characteristics of the motor armature, the reaction of the current to this high frequency switching is buffered and can be effectively assumed to be a function of the *average* voltage at the ports. The Power Processing Unit, PPU, is responsible for the high frequency switching, effectively amplifying a reference signal, v_c, by a constant, K_{PWM} as shown the figure.

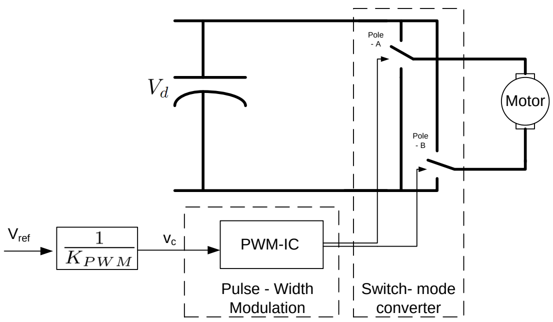


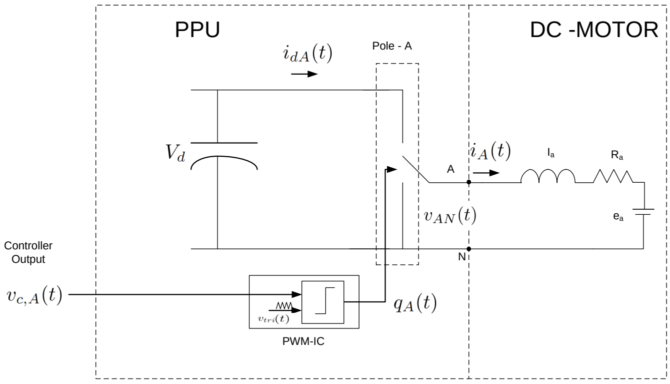
Figure B.1: Overview of the PPU

In order to gain incite in the open loop gains of the system as well as detecting any undesired dynamical effect the PPU may introduce to the overall system, the basic operations of the the PPU have been studied in this section.

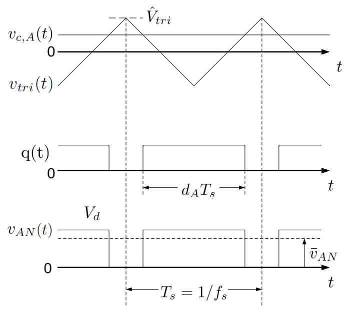
B.2 Pulse - Width Modulation

For clarity, it is best to start with a simplified model of the PPU whereby only one switching pole is present, as shown in figure B.2(a).

The PPU can be split up into 2 parts, namely the pulse-width modulation and switch mode converter part. The latter is responsible of the bi-positional switching, depending on the binary signal q(t). The former takes the control signal, $v_{c,A}(t)$, and compares this to a triangular trigger signal, $v_{tri}(t)$, of amplitude $\hat{V}_{tri}$ and switching frequency f_s. Output, q(t), then depends on the switching rule:



(a) PPU



(b) Signals

Figure B.2: A) Simplified block diagram of the PPU with one switching pole B) Output signals as a function of a particular input signal v_{AN}

$$\begin{aligned} v_{c,A} < v_{tri} \quad \rightarrow q &= 0 \\ v_{c,A} > v_{tri} \quad \rightarrow q &= 1 \end{aligned}$$

The voltage at pole-A , $v_{AN} = V_b q(t)$, is cut into pulses with width $d_A T_s$, where the duty-ratio, d_A, is a fraction of the switch time, $T_s = 1/f_s$. Hence

$$\bar{v}_{AN} = d_A V_d \tag{B.1}$$

The relationship is best described by example as shown in figure B.2(b). From inspection we see that changing the control voltage, $v_{c,A}$ by $2\hat{V}_{tri}$ changes the duty-ratio by unity, thus describing the relationship:

$$\frac{\Delta d_A}{\Delta v_{c,A}} = \frac{1}{2\hat{V}_{tri}} \tag{B.2}$$

This allows us the express the linear relationship between duty-ratio and control voltage:

$$d_A = \frac{1}{2\hat{V}_{tri}} v_{c,A} + Offset \qquad (-\hat{V}_{tri} \leq v_{c,A} \leq \hat{V}_{tri}) \tag{B.3}$$

Substituting values of an operating point, (for example $v_{c,A} = \hat{V}_{tri} \rightarrow d_A = 1$), we get $Offset = 1/2$ Inserting equation B.3 in B.1, we get a relationship for the motor voltage, v_{AN} as a function of the control signal, $v_{c,A}$

$$\bar{v}_{AN} = \frac{V_d}{2} + \frac{V_d}{2\hat{V}_{tri}} v_{c,A} \tag{B.4}$$

The switching function for the other pole (pole-B in figure B.1), works similarly to that for pole - A, however the controle voltage, $v_{c,B}$ is set to the negative of $v_{c,A}$. Assigning a general control input values, v_c to $v_{c,A}$, we can express the duty-ratios and subsequent average pole voltages as:

$$d_A = \frac{1}{2} + \frac{1}{2\hat{V}_{tri}} v_c \quad \rightarrow \quad \bar{v}_{AN}(t) = \frac{V_d}{2} + \frac{V_d}{2\hat{V}_{tri}} v_c(t) \tag{B.5}$$

$$d_B = \frac{1}{2} - \frac{1}{2\hat{V}_{tri}} v_c \quad \rightarrow \quad \bar{v}_{BN}(t) = \frac{V_d}{2} + \frac{V_d}{2\hat{V}_{tri}} v_c(t) \tag{B.6}$$

At the output terminal, the output voltage is the difference between the pole output voltage, therefor

$$\bar{v}_{AB} = \bar{v}_{AN} - \bar{v}_{BN} \tag{B.7}$$

$$= \frac{V_d}{\hat{V}_{tri}} v_c(t) \tag{B.8}$$

$$= k_{PWM} v_c(t) \tag{B.9}$$

Where $k_{PWM} = \frac{V_d}{\hat{V}_{tri}}$, is the constant $\bar{v}_{AB}/v_c$ amplifier gain.

Appendix C

appendix Modeling kinematics

DH parameters:

L_i	Link Description	a_i [m]	$d_i[m]$	α [rad]	θ_i [rad]	q_{offset}
1	virtual link	0	q_1	$\pi/2$	0	0
2	virtual link	0	q_2	$\pi/2$	$\pi/2$	0
3	virtual link	0	q_3	$\pi/2$	$\pi/2$	0
4	virtual link	0	0	$\pi/2$	q_4	$\pi/2$
5	virtual link	0	0	$\pi/2$	q_5	$\pi/2$
6	Right foot	0	0	$\pi/2$	q_6	$-\pi/2$
7	right ankle	L_1	0	$\pi/2$	q_7	$\pi/2$
8	right lower leg	L_2	0	0	q_8	0
9	right upper leg	L_3	0	0	q_9	0
10	right lower hip	L_4	$-L_6$	$-\pi/2$	q_{10}	0
11	right upper hip	0	0	$-\pi/2$	q_{11}	$\pi/2$
12	torso	L_7	0	π	q_{12}	0
13	left upper hip	0	0	$-\pi/2$	q_{13}	π
14	left lower hip	L_4	0	$-\pi/2$	q_{14}	$\pi/2$
15	left upper leg	L_3	L_6	0	q_{15}	0
16	left lower leg	L_2	0	0	q_{16}	0
17	left ankle	L_1	0	$\pi/2$	q_{17}	0
18	left foot	0	0	0	q_{18}	0

Contact Points, CP, in link coordinate frame:

CP_i	Description	Right Foot, $\underline{\text{r}}^{6}_{RCP_i}$	Left Foot, $\underline{\text{r}}^{18}_{LCP_i}$
1	Outside toe	$[\ L_{12};\ -L_{10};\ -L_8\ ]$	$[\ L_{10};\ L_{12};\ L_8\ ]$
2	Inside toe	$[\ -L_{11};\ -L_{10};\ -L_8\ ]$	$[\ L_{10};\ -L_{11};\ L_8\ ]$
3	Inside heel	$[\ -L_{11};\ -L_{10};\ L_9\ ]$	$[\ L_{10};\ -L_{11};\ -L_9\ ]$
4	Outside heel	$[\ L_{12};\ -L_{10};\ L_9\ ]$	$[\ L_{10};\ L_{12};\ -L_9\ ]$

Link inertial properties: Mass, Inertia, and Center of mass (CM) of link,L_i, in link coordinate frame, {i}:

L_i	Description	M_i	I_i^i	$\underline{r}_{CM}^i$
6	right foot	0.366	$(R^{w6})^T I_6^w R^{w6}$	$[\ -L_{10}/2;\ 0;\ L_8/2\ ]$
7	right ankle	0.614	$(R^{w7})^T I_7^w R^{w7}$	$[\ -L_1/2;\ 0;\ 0\ ]$
8	right lower leg	0.315	$(R^{w8})^T I_8^w R^{w8}$	$[\ -L_2/2;\ 0;\ 0\ ]$
9	right upper leg	2.141	$(R^{w9})^T I_9^w R^{w9}$	$[\ -L_3/2;\ 0;\ 0\ ]$
10	right lower hip	0.614	$(R^{w10})^T I_{10}^w R^{w10}$	$[\ -L_4;\ L_6;\ 0\ ]$
11	right upper hip	0.614	$(R^{w11})^T I_{11}^w R^{w11}$	$[\ 0;\ 0;\ -L_5/2\ ]$
12	torso	8.594	$(R^{w12})^T I_{12}^w R^{w12}$	$[\ -L_7/2;\ 0;\ L_5+0.17\ ]$
13	left upper hip	M_{11}	$(R^{w13})^T I_{11}^w R^{w13}$	$[\ 0;\ -L_5/2;\ 0\ ]$
14	left lower hip	M_{10}	$(R^{w14})^T I_{10}^w R^{w14}$	$[\ 0;\ 0;\ 0\ ]$
15	left upper leg	M_9	$(R^{w15})^T I_9^w R^{w15}$	$[\ -L_3/2;\ 0;\ 0\ ]$
16	left lower leg	M_8	$(R^{w16})^T I_8^w R^{w16}$	$[\ -L_2/2;\ 0;\ 0\ ]$
17	left ankle	M_7	$(R^{w17})^T I_7^w R^{w17}$	$[\ -L_1/2;\ 0;\ 0\ ]$
18	left foot	M_6	$(R^{w18})^T I_6^w R^{w18}$	$[\ L_{10}/2;\ 0;\ L_8/2\ ]$

Rotation matrix from link to world coordinate frame, R^{wi}, in zero pose, $\underline{q}_z = \underline{0}$:

L_i	Link name	$R^{wi}(q_z)$
6	right foot	[0 0 -1 ; -1 0 0 ; 0 1 0]
7	right ankle	[0 -1 0 ; 0 0 -1 ; 1 0 0]
8	right lower leg	[0 -1 0 ; 0 0 -1 ; 1 0 0]
9	right upper leg	[0 -1 0 ; 0 0 -1 ; 1 0 0]
10	right lower hip	[0 0 -1 ; 0 1 0 ; -1 0 0]
11	right upper hip	[0 1 0 ; 1 0 0 ; 0 0 -1]
12	torso	[0 -1 0 ; 1 0 0 ; 0 0 1]
13	left upper hip	[0 0 1 ; -1 0 0 ; 0 -1 0]
14	left lower hip	[0 -1 0 ; 0 0 1 ; -1 0 0]
15	left upper leg	[0 -1 0 ; 0 0 1 ; -1 0 0]
16	left lower leg	[0 -1 0 ; 0 0 1 ; -1 0 0]
17	left ankle	[0 0 1 ; 0 1 0 ; -1 0 0]
18	left foot	[0 0 1 ; 0 1 0 ; -1 0 0]

C.1 Simulation error sensitivity to inertial matrix condition number

The EOM, (3.7) can be expressed as a set of linear equations, $Mx = b$, where M is the inertia matrix, x are joint the accelerations and $b = \tau - C - F$. It can be easily shown that the sensitivity with respect to a perturbation E in A, is directly related to the condition number of M. Let y be the solution of

$$\begin{aligned}
(M+E)y &= b && \text{(C.1)}\\
\Rightarrow M^{-1}(M+E)y &= M^{-1}b && \text{(C.2)}\\
y + M^{-1}Ey &= x && \text{(C.3)}\\
y - x &= M^{-1}Ey && \text{(C.4)}\\
\|y-x\|/\|y\| &\leq \|M^{-1}E\| && \text{(C.5)}\\
\|y-x\|/\|y\| &\leq \varepsilon\|M\|\|M^{-1}\| && \text{(C.6)}\\
\|y-x\|/\|y\| &\leq \varepsilon\kappa(M) && \text{(C.7)}
\end{aligned}$$

Where ε is the relative error in A, $\varepsilon = \|E\|/\|A\|$, and $\kappa(M)$ is the condition number of M, $\kappa(M) = \|M\|\|M^{-1}\|$.

C.2 Constraint Method

Hard contact has a non-penetrating condition which requires that

$$h \geq 0. \tag{C.8}$$

Forces that impose this constraint λ, are equally one sided, acting to prevent penetration but to not separation, therefore:

$$\lambda \geq 0 \tag{C.9}$$

Briefly simplifying the model of tulip to just one body for clarity, we can define that 4 distinct states, shown in figure C.2, which system will always find itself in. When $h(q) > 0$, state 1; there is no contact and (3.7) remains unchanged. State 2; $h(q) = 0$ and $\dot{h} < 0$, occurs just before impact, time t^-. Newton's second law of motion states that the rate of change in momentum is equal to applied force, $F = \dot{P} = m\dot{v}$. Therefore the bodies will clearly penetrate at the next time instant, t^+, unless an impuls, S, is applied, where S is

$$S = \int_{\Delta t \to 0} F(t)dt = m\Delta v = m(v(t^+) - v(t^-)). \tag{C.10}$$

For the case of Tulip, the mass m is the multibody mass matrix $\mathbf{M}$. Also notice that taking the time derivative of the gap expression, h(q), we get the expression for the relative velocity, and using Newton law of restitution [50] can be used to find relative velocities of colliding bodies before and after collision:

$$\frac{h(q)}{dt} = W^T\dot{q} \tag{C.11}$$

$$\Rightarrow W^T\dot{q}^- = eW^T\dot{q}^+ \tag{C.12}$$

Where $W^T = \frac{\partial h}{\partial q}$ and $\dot{q}^-$,$\dot{q}^+$ are the relative velocities just before and after after impact.

Combining Newton's impact law, with the conservation of momentum and applying the Lagrange multiplier theorem, [64], we get the impact equations:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{M} \\ W^T & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ \dot{q}^- \end{bmatrix} = \begin{bmatrix} W\lambda \\ -e^T\dot{q}^- \end{bmatrix} \tag{C.13}$$

C.2.1 Integration

At certain moments during the simulation, known as a switch points, a change in the system dynamics will occur when a constraint becomes active or inactive, or when an impact occurs. The integration would therefore have to stop and and restart with a different state and/or model. This start stop characteristic is illustrated in Figure C.3, showing the flow of the integration.

Checking for these switch points, referred to as collision detection, needs to be doen frequently and ideally when found the integration proces needs revers and find the moment of the switch point to an even higher degree of accuracies. In Matlab this can be achieved by passing the collision detection function as an argument of the integration routine. As the integration process is now not only a function of the EOM and time, the integration is often referred to as event-driven integration. This event driven method is necessary due to the changes in the internal structure of the dynamics equations at so called switch moments.

The stop-start behavior of event-driven integration can considerably slow down the simulation. An even bigger problem arises if partially elastic collisions, rather than inelastic collision (e=0), are considered. This is best explained by examining the behavior of a bouncing ball. Each time the ball collides with the ground it will lose a fraction of its energy and it's resulting speed will approach zero. As the time span between collisions depends on this speed, this too approaches zero and therefore the number of collisions over time approaches infinity.

This problem can be avoided using so called time-stepping integration schemes [13, 53]. The fundamentals of this approach were pioneered by Moreau [33] who formulated unilateral contact forces as set valued force laws which were used to describe the dynamics as measure differential inclusion problems. This integration method works on the integral of the contact forces during a given time step, not with the forces themselves which makes the method insensitive to the exact moment of impact. The method treats the state of the system as constant during the complete time interval. A drawback is that the numerical accuracies of such integration (for example using Moreau's midpoint method [24]) is low. This can be improved using a variable step size strategy [54].

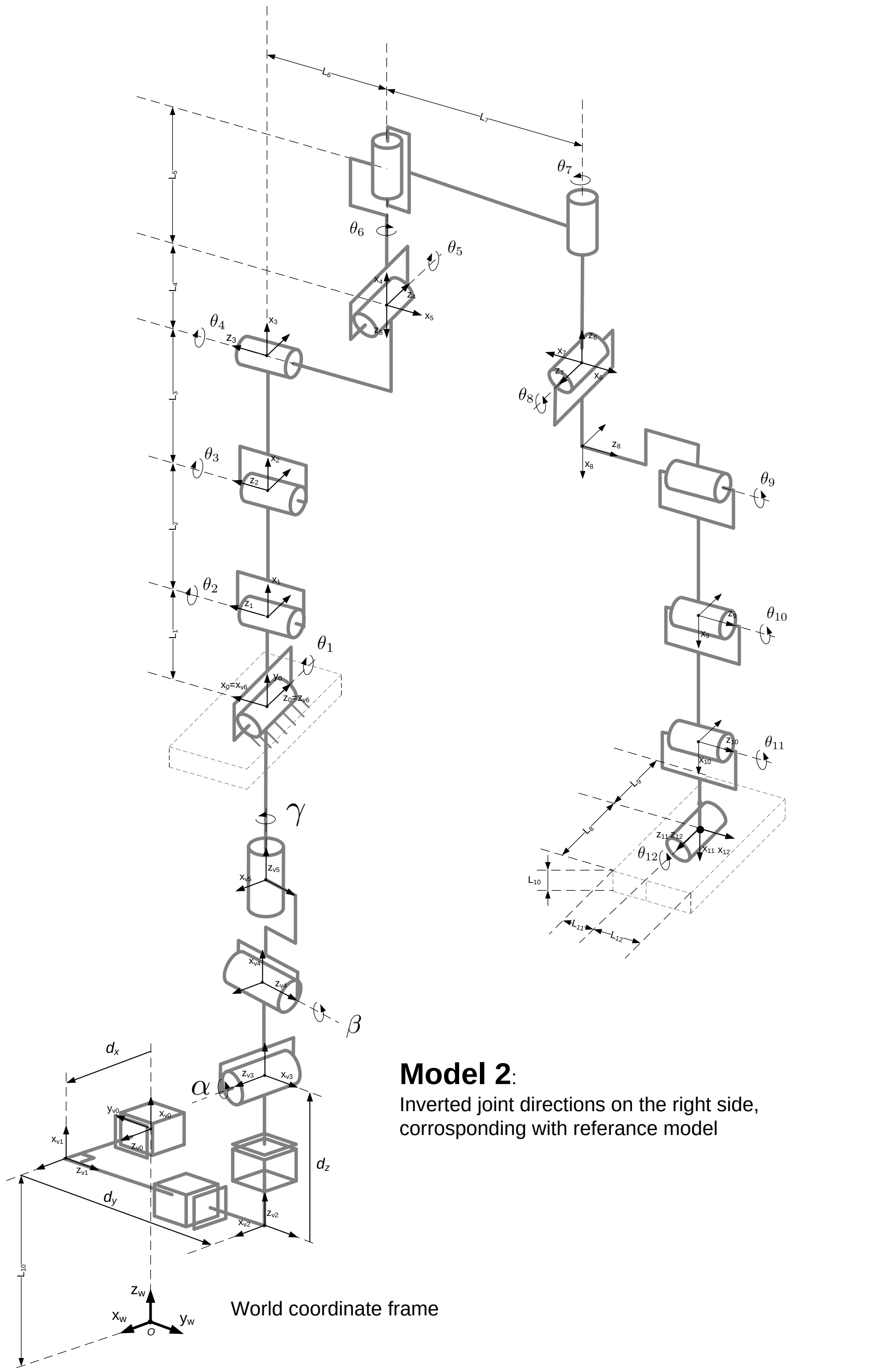


Figure C.1: Schematic view of the kinematic model of TUlip with associated DH coordinate frames. (joint directions defined as in Tulip?)

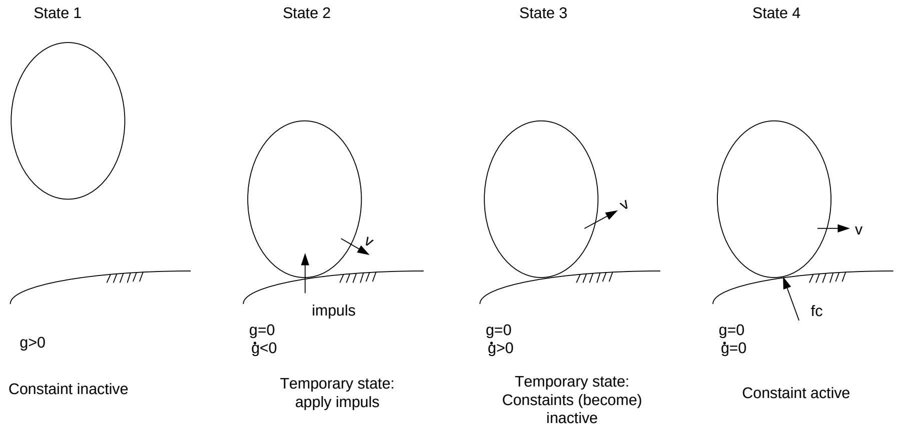


Figure C.2: Possible states of colliding bodies.

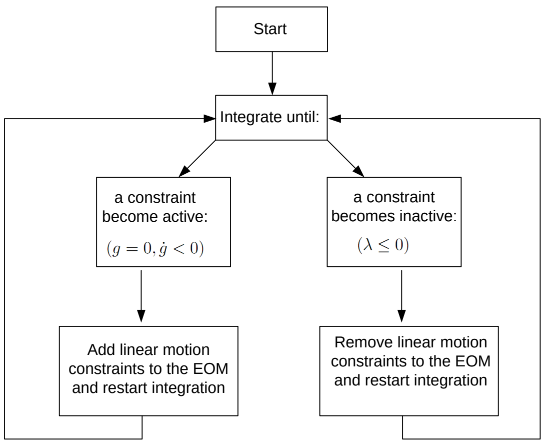


Figure C.3: Flowchart showing event driven integration used to simulate purely sticking friction behavior.

Appendix D

Simple 2D model

In this appendix, manual construction of symbolic equations of motion was analyzed using a 3 body model of an actuated leg in 2D. The leg model has an actuated ankle and knee joint and is free to move in 2D space. This simple model, shown in figure D.1, was used to better understand the complexity of the problem and help decide whether or not a multibody simulation software should be used. The Denavit-Hartenberg, DH, convention was used to describe the system.

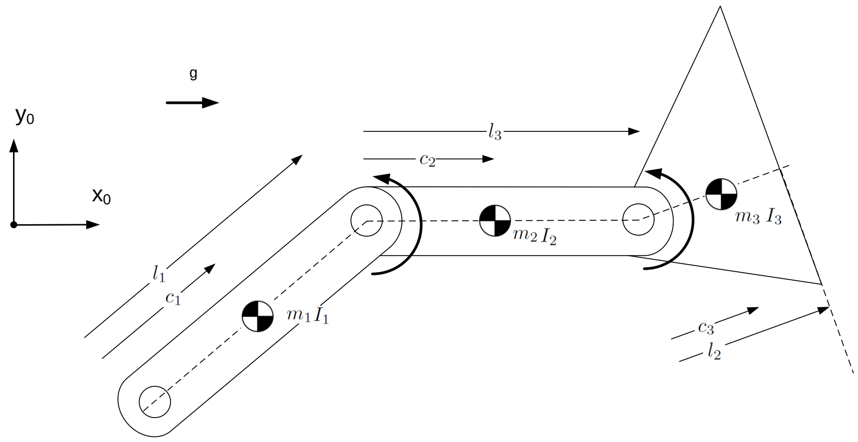


Figure D.1: Parameters of simple leg model.

D.1 Lagrange Equations

To derive the EOM we start with start with D'Alembert's principle for the virtual work of applied forces, $\mathbf{F}_i$, and inertial forces on a three dimensional accelerating system of n particles, i, whose motion is consistent with its constraints,

$$\delta W = \sum_{i=1}^{n}(\mathbf{F}_i - m_i\mathbf{a}_i)\cdot\delta\mathbf{r}_i = 0 \tag{D.1}$$

$$= \sum_{i=1}^{n}\mathbf{F}_i\cdot\delta\mathbf{r}_i - \sum_{i=1}^{n}m_i\mathbf{a}_i\cdot\delta\mathbf{r}_i = 0 \tag{D.2}$$

where

- W is the virtual work;

- $\delta\mathbf{r}_i$ is the virtual displacement consistent with the constraints;
- m_i and a_i are the mass and acceleration of particle i;
- n total number of particles in the system.

For the system under consideration (figure D.1), the position each particle can clearly be expressed as a function of 5 independant generalized coordinates q_i:

$$\begin{aligned} \mathbf{r}_1 &= \mathbf{r}_1(q_1, q_2, \ldots, q_5) \\ \mathbf{r}_2 &= \mathbf{r}_2(q_1, q_2, \ldots, q_5) \\ &\vdots \\ \mathbf{r}_n &= \mathbf{r}_n(q_1, q_2, \ldots, q_5) \end{aligned} \tag{D.3}$$

The virtual displacement $\delta\mathbf{r}_i$ can thusly be described as

$$\delta\mathbf{r}_i = \sum_{j=1}^{m=5} \frac{\partial\mathbf{r}_i}{\partial q_j}\delta q_j \tag{D.4}$$

Next we define the generalized forces, Q_j, as:

$$Q_j = \sum_{i=1}^{n} \mathbf{F}_i \cdot \frac{\partial\mathbf{r}_i}{\partial q_j} \tag{D.5}$$

Substituting D.9 and D.4 into D.2 we get:

$$\delta W = \sum_{j=1}^{m=5} Q_j\delta q_j - \sum_{j=1}^{m=5}\sum_{i=1}^{n} m_i\mathbf{a}_i \cdot \frac{\partial\mathbf{r}_i}{\partial q_j}\delta q_j = 0 \tag{D.6}$$

$$\sum_{j=1}^{m=5} Q_j\delta q_j - \sum_{j=1}^{m=5}\left(\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_j}\right) - \frac{\partial T}{\partial q_j}\right)\delta q_j = 0 \tag{D.7}$$

Where the inertial forces in D.6 have been expressed as a function of the kinetic energy, T. As D.7 holds for arbitrary ∂q_j, we can write:

$$Q_j = \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_j}\right) - \frac{\partial T}{\partial q_j} \tag{D.8}$$

The final step is to separate the applied forces into conservative and non-conservative terms: $F_i = Fc_j + Fnc_j$. Substituting this into D.9, we get

$$\begin{aligned} Q_j &= \sum_{i=1}^{n} \mathbf{F}c_i \cdot \frac{\partial\mathbf{r}_i}{\partial q_j} + \sum_{i=1}^{n} \mathbf{F}nc_i \cdot \frac{\partial\mathbf{r}_i}{\partial q_j} && \text{(D.9)} \\ &= Qc_j + Qnc_j && \text{(D.10)} \end{aligned}$$

Where $Qc_j = \sum_{i=1}^{n} \mathbf{F}c_i \cdot \frac{\partial\mathbf{r}_i}{\partial q_j}$ and $Qnc_j = \sum_{i=1}^{n} \mathbf{F}nc_i \cdot \frac{\partial\mathbf{r}_i}{\partial q_j}$, are defined as the conservative and non-conservative generalized forces. Conservative forces $\mathbf{F}c_i$ can be represented by a scalar potential field, V (in this example, a gravitational field, $V = -m_i\vec{g} \cdot \vec{r}_i$), thus:

$$\mathbf{F}c_i = -\nabla V \Rightarrow G_j = -\sum_{i=1}^{n} \nabla V \cdot \frac{\partial\mathbf{r}_i}{\partial q_j} = -\frac{\partial V}{\partial q_j} \tag{D.11}$$

The EOM are derived using the lagrange equation D.12, which combines conservation of momentum with conservation of energy.

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = \tau \tag{D.12}$$

Where q are a set of generalised coordinates, τ is the vector of generalized applied forces and $\mathcal{L}$, called Lagrangian, is equal to the kinetic T, minus the potential V, energy: $\mathcal{L} = T - V$. Therefor we can write:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} + \frac{\partial V}{\partial q} = \tau \tag{D.13}$$

D.2 Denavit-Hartenberg approach

For the example model, the standard DH formulism leads to the coordinate frames shown in table D.1 and figure D.2

L_i	Link Description	a_i [m]	$d_i[m]$	α [rad]	θ_i [rad]
1	virtual link	0	0	$\pi/2$	q_1
2	virtual link	0	q_2	$-\pi/2$	0
3	upper leg	l_1	0	0	q_3
4	lower leg	l_2	0	0	q_4
5	foot	l_3	0	0	q_5

Table D.1: DH - parameters of leg model

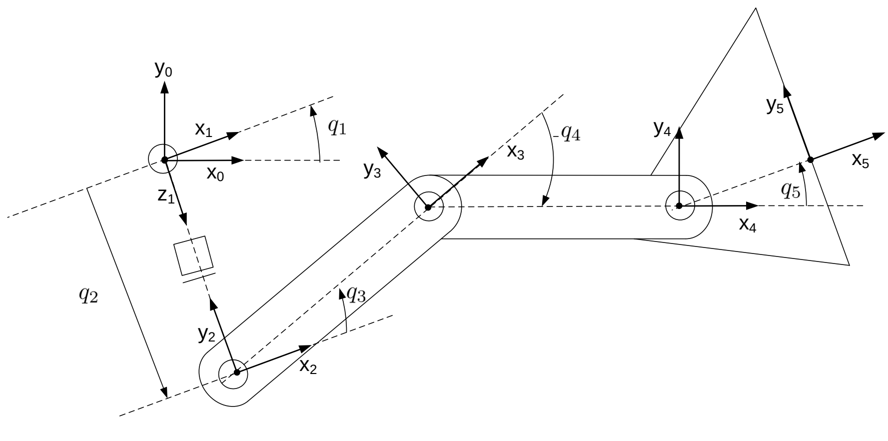


Figure D.2: Coordinates of simple leg model, assigned using DH convention.

For every link/joint pair the homogenous coordinate transformation from the previous coordinate system to the next coordinate system is described as

Inserting the DH parameters, table D.1, to 3.1, we get the homogenous transformations: $A_1^0(q_1)$, $A_2^1(q_2)$,..., $A_5^4(q_5)$. The Homogenous transformation of the body fixed coordinate frames,$\vec{e}^{i+20}$, wrt the inertial frame, $\vec{e}^0$, are then found: $T_3^0 = A_1^0 A_2^1 A_3^2$, $T_4^0 = T_3^0 A_4^3$ and $T_5^0 = T_4^0 A_5^4$. These relations can be used to express the positions of the center of masses wrt O_0, p_{ci}^0, for bodies $i = 1:3$:

$$
\begin{aligned}
\begin{bmatrix} p_{ci}^0 \\ 1 \end{bmatrix} &= T_{2+i}^0(q_1, q_2..q_{i+2}) \begin{bmatrix} p_{ci}^{2+i} \\ 1 \lhd \end{bmatrix} && \text{(D.14)} \\
&= T_{2+i}^0(q_1, q_2..q_{i+2}) \begin{bmatrix} c_i - l_i \\ 0 \\ 0 \\ 1 \end{bmatrix} && \text{(D.15)} \\
&= \begin{bmatrix} & R_{i+2}^0(q_1, q_2..q_{i+2}) & & o_{i+2}^0(q_1, q_2..q_{i+2}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_i - l_i \\ 0 \\ 0 \\ 1 \end{bmatrix} && \text{(D.16)} \\
\Rightarrow p_{ci}^0 &= R_{i+2}^0(q_1, q_2..q_{i+2}) \begin{bmatrix} c_i - l_i \\ 0 \\ 0 \end{bmatrix} + o_{i+2}^0(q_1, q_2..q_{i+2}) && \text{(D.17)}
\end{aligned}
$$

Differentiating p_{ci}^0 and using the addition proportie of angular velocity [51], we find $Jc_{vi}, Jc_{\omega i}$, the linear and angular velocity jacobian at the center of mass for bodies i:

$$
\begin{aligned}
Jc_{vi} &= \begin{bmatrix} J_{v1} & J_{v2} & J_{v3} & J_{v4} & J_{v5} \end{bmatrix} && \text{(D.18)} \\
Jc_{wi} &= \begin{bmatrix} J_{w1} & J_{w2} & J_{w3} & J_{v4} & J_{v5} \end{bmatrix}, && \text{(D.19)}
\end{aligned}
$$

with

$$
J_{vj} = \begin{cases} z_{j-1}^0 \times (p_{ci}^0 - o_{j-1}^0) & \text{for revolute joint j} \quad \text{(D.20)} \\ z_{j-1}^0 & \text{for prismatic joint j} \quad \text{(D.21)} \\ 0 & \text{if } j > i \quad \text{(D.22)} \end{cases}
$$

and

$$
J_{wj} = \begin{cases} z_{j-1}^0 & \text{for revolute joint j} \quad \text{(D.23)} \\ 0 & \text{for prismatic joint j or if } j > i \quad \text{(D.24)} \end{cases}
$$

D.3 M(q)

Elements of the inertia matrix

$$
\begin{aligned}
m_{11} &= -\mathrm{q2}\ (2\,\mathrm{l1}\,\mathrm{m2}\ \sin(\mathrm{q3}) + 2\,\mathrm{l1}\,\mathrm{m3}\ \sin(\mathrm{q3}) + 2\,\mathrm{lc1}\,\mathrm{m1}\ \sin(\mathrm{q3}) + 2\,\mathrm{lc3}\,\mathrm{m3}\ \sin(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) + 2\,\mathrm{l2}\,\mathrm{m3}\ \sin(\mathrm{q3}+\mathrm{q4}) + 2\,\mathrm{lc2}\,\mathrm{m2}\ \sin(\mathrm{q3}+\mathrm{q4})) + \ldots \\
&\quad \mathrm{q2}^2\ (\mathrm{m1}+\mathrm{m2}+\mathrm{m3}) + \mathrm{Izz}_1 + \mathrm{Izz}_2 + \mathrm{Izz}_3 + \mathrm{l1}^2\,\mathrm{m2} + \mathrm{l1}^2\,\mathrm{m3} + \mathrm{l2}^2\,\mathrm{m3} + \mathrm{lc1}^2\,\mathrm{m1} + \mathrm{lc2}^2\,\mathrm{m2} + \mathrm{lc3}^2\,\mathrm{m3} + 2\,\mathrm{l1}\,\mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q4}+\mathrm{q5}) + 2\,\mathrm{l1}\,\mathrm{l2}\,\mathrm{m3}\ \cos(\mathrm{q4}) + \ldots \\
&\quad 2\,\mathrm{l1}\,\mathrm{lc2}\,\mathrm{m2}\ \cos(\mathrm{q4}) + 2\,\mathrm{l2}\,\mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q5}) && \text{(D.25)} \\
m_{12} &= -\mathrm{l1}\,\mathrm{m2}\ \cos(\mathrm{q3}) - \mathrm{l1}\,\mathrm{m3}\ \cos(\mathrm{q3}) - \mathrm{lc1}\,\mathrm{m1}\ \cos(\mathrm{q3}) - \mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) - \mathrm{l2}\,\mathrm{m3}\ \cos(\mathrm{q3}+\mathrm{q4}) - \mathrm{lc2}\,\mathrm{m2}\ \cos(\mathrm{q3}+\mathrm{q4}) && \text{(D.26)} \\
m_{13} &= \mathrm{Izz}_1 + \mathrm{Izz}_2 + \mathrm{Izz}_3 + \mathrm{l1}^2\,\mathrm{m2} - \mathrm{q2}\ (\mathrm{l1}\,\mathrm{m2}\ \sin(\mathrm{q3}) + \mathrm{l1}\,\mathrm{m3}\ \sin(\mathrm{q3}) + \mathrm{lc1}\,\mathrm{m1}\ \sin(\mathrm{q3}) + \mathrm{lc3}\,\mathrm{m3}\ \sin(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) + \mathrm{l2}\,\mathrm{m3}\ \sin(\mathrm{q3}+\mathrm{q4}) + \mathrm{lc2}\,\mathrm{m2}\ \sin(\mathrm{q3}+\mathrm{q4})) + \ldots \\
&\quad \mathrm{m3}\,\mathrm{l1}^2 + 2\,\mathrm{m3}\ \cos(\mathrm{q4})\ \mathrm{l1}\,\mathrm{l2} + 2\,\mathrm{m2}\ \cos(\mathrm{q4})\ \mathrm{l1}\,\mathrm{lc2} + 2\,\mathrm{m3}\ \cos(\mathrm{q4}+\mathrm{q5})\ \mathrm{l1}\,\mathrm{lc3} + \mathrm{m3}\,\mathrm{l2}^2 + 2\,\mathrm{m3}\ \cos(\mathrm{q5})\ \mathrm{l2}\,\mathrm{lc3} + \mathrm{m1}\,\mathrm{lc1}^2 + \mathrm{m2}\,\mathrm{lc2}^2 + \mathrm{m3}\,\mathrm{lc3}^2 && \text{(D.27)} \\
m_{14} &= \mathrm{Izz}_2 + \mathrm{Izz}_3 + \mathrm{l2}^2\,\mathrm{m3} + \mathrm{lc2}^2\,\mathrm{m2} + \mathrm{lc3}^2\,\mathrm{m3} + \mathrm{l1}\,\mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q4}+\mathrm{q5}) - \mathrm{l2}\,\mathrm{m3}\,\mathrm{q2}\ \sin(\mathrm{q3}+\mathrm{q4}) - \mathrm{lc2}\,\mathrm{m2}\,\mathrm{q2}\ \sin(\mathrm{q3}+\mathrm{q4}) + \mathrm{l1}\,\mathrm{l2}\,\mathrm{m3}\ \cos(\mathrm{q4}) + \mathrm{l1}\,\mathrm{lc2}\,\mathrm{m2}\ \cos(\mathrm{q4}) + \ldots \\
&\quad 2\,\mathrm{l2}\,\mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q5}) - \mathrm{lc3}\,\mathrm{m3}\,\mathrm{q2}\ \sin(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) && \text{(D.28)} \\
m_{15} &= \mathrm{Izz}_3 + \mathrm{lc3}^2\,\mathrm{m3} + \mathrm{l1}\,\mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q4}+\mathrm{q5}) + \mathrm{l2}\,\mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q5}) + \ldots \\
&\quad \mathrm{lc3}\,\mathrm{m3}\,\mathrm{q2}\ \sin(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) && \text{(D.29)} \\
m_{22} &= \mathrm{m1}+\mathrm{m2}+\mathrm{m3} && \text{(D.30)} \\
m_{23} &= -\mathrm{l1}\,\mathrm{m2}\ \cos(\mathrm{q3}) - \mathrm{l1}\,\mathrm{m3}\ \cos(\mathrm{q3}) - \mathrm{lc1}\,\mathrm{m1}\ \cos(\mathrm{q3}) - \mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) - \mathrm{l2}\,\mathrm{m3}\ \cos(\mathrm{q3}+\mathrm{q4}) - \mathrm{lc2}\,\mathrm{m2}\ \cos(\mathrm{q3}+\mathrm{q4}) && \text{(D.31)} \\
m_{24} &= -\mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) - \mathrm{l2}\,\mathrm{m3}\ \cos(\mathrm{q3}+\mathrm{q4}) - \mathrm{lc2}\,\mathrm{m2}\ \cos(\mathrm{q3}+\mathrm{q4}) && \text{(D.32)} \\
m_{25} &= -\mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) && \text{(D.33)} \\
m_{33} &= \mathrm{Izz}_1 + \mathrm{Izz}_2 + \mathrm{Izz}_3 + \mathrm{l1}^2\,\mathrm{m2} + \mathrm{l1}^2\,\mathrm{m3} + \mathrm{l2}^2\,\mathrm{m3} + \mathrm{lc1}^2\,\mathrm{m1} + \mathrm{lc2}^2\,\mathrm{m2} + \mathrm{lc3}^2\,\mathrm{m3} + 2\,\mathrm{l1}\,\mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q4}+\mathrm{q5}) + 2\,\mathrm{l1}\,\mathrm{l2}\,\mathrm{m3}\ \cos(\mathrm{q4}) + \ldots && \text{(D.34)} \\
&\quad 2\,\mathrm{l1}\,\mathrm{lc2}\,\mathrm{m2}\ \cos(\mathrm{q4}) + 2\,\mathrm{l2}\,\mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q5}) \\
m_{34} &= \mathrm{m3}\,\mathrm{l2}^2 + 2\,\mathrm{m3}\ \cos(\mathrm{q5})\ \mathrm{l2}\,\mathrm{lc3} + \mathrm{l1}\,\mathrm{m3}\ \cos(\mathrm{q4})\ \mathrm{l2} + \mathrm{m2}\,\mathrm{lc2}^2 + \mathrm{l1}\,\mathrm{m2}\ \cos(\mathrm{q4})\ \mathrm{lc2} + \mathrm{m3}\,\mathrm{lc3}^2 + \mathrm{l1}\,\mathrm{m3}\ \cos(\mathrm{q4}+\mathrm{q5})\ \mathrm{lc3} + \mathrm{Izz}_2 + \mathrm{Izz}_3 && \text{(D.35)} \\
m_{35} &= \mathrm{Izz}_3 + \mathrm{lc3}^2\,\mathrm{m3} + \mathrm{l1}\,\mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q4}+\mathrm{q5}) + \mathrm{l2}\,\mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q5}) && \text{(D.36)} \\
m_{44} &= \mathrm{m3}\,\mathrm{l2}^2 + 2\,\mathrm{m3}\ \cos(\mathrm{q5})\ \mathrm{l2}\,\mathrm{lc3} + \mathrm{m2}\,\mathrm{lc2}^2 + \mathrm{m3}\,\mathrm{lc3}^2 + \mathrm{Izz}_2 + \mathrm{Izz}_3 && \text{(D.37)} \\
m_{45} &= \mathrm{m3}\,\mathrm{lc3}^2 + \mathrm{l2}\,\mathrm{m3}\ \cos(\mathrm{q5})\ \mathrm{lc3} + \mathrm{Izz}_3 && \text{(D.38)} \\
m_{55} &= \mathrm{Izz}_3 + \mathrm{lc3}^2\,\mathrm{m3} + \mathrm{l1}\,\mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q4}+\mathrm{q5}) + \mathrm{l2}\,\mathrm{lc3}\,\mathrm{m3}\ \cos(\mathrm{q5}) && \text{(D.39)} \\
& && \text{(D.40)}
\end{aligned}
$$

Note that the other off diagonal terms $M_{ij} = M_{ji}$ due to symmetry in the inertial matrix.

D.4 C(q)

Matrix C(q) appears in the term representing Coriolis and centripetal effects

$$
\begin{aligned}
c_{11} &= -\mathrm{qd2}\ (\mathrm{l1\,m2\,\sin(q3) - q2\ (m1 + m2 + m3) + l1\,m3\,\sin(q3) + lc1\,m1\,\sin(q3) + lc3\,m3\,\sin(q3 + q4 + q5) + l2\,m3\,\sin(q3 + q4) + lc2\,m2\,\sin(q3 + q4)}) + \ldots \\
&\quad -\mathrm{qd4}\ (\mathrm{q2\ (lc3\,m3\,\cos(q3 + q4 + q5) + l2\,m3\,\cos(q3 + q4) + lc2\,m2\,\cos(q3 + q4)) + l1\,lc3\,m3\,\sin(q4 + q5) + l1\,l2\,m3\,\sin(q4) + l1\,lc2\,m2\,\sin(q4)}) + \ldots \\
&\quad -\mathrm{q2\,qd3}\ (\mathrm{l1\,m2\,\cos(q3) + l1\,m3\,\cos(q3) + lc1\,m1\,\cos(q3) + lc3\,m3\,\cos(q3 + q4 + q5) + l2\,m3\,\cos(q3 + q4) + lc2\,m2\,\cos(q3 + q4)}) + \ldots \\
&\quad -\mathrm{lc3\,m3\,qd5}\ (\mathrm{l1\,\sin(q4 + q5) + l2\,\sin(q5) + q2\,\cos(q3 + q4 + q5)}) && \text{(D.41)} \\
c_{12} &= -\mathrm{qd1}\ (\mathrm{l1\,m2\,\sin(q3) - q2\ (m1 + m2 + m3) + l1\,m3\,\sin(q3) + lc1\,m1\,\sin(q3) + lc3\,m3\,\sin(q3 + q4 + q5) + l2\,m3\,\sin(q3 + q4) + lc2\,m2\,\sin(q3 + q4)}) && \text{(D.42)} \\
c_{13} &= -\mathrm{qd4}\ (\mathrm{q2\ (lc3\,m3\,\cos(q3 + q4 + q5) + l2\,m3\,\cos(q3 + q4) + lc2\,m2\,\cos(q3 + q4)) + l1\,lc3\,m3\,\sin(q4 + q5) + l1\,l2\,m3\,\sin(q4) + l1\,lc2\,m2\,\sin(q4)}) + \ldots \\
&\quad -\mathrm{q2\,qd1}\ (\mathrm{l1\,m2\,\cos(q3) + l1\,m3\,\cos(q3) + lc1\,m1\,\cos(q3) + lc3\,m3\,\cos(q3 + q4 + q5) + l2\,m3\,\cos(q3 + q4) + lc2\,m2\,\cos(q3 + q4)}) + \ldots \\
&\quad -\mathrm{q2\,qd3}\ (\mathrm{l1\,m2\,\cos(q3) + l1\,m3\,\cos(q3) + lc1\,m1\,\cos(q3) + lc3\,m3\,\cos(q3 + q4 + q5) + l2\,m3\,\cos(q3 + q4) + lc2\,m2\,\cos(q3 + q4)}) + \ldots \\
&\quad -\mathrm{lc3\,m3\,qd5}\ (\mathrm{l1\,\sin(q4 + q5) + l2\,\sin(q5) + q2\,\cos(q3 + q4 + q5)}) && \text{(D.43)} \\
c_{14} &= -\mathrm{qd1}\ (\mathrm{q2\ (lc3\,m3\,\cos(q3 + q4 + q5) + l2\,m3\,\cos(q3 + q4) + lc2\,m2\,\cos(q3 + q4)) + l1\,lc3\,m3\,\sin(q4 + q5) + l1\,l2\,m3\,\sin(q4) + l1\,lc2\,m2\,\sin(q4)}) + \ldots \\
&\quad -\mathrm{qd3}\ (\mathrm{q2\ (lc3\,m3\,\cos(q3 + q4 + q5) + l2\,m3\,\cos(q3 + q4) + lc2\,m2\,\cos(q3 + q4)) + l1\,lc3\,m3\,\sin(q4 + q5) + l1\,l2\,m3\,\sin(q4) + l1\,lc2\,m2\,\sin(q4)}) + \ldots \\
&\quad -\mathrm{qd4}\ (\mathrm{q2\ (lc3\,m3\,\cos(q3 + q4 + q5) + l2\,m3\,\cos(q3 + q4) + lc2\,m2\,\cos(q3 + q4)) + l1\,lc3\,m3\,\sin(q4 + q5) + l1\,l2\,m3\,\sin(q4) + l1\,lc2\,m2\,\sin(q4)}) + \ldots \\
&\quad -\mathrm{lc3\,m3\,qd5}\ (\mathrm{l1\,\sin(q4 + q5) + l2\,\sin(q5) + q2\,\cos(q3 + q4 + q5)}) && \text{(D.44)} \\
c_{15} &= -\mathrm{lc3\,m3}\ (\mathrm{l1\,\sin(q4 + q5) + l2\,\sin(q5) + q2\,\cos(q3 + q4 + q5)})\ (\mathrm{qd1 + qd3 + qd4 + qd5}) && \text{(D.45)} \\
c_{21} &= \mathrm{qd1}\ (l_1\,\mathrm{m2\,\sin(q3) - m2\,q2 - m3\,q2 - m1\,q2} + l_1\,\mathrm{m3\,\sin(q3)} + \mathrm{lc_1\,m1\,\sin(q3)} + \mathrm{lc_3\,m3\,\sin(q3 + q4 + q5)} + l_2\,\mathrm{m3\,\sin(q3 + q4)} + \mathrm{lc_2\,m2\,\sin(q3 + q4)}) + .. \\
&\quad \mathrm{qd3}\ (l_1\,\mathrm{m2\,\sin(q3)} + l_1\,\mathrm{m3\,\sin(q3)} + \mathrm{lc_1\,m1\,\sin(q3)} + \mathrm{lc_3\,m3\,\sin(q3 + q4 + q5)} + l_2\,\mathrm{m3\,\sin(q3 + q4)} + \mathrm{lc_2\,m2\,\sin(q3 + q4)}) + .. \\
&\quad \mathrm{qd4}\ (\mathrm{lc_3\,m3\,\sin(q3 + q4 + q5)} + l_2\,\mathrm{m3\,\sin(q3 + q4)} + \mathrm{lc_2\,m2\,\sin(q3 + q4)}) + \mathrm{lc_3\,m3\,qd5\,\sin(q3 + q4 + q5)} && \text{(D.46)} \\
c_{22} &= 0 && \text{(D.47)} \\
c_{23} &= \mathrm{qd4}\ (\mathrm{lc_3\,m3\,\sin(q3 + q4 + q5)} + l_2\,\mathrm{m3\,\sin(q3 + q4)} + \mathrm{lc_2\,m2\,\sin(q3 + q4)}) + \mathrm{lc_3\,m3\,qd5\,\sin(q3 + q4 + q5)} + .. \\
&\quad \mathrm{qd1}\ (l_1\,\mathrm{m2\,\sin(q3)} + l_1\,\mathrm{m3\,\sin(q3)} + \mathrm{lc_1\,m1\,\sin(q3)} + \mathrm{lc_3\,m3\,\sin(q3 + q4 + q5)} + l_2\,\mathrm{m3\,\sin(q3 + q4)} + \mathrm{lc_2\,m2\,\sin(q3 + q4)}) + .. \\
&\quad \mathrm{qd3}\ (l_1\,\mathrm{m2\,\sin(q3)} + l_1\,\mathrm{m3\,\sin(q3)} + \mathrm{lc_1\,m1\,\sin(q3)} + \mathrm{lc_3\,m3\,\sin(q3 + q4 + q5)} + l_2\,\mathrm{m3\,\sin(q3 + q4)} + \mathrm{lc_2\,m2\,\sin(q3 + q4)}) && \text{(D.48)} \\
c_{24} &= \mathrm{qd1}\ (\mathrm{lc_3\,m3\,\sin(q3 + q4 + q5)} + l_2\,\mathrm{m3\,\sin(q3 + q4)} + \mathrm{lc_2\,m2\,\sin(q3 + q4)}) + \mathrm{lc_3\,m3\,qd5\,\sin(q3 + q4 + q5)} + .. \\
&\quad \mathrm{qd3}\ (\mathrm{lc_3\,m3\,\sin(q3 + q4 + q5)} + l_2\,\mathrm{m3\,\sin(q3 + q4)} + \mathrm{lc_2\,m2\,\sin(q3 + q4)}) + .. \\
&\quad \mathrm{qd4}\ (\mathrm{lc_3\,m3\,\sin(q3 + q4 + q5)} + l_2\,\mathrm{m3\,\sin(q3 + q4)} + \mathrm{lc_2\,m2\,\sin(q3 + q4)}) && \text{(D.49)} \\
c_{25} &= \mathrm{lc_3\,m3\,\sin(q3 + q4 + q5)}\ (\mathrm{qd1 + qd3 + qd4 + qd5}) && \text{(D.50)}
\end{aligned}
$$

$$
\begin{aligned}
c_{31} &= \mathrm{q2}\,\mathrm{qd1}\,\left(l_1\,\mathrm{m2}\,\cos(\mathrm{q3}) + l_1\,\mathrm{m3}\,\cos(\mathrm{q3}) + \mathrm{lc}_1\,\mathrm{m1}\,\cos(\mathrm{q3}) + \mathrm{lc}_3\,\mathrm{m3}\,\cos(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) + l_2\,\mathrm{m3}\,\cos(\mathrm{q3}+\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\cos(\mathrm{q3}+\mathrm{q4})\right) + ..\\
&\quad -\mathrm{qd2}\,\left(l_1\,\mathrm{m2}\,\sin(\mathrm{q3}) + l_1\,\mathrm{m3}\,\sin(\mathrm{q3}) + \mathrm{lc}_1\,\mathrm{m1}\,\sin(\mathrm{q3}) + \mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) + l_2\,\mathrm{m3}\,\sin(\mathrm{q3}+\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q3}+\mathrm{q4})\right) + ..\\
&\quad -l_1\,\mathrm{qd4}\,\left(l_2\,\mathrm{m3}\,\sin(\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q4}) + \mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q4}+\mathrm{q5})\right) - \mathrm{lc}_3\,\mathrm{m3}\,\mathrm{qd5}\,\left(l_1\,\sin(\mathrm{q4}+\mathrm{q5}) + l_2\,\sin(\mathrm{q5})\right) && \text{(D.51)}\\
c_{32} &= -\mathrm{qd1}\,\left(l_1\,\mathrm{m2}\,\sin(\mathrm{q3}) + l_1\,\mathrm{m3}\,\sin(\mathrm{q3}) + \mathrm{lc}_1\,\mathrm{m1}\,\sin(\mathrm{q3}) + \mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) + l_2\,\mathrm{m3}\,\sin(\mathrm{q3}+\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q3}+\mathrm{q4})\right) && \text{(D.52)}\\
c_{33} &= -l_1\,\mathrm{qd4}\,\left(l_2\,\mathrm{m3}\,\sin(\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q4}) + \mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q4}+\mathrm{q5})\right) - \mathrm{lc}_3\,\mathrm{m3}\,\mathrm{qd5}\,\left(l_1\,\sin(\mathrm{q4}+\mathrm{q5}) + l_2\,\sin(\mathrm{q5})\right) && \text{(D.53)}\\
c_{34} &= -l_1\,\mathrm{qd1}\,\left(l_2\,\mathrm{m3}\,\sin(\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q4}) + \mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q4}+\mathrm{q5})\right) - \mathrm{lc}_3\,\mathrm{m3}\,\mathrm{qd5}\,\left(l_1\,\sin(\mathrm{q4}+\mathrm{q5}) + l_2\,\sin(\mathrm{q5})\right) + ..\\
&\quad -l_1\,\mathrm{qd3}\,\left(l_2\,\mathrm{m3}\,\sin(\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q4}) + \mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q4}+\mathrm{q5})\right) + ..\\
&\quad -l_1\,\mathrm{qd4}\,\left(l_2\,\mathrm{m3}\,\sin(\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q4}) + \mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q4}+\mathrm{q5})\right) && \text{(D.54)}\\
c_{35} &= -\mathrm{lc}_3\,\mathrm{m3}\,\left(l_1\,\sin(\mathrm{q4}+\mathrm{q5}) + l_2\,\sin(\mathrm{q5})\right)\,\left(\mathrm{qd1}+\mathrm{qd3}+\mathrm{qd4}+\mathrm{qd5}\right) && \text{(D.55)}\\
c_{41} &= \mathrm{qd1}\,\left(\mathrm{q2}\,\left(\mathrm{lc}_3\,\mathrm{m3}\,\cos(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) + l_2\,\mathrm{m3}\,\cos(\mathrm{q3}+\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\cos(\mathrm{q3}+\mathrm{q4})\right) + l_1\,\mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q4}+\mathrm{q5}) + l_1\,l_2\,\mathrm{m3}\,\sin(\mathrm{q4}) + l_1\,\mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q4})\right) + ..\\
&\quad -\mathrm{qd2}\,\left(\mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) + l_2\,\mathrm{m3}\,\sin(\mathrm{q3}+\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q3}+\mathrm{q4})\right) + ..\\
&\quad l_1\,\mathrm{qd3}\,\left(l_2\,\mathrm{m3}\,\sin(\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q4}) + \mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q4}+\mathrm{q5})\right) - l_2\,\mathrm{lc}_3\,\mathrm{m3}\,\mathrm{qd5}\,\sin(\mathrm{q5}) && \text{(D.56)}\\
c_{42} &= -\mathrm{qd1}\,\left(\mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) + l_2\,\mathrm{m3}\,\sin(\mathrm{q3}+\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q3}+\mathrm{q4})\right) && \text{(D.57)}\\
c_{43} &= l_1\,\mathrm{qd1}\,\left(l_2\,\mathrm{m3}\,\sin(\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q4}) + \mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q4}+\mathrm{q5})\right) + l_1\,\mathrm{qd3}\,\left(l_2\,\mathrm{m3}\,\sin(\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q4}) + \mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q4}+\mathrm{q5})\right) + ..\\
&\quad -l_2\,\mathrm{lc}_3\,\mathrm{m3}\,\mathrm{qd5}\,\sin(\mathrm{q5}) && \text{(D.58)}\\
c_{44} &= -l_2\,\mathrm{lc}_3\,\mathrm{m3}\,\mathrm{qd5}\,\sin(\mathrm{q5}) && \text{(D.59)}\\
c_{45} &= -l_2\,\mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q5})\,\left(\mathrm{qd1}+\mathrm{qd3}+\mathrm{qd4}+\mathrm{qd5}\right) && \text{(D.60)}\\
c_{51} &= \mathrm{lc}_3\,\mathrm{m3}\,\left(l_2\,\mathrm{qd1}\,\sin(\mathrm{q5}) - \mathrm{qd2}\,\sin(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) + l_2\,\mathrm{qd3}\,\sin(\mathrm{q5}) + l_2\,\mathrm{qd4}\,\sin((\mathrm{q5}) + \mathrm{q2}\,\mathrm{qd1}\,\cos(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) + ..\right.\\
&\quad \left. l_1\,\mathrm{qd1}\,\sin(\mathrm{q4}+\mathrm{q5}) + l_1\,\mathrm{qd3}\,\sin(\mathrm{q4}+\mathrm{q5})\right) && \text{(D.61)}\\
c_{52} &= -\mathrm{lc}_3\,\mathrm{m3}\,\mathrm{qd1}\,\sin(\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) && \text{(D.62)}\\
c_{53} &= \mathrm{lc}_3\,\mathrm{m3}\,\left(l_2\,\mathrm{qd1}\,\sin(\mathrm{q5}) + l_2\,\mathrm{qd3}\,\sin(\mathrm{q5}) + l_2\,\mathrm{qd4}\,\sin(\mathrm{q5}) + l_1\,\mathrm{qd1}\,\sin(\mathrm{q4}+\mathrm{q5}) + l_1\,\mathrm{qd3}\,\sin(\mathrm{q4}+\mathrm{q5})\right) && \text{(D.63)}\\
c_{54} &= l_2\,\mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q5})\,\left(\mathrm{qd1}+\mathrm{qd3}+\mathrm{qd4}\right) && \text{(D.64)}\\
c_{55} &= 0 && \text{(D.65)}
\end{aligned}
$$

D.5 C(q)

Matrix G appears in the term representing Coriolis and centripetal effects

$$
\begin{aligned}
g1 &= g\,((\mathrm{m1}\,\mathrm{q2}\,\cos(\mathrm{q1}) + \mathrm{m2}\,\mathrm{q2}\,\cos(\mathrm{q1}) + \mathrm{m3}\,\mathrm{q2}\,\cos(\mathrm{q1}) - l_2\,\mathrm{m3}\,\sin(\mathrm{q1}+\mathrm{q3}+\mathrm{q4}) - \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q1}+\mathrm{q3}+\mathrm{q4}) + ..\\
&\quad -\mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q1}+\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) - l_1\,\mathrm{m2}\,\sin(\mathrm{q1}+\mathrm{q3}) - l_1\,\mathrm{m3}\,\sin(\mathrm{q1}+\mathrm{q3}) - \mathrm{lc}_1\,\mathrm{m1}\,\sin(\mathrm{q1}+\mathrm{q3})) && \text{(D.66)}\\
g2 &= g\,(\mathrm{m1}\,\sin(\mathrm{q1}) + \mathrm{m2}\,\sin(\mathrm{q1}) + \mathrm{m3}\,\sin(\mathrm{q1})) && \text{(D.67)}\\
g3 &= -g\,((l_2\,\mathrm{m3}\,\sin(\mathrm{q1}+\mathrm{q3}+\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q1}+\mathrm{q3}+\mathrm{q4}) + \mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q1}+\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) + l_1\,\mathrm{m2}\,\sin(\mathrm{q1}+\mathrm{q3}) + ..\\
&\quad l_1\,\mathrm{m3}\,\sin(\mathrm{q1}+\mathrm{q3}) + \mathrm{lc}_1\,\mathrm{m1}\,\sin(\mathrm{q1}+\mathrm{q3}) && \text{(D.68)}\\
g4 &= -g\,(l_2\,\mathrm{m3}\,\sin(\mathrm{q1}+\mathrm{q3}+\mathrm{q4}) + \mathrm{lc}_2\,\mathrm{m2}\,\sin(\mathrm{q1}+\mathrm{q3}+\mathrm{q4}) + \mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q1}+\mathrm{q3}+\mathrm{q4}+\mathrm{q5})) && \text{(D.69)}\\
g5 &= -g\,\mathrm{lc}_3\,\mathrm{m3}\,\sin(\mathrm{q1}+\mathrm{q3}+\mathrm{q4}+\mathrm{q5}) && \text{(D.70)}
\end{aligned}
$$

Appendix E

Contact model parameters

The table **??** contains the properties of the contacting bodies which were used in the computation of the contact dynamics.

Material Material	Young's modulus, E [N/m]	Poisson's ratio, υ [-]	Radius R [m]	Coefficient of restitution, er [-]	Coefficient of friction, μ [-]
Rubber	5.0e7	0.50	7.5e-3	-	-
Concrete	3.0e10	0.20	∞	-	-
Rubber/Concrete	-	-	-	0.80	1

Table E.1: Properties of the contacting bodies used in the computation of the contact dynamics

Appendix F

Simulink model

The systems and subsystems which build up the robot system in Simulink will be shown here.

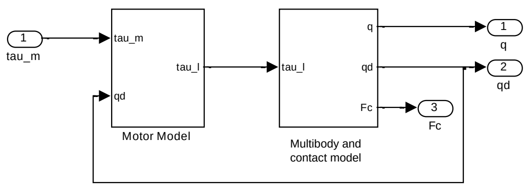


Figure F.1: Simulink model of robot system

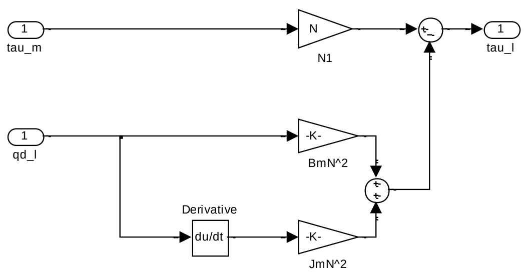


Figure F.2: Motor model subsystem

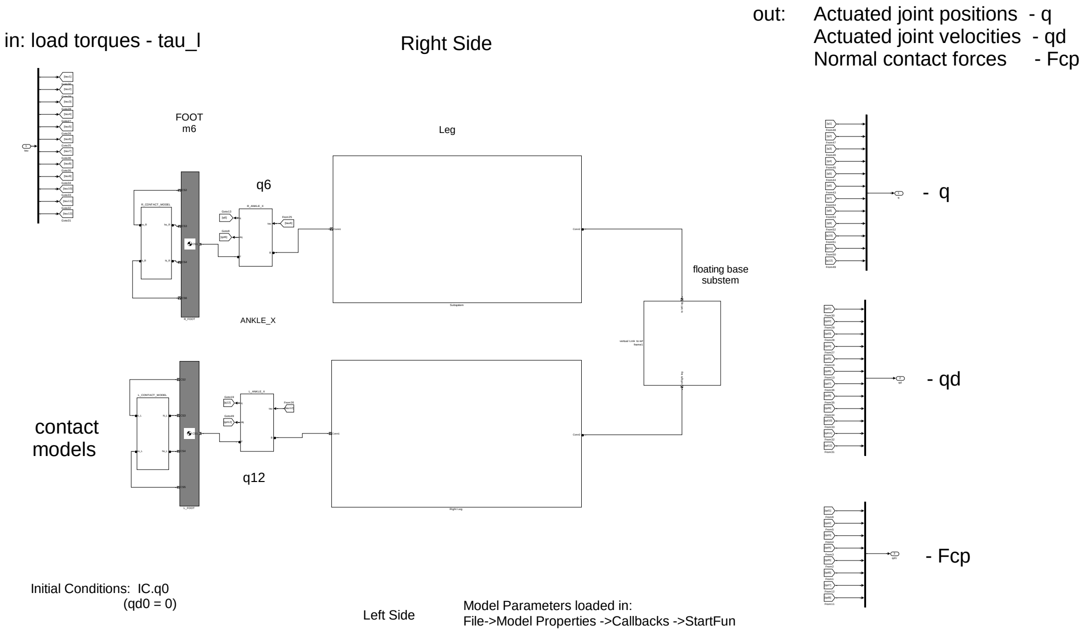


Figure F.3: SimMechanics Multibody and contact model subsystem

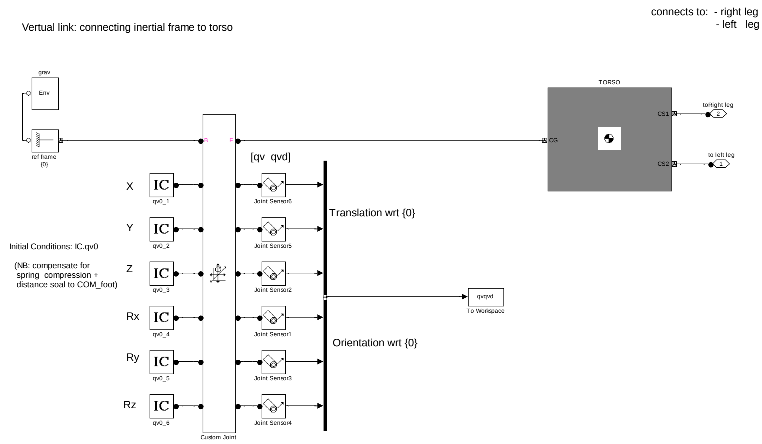


Figure F.4: SimMechanics model of the torso connected to the inertial frame

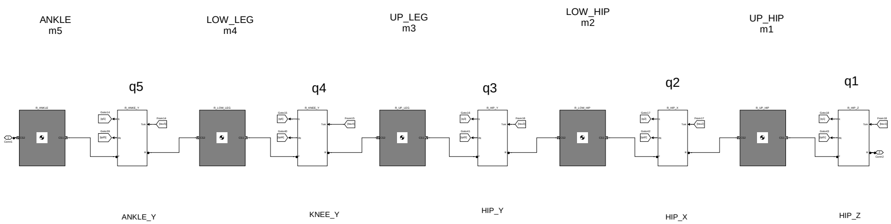


Figure F.5: SimMechanics model of the right leg, which was defined as the masses and joint elements from the Hip Z joint down to the ankle mass

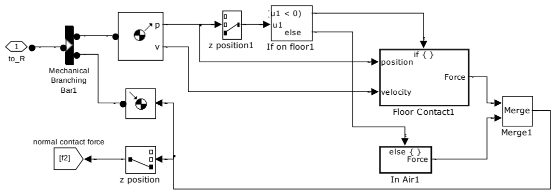


Figure F.6: Contact model.

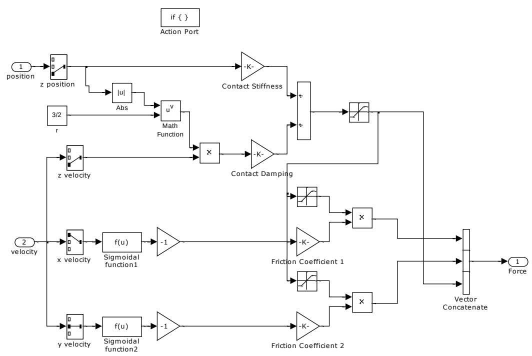


Figure F.7: Floor contact subsystem.

Appendix G

Measurements

To verify the model, the simulated and measured robot motor torques needed to be compared, given a particular reference trajectory. There were some issues generating the joint reference trajectories and measuring the joint torques which are described here.

G.1 Reference Trajectory generation

The algorithm which computed the joint reference trajectories is a function of 4 variables, namely step distance and size as well as the so called swing and stands phase time. Swing time refers to the time span allocated for the motion that the biped is supported by one foot. The stance time refers to the time span allocated to the motion when both feet support the biped. An initial attempt at reconstructing the trajectory generating algorithm implemented on Tulip, using the Matlab algorithm with this same variables [1] shows significant deviation to the trajectory which was computed on Tulip, as illustrated in figure G.1.

To investigate where the difference was coming from, forward kinematics of the biped were computed using the measured reference trajectory. Subsequent plots revealed that, the trajectory computed on the biped moved the biped as if it were going up stairs. This led to the conclusion that the gait algorithm implemented on Tulip was a tuned version of the original gait to help compensate for steady state errors due gravity forces. To make a meaningful comparison between model and robot the reference trajectories needed to be the same therefore the measured reference trajectory was interpolated and used in the simulation.

G.2 Torque measurements

There were no torque sensors on the motors so these were derived from a control signal to the motor, namely the signal labeled 'pwm'in figure G.2. As the figure shows, these are integer values between ±1024, which relates to $\pm\tau_{max}$. From this the motor torque, τ_m can be computed as:

$$\tau_m = \text{pwm}\frac{I_{max}}{1024}K_m \tag{G.1}$$

As the messa board source code was not available, to check that the single beging measured was infact the 'pwm' as represented in figure G.2, the values for the right side of the robot (see figure XXX) were checked to make sure that they were indead bounded by ±1024, as shown in figure G.3.

Figure G.3 shows that the control signal approaches the bounds for the ankle X joint, indicating large position errors at this joint. This makes physical sense as during the execution of the

[1] The values used were: 0.35m ,0.1m, 2sec and 3sec for the step distance, step size, swing time and stands time

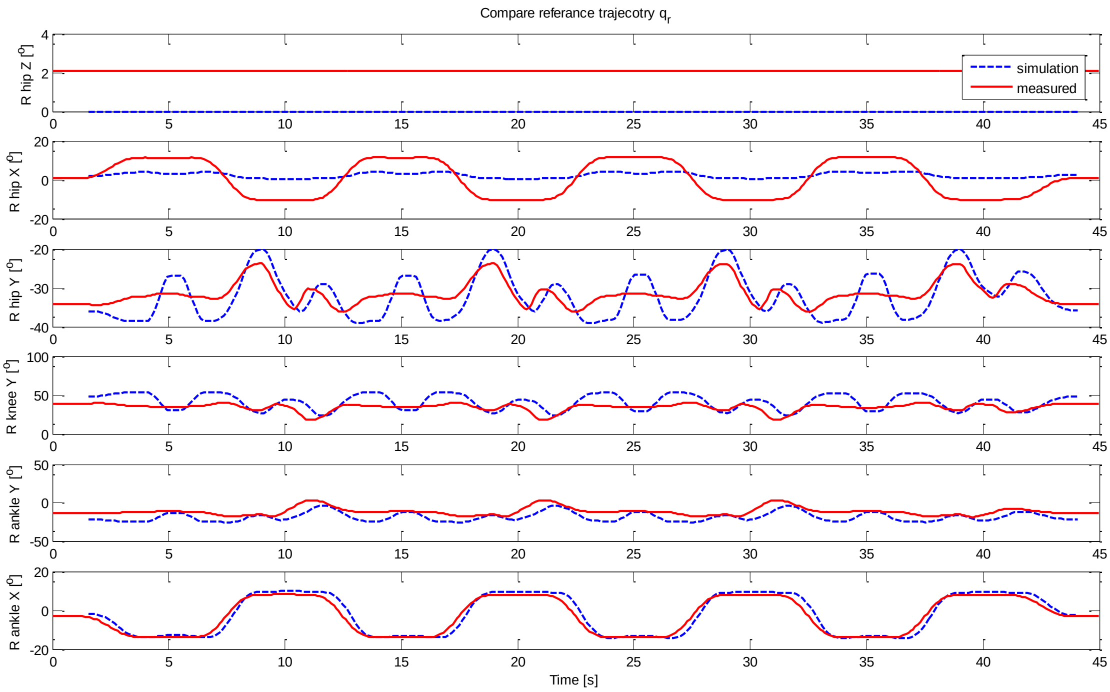


Figure G.1: Reference trajectories computed by the algorithm implemented on tulip (in C++) and those computed by a Matlab algorithm

experiments the biped was given assistance to keep its balance during the walking gait. This assistance was provided by applying counter forces to the torso thus effectively applying large moments about this joint.

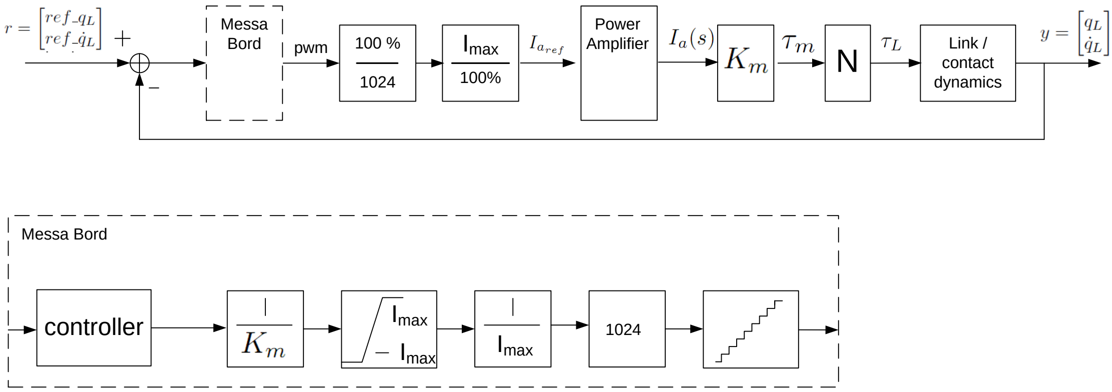


Figure G.2: Block diagram of the system, showing signal flow though the messa bords

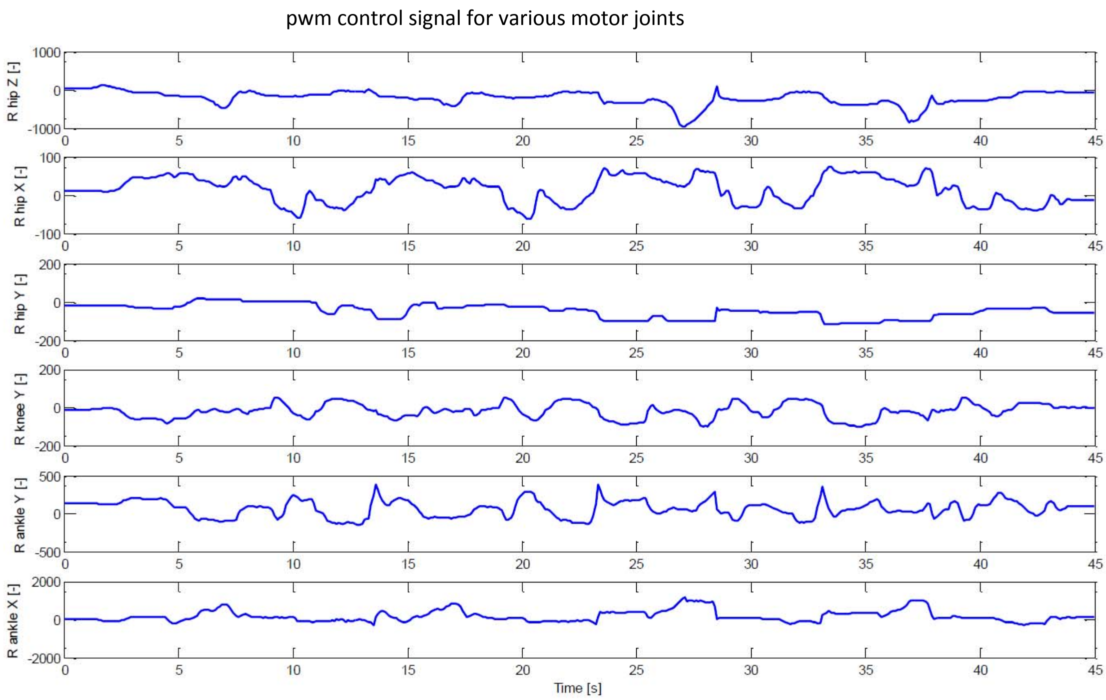


Figure G.3: Measured 'pwm' values for various motors, where the motor torque, $\tau_m = \text{pwm}\frac{I_{max}}{1024}K_m$