MITSUBISHI ELECTRIC
Changes for the Better

MITSUBISHI ELECTRIC RESEARCH LABORATORIES
Cambridge, Massachusetts

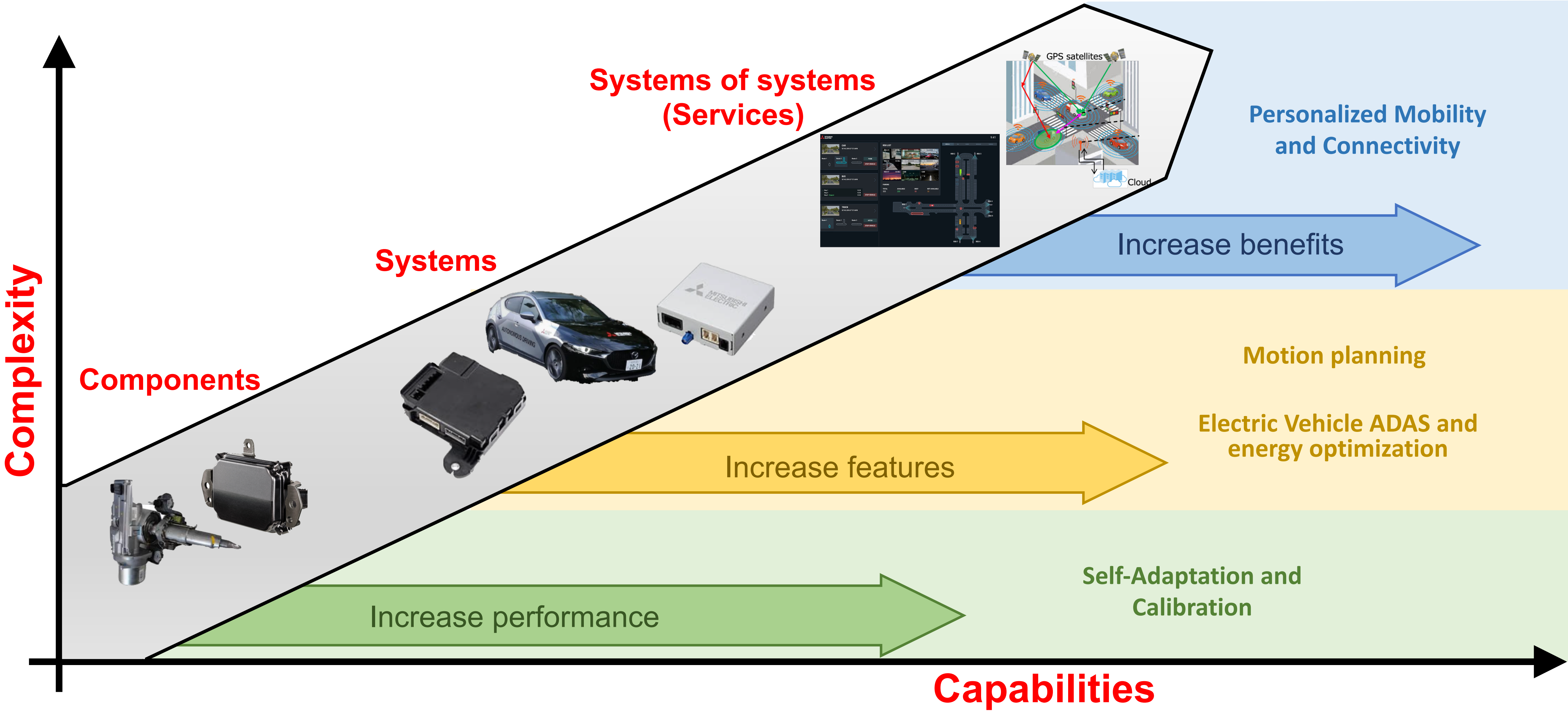
Self-Adaptation and Automated Control Design

Marcel Menner

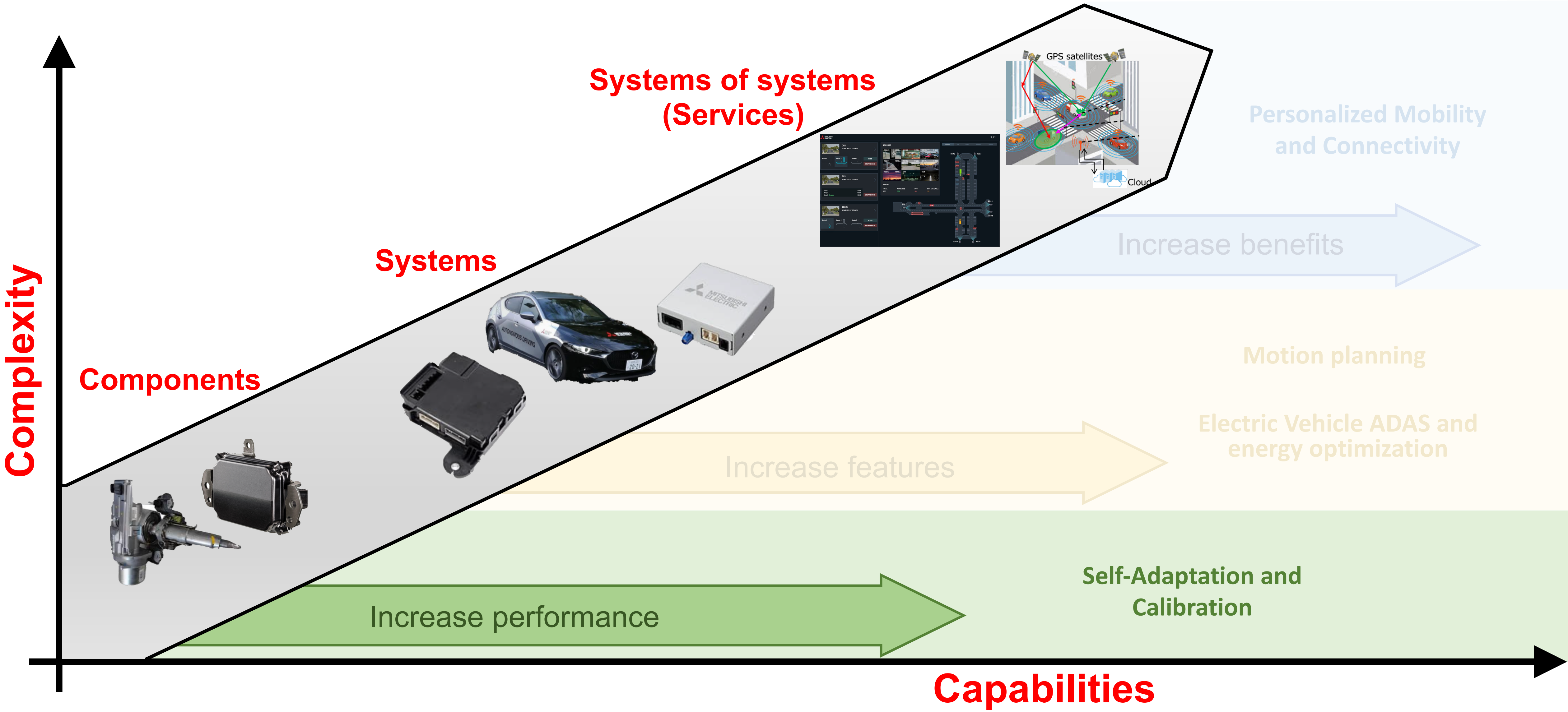
(menner@merl.com)

Contributions by:
MERL: K. Berntorp, R. Quirynen, S. Di Cairano

Overview of Research Directions



Overview of Research Directions

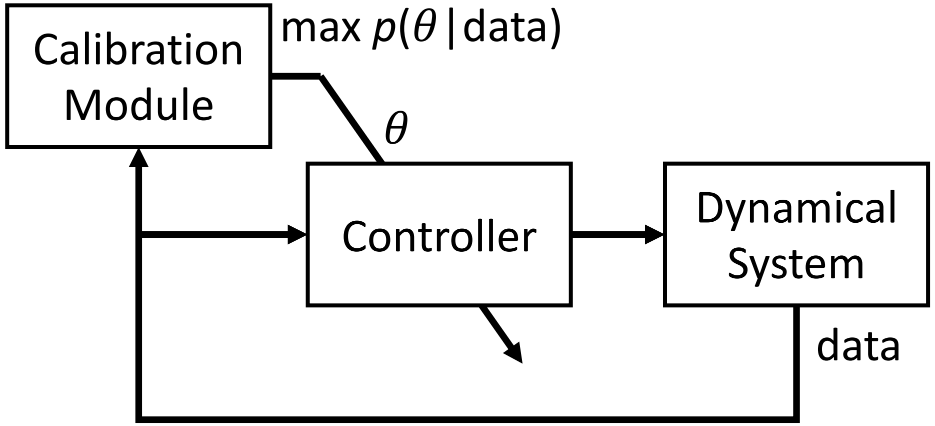


Automated Controller Calibration

Calibrate Controller to Satisfy Specifications for System
Recursive controller calibration algorithm

Main idea: Kalman filter estimating controller parameters

➞ Controller parameters can be: MPC cost function (weights), PID gains, filter coefficients, Neural Network weights, …



Control law with parameters θ

$$u_k = \kappa_\theta(x_k, z_k)$$

θ Control parameter

u_k Control input

x_k Measured states of dynamical system

z_k Internal controller states, e.g., integrator, state estimate

Automated Controller Calibration

Control law with parameters θ $\quad u_k = \kappa_\theta(x_k, z_k)$

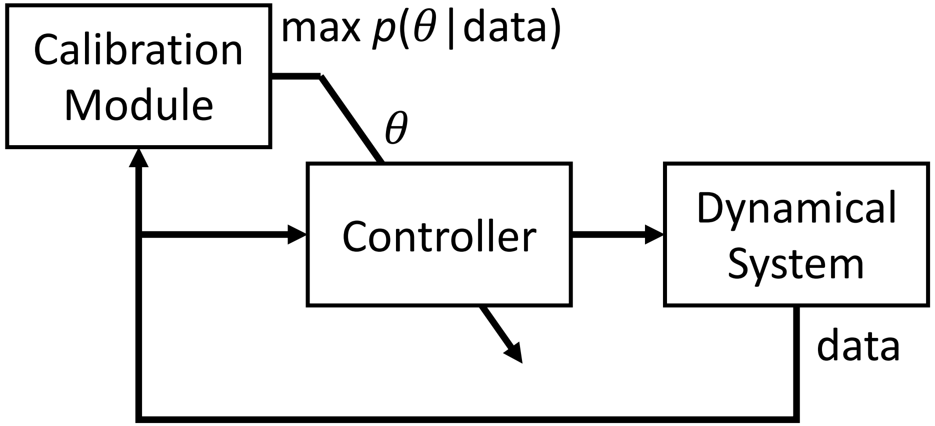
Model mismatch

Model-based calibration $\quad x_{k+1} = f(x_k, u_k) + w_k$

Specification function $\quad h(\theta) = r(x_{k-N:k}, u_{k-N:k})$

Desired values $\quad h_{\mathrm{ref},k}$

Training objective $\quad \left\| h_{\mathrm{ref},k} - h(\theta) \right\|_{C_h^{-1}}$



Objective: Find control parameter update law

$$\theta_{k+1} = \theta_k + \Delta\theta_k$$

Control parameter adaptation as estimation problem

with priors $\theta_{k+1}^{\mathrm{prior}} \sim N(\theta_k, C_\theta)$ and $h_{\mathrm{ref},k} \sim N(h(\theta_k), C_h)$

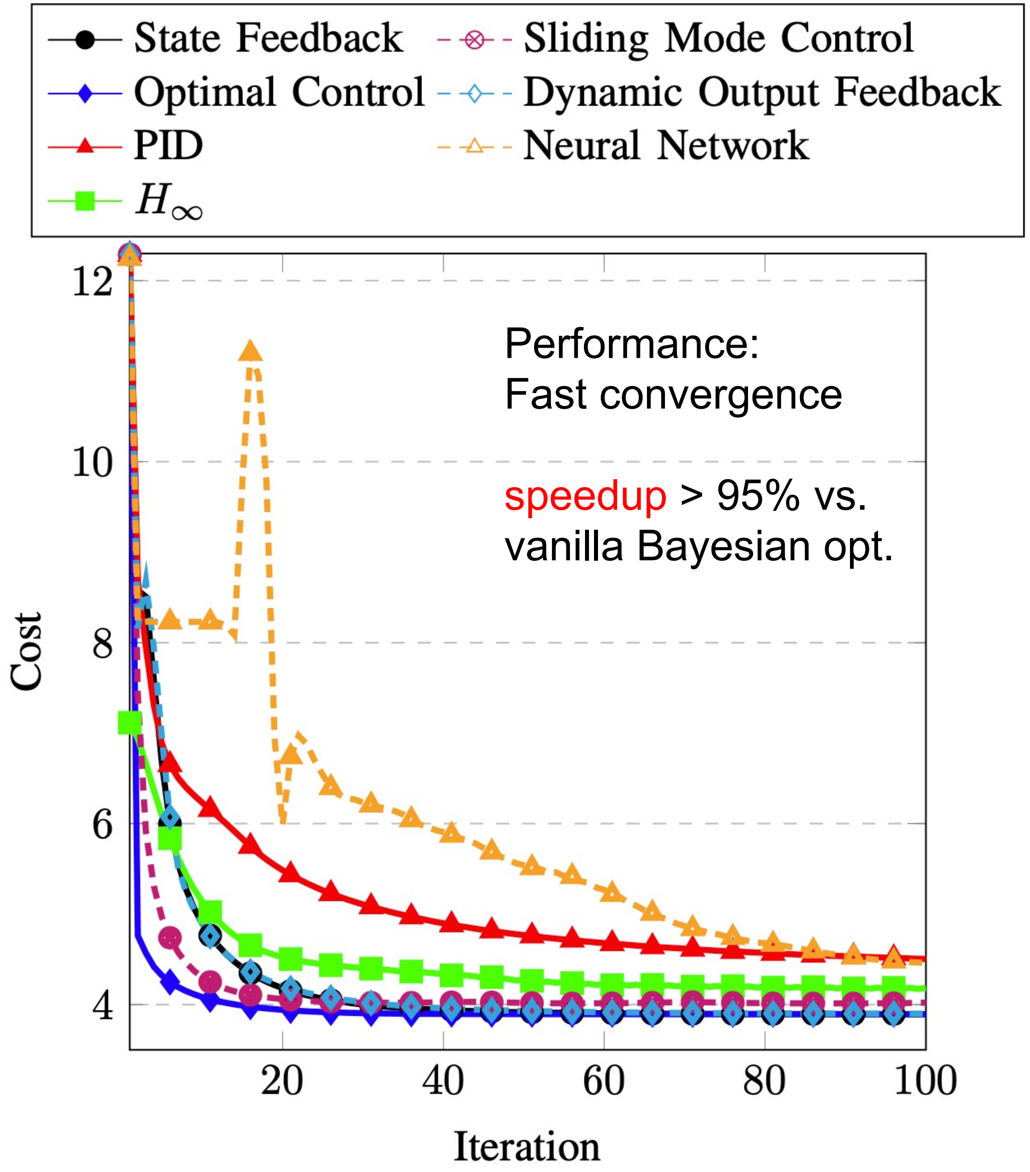
Control parameters from posterior distribution

$$p\left(\theta_{k+1} | \theta_k, \theta_{k-1}, \dots, \theta_0, h_{\mathrm{ref},k}, \dots, h_{\mathrm{ref},0}\right)$$
$$= \prod_{i=0}^{k} p\left(\theta_{i+1} | \theta_i, h_{\mathrm{ref},i}\right) p(\theta_0)$$

Automated Controller Calibration

Calibration applied to different controllers

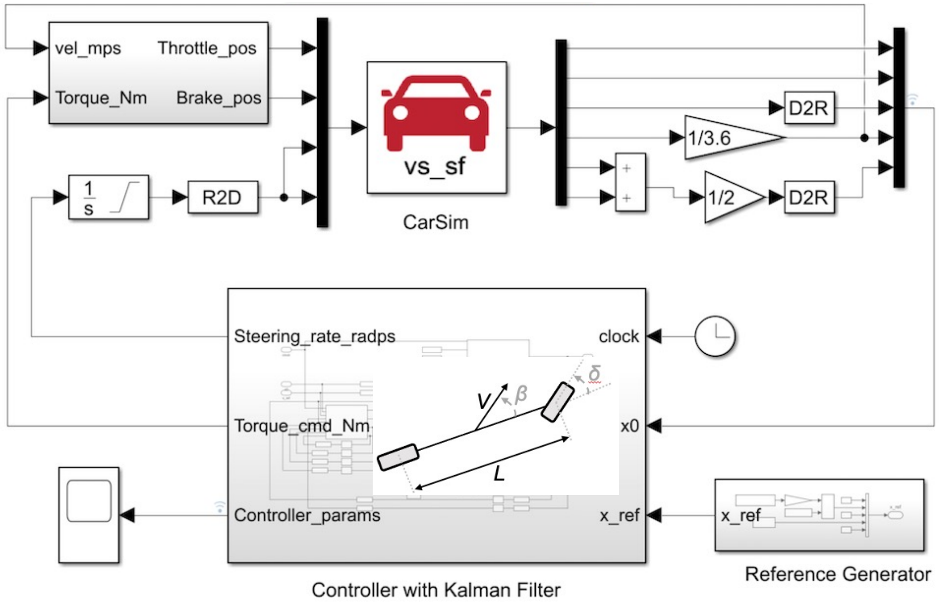
- State feedback: Gains
- Optimal control: Cost function weights
- PID: Gains
- H_∞: Filter coefficients of pre- and post-compensator
- Sliding mode controller: Gains and sliding surface
- Dynamic output feedback: Feedback gains and Luenberger observer gains
- Neural Network: Weights

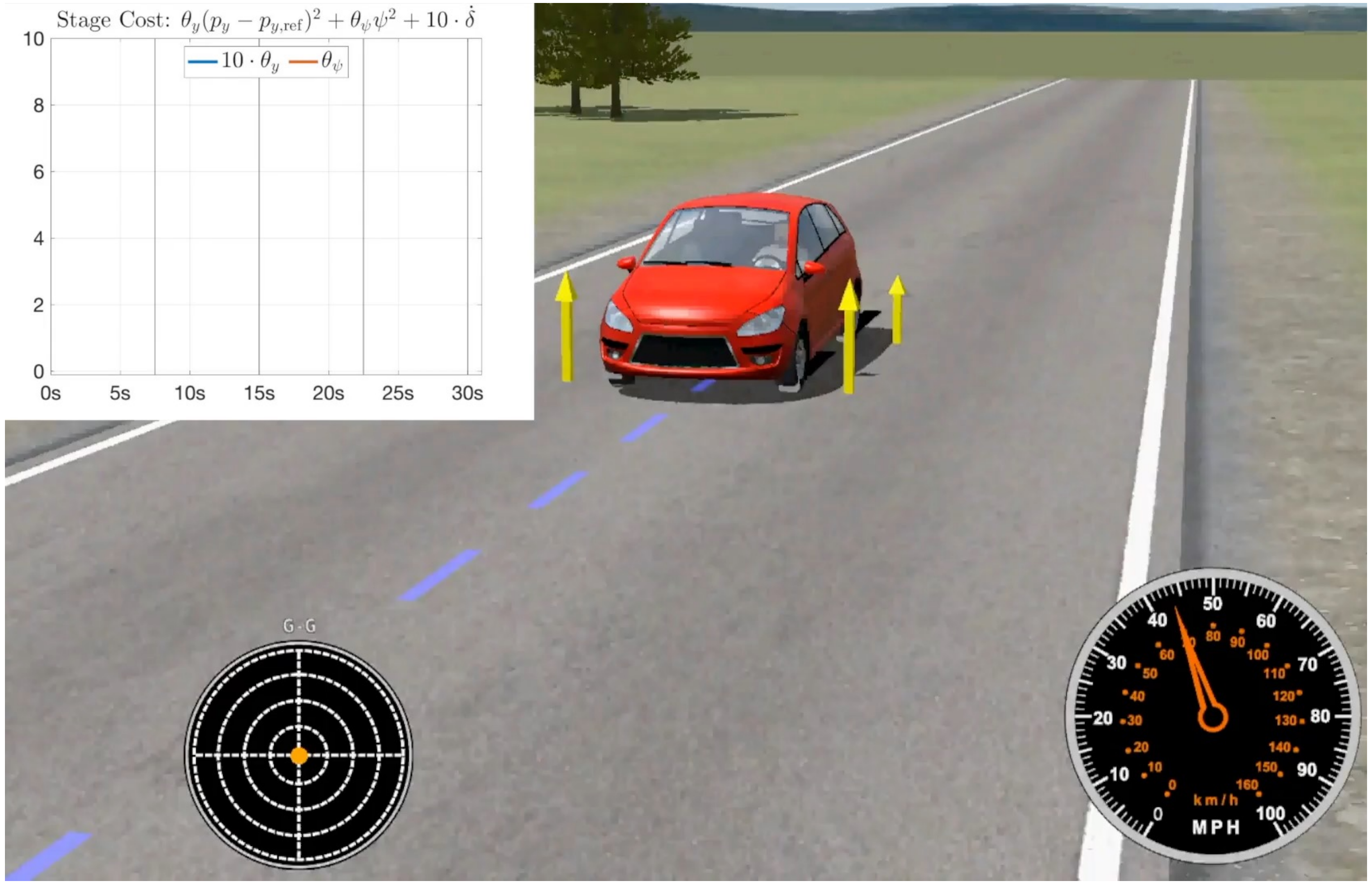


Automated Controller Calibration for Vehicle Control

Kalman filter estimates parameters
Calibration driven by specifications

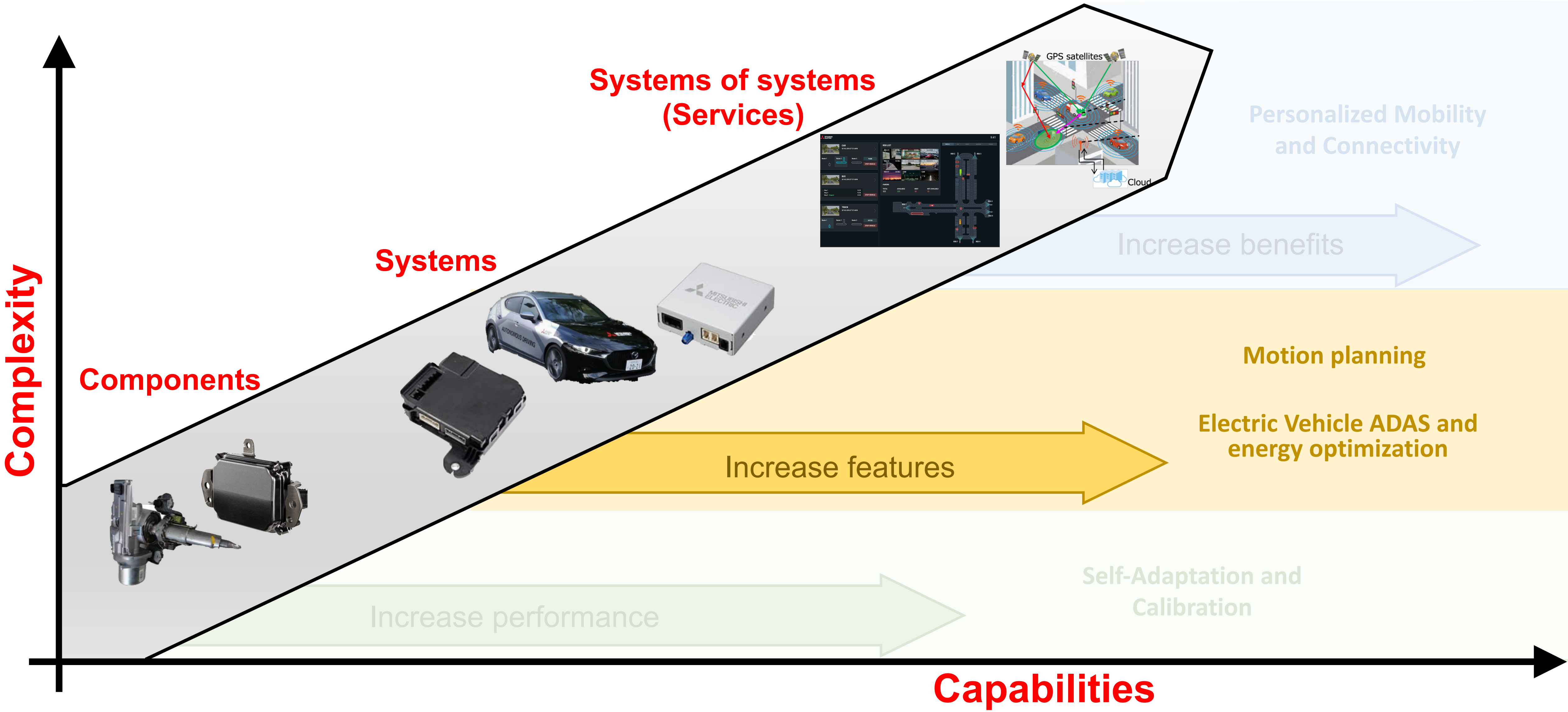
- Model-based approach
  → Little data needed
- Recursive implementation
  → Low hardware requirements





CarSim – Lane Change Controller

Overview of Research Directions



Kernel Regression for Energy-Optimal Control of EVs

Degrees of freedom for controlling battery electric vehicles

i. Transmission gears

ii. Torque-split ratio between motors

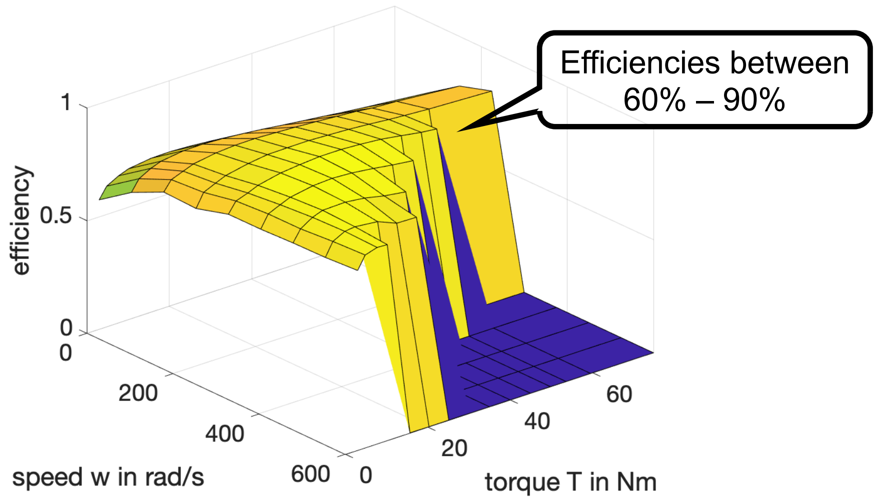
iii. Velocity profile, e.g., for adaptive cruise control or autonomous driving

Challenges

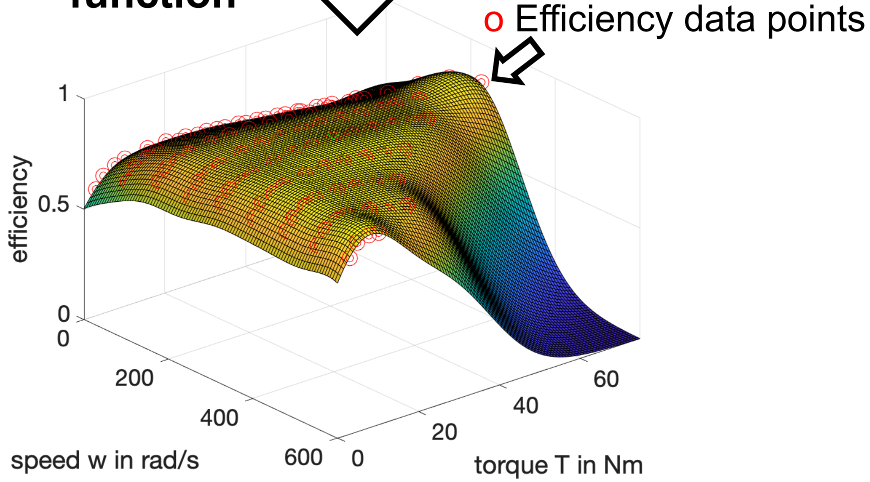
- Fast control algorithms needed
- Mathematical model (cost function) must have flexible shape

Approach for real-time efficiency optimization

- Learn model suited for numerical optimization using kernel regression
- Combine motor losses (from tabulated data of motor efficiency) with driving losses (from Newtonian mechanics)

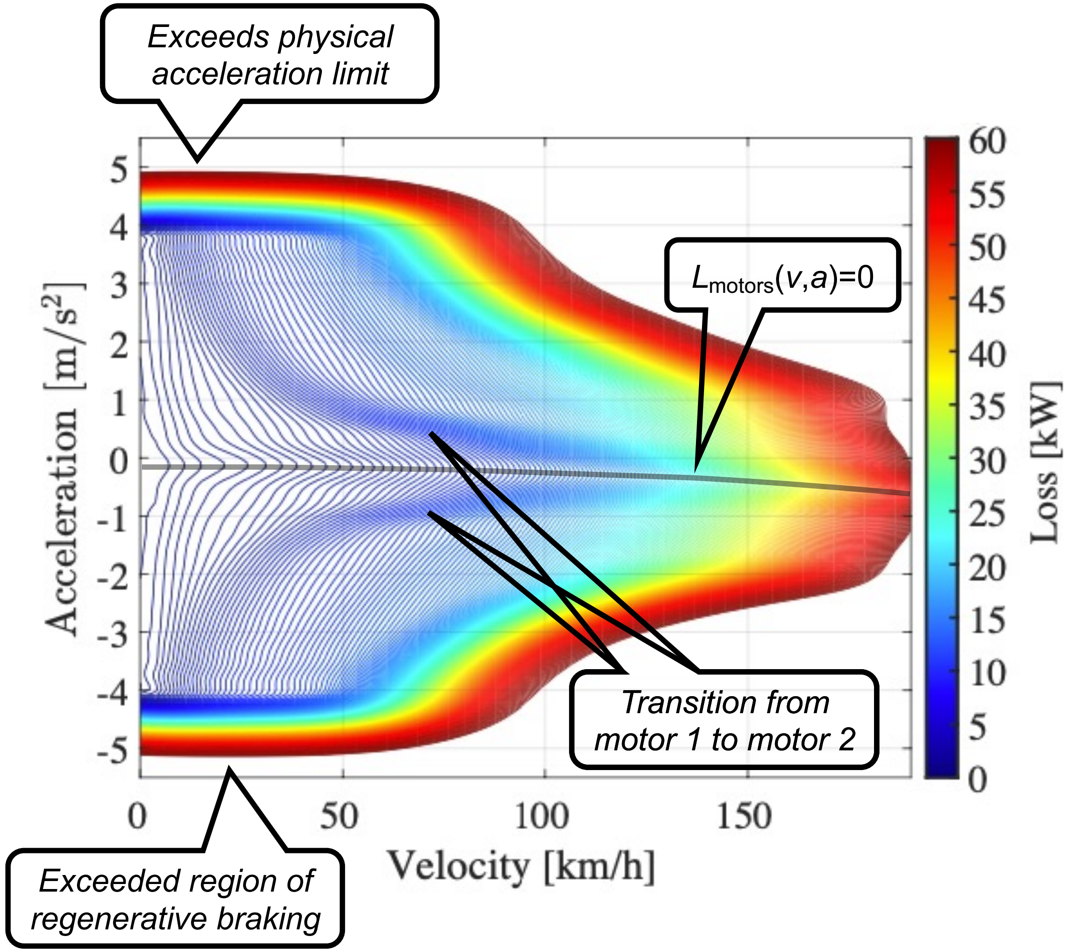


Fit continuous function



Kernel Regression for Energy-Optimal Control of EVs

Pseudo-Convex Cost Function for Velocity Profile Optimization
Combined minimization of driving losses and motor losses



EV loss function $L_{EV}(v,a)$

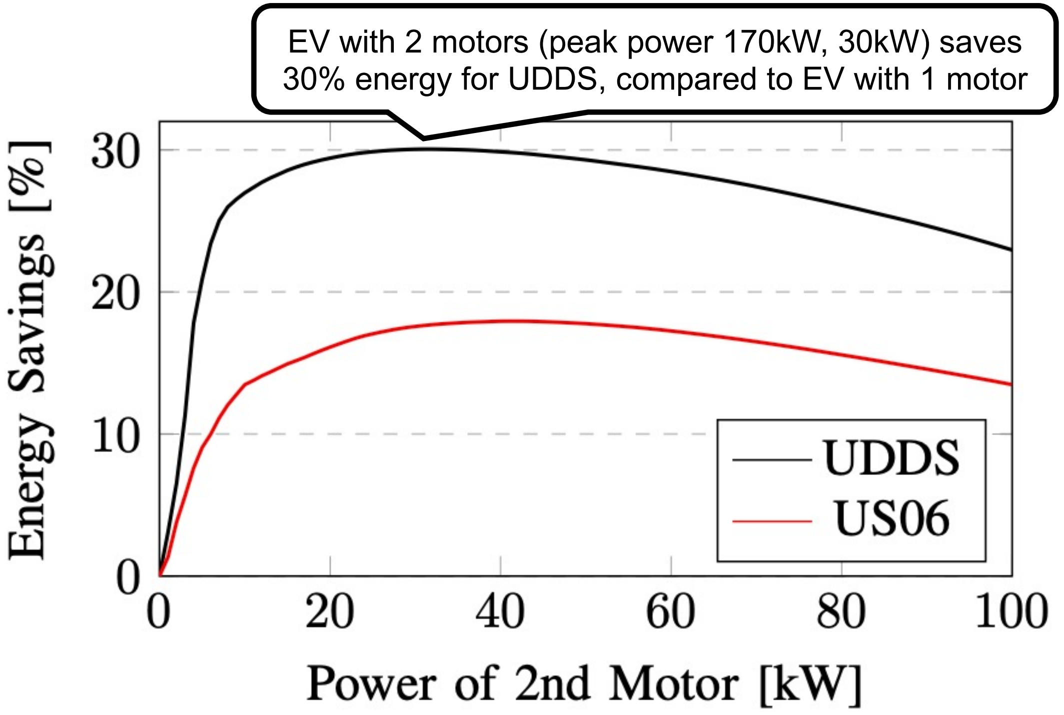
$L_{EV}(v,a) = L_{driving}(v) + L_{motors}(v,a)$

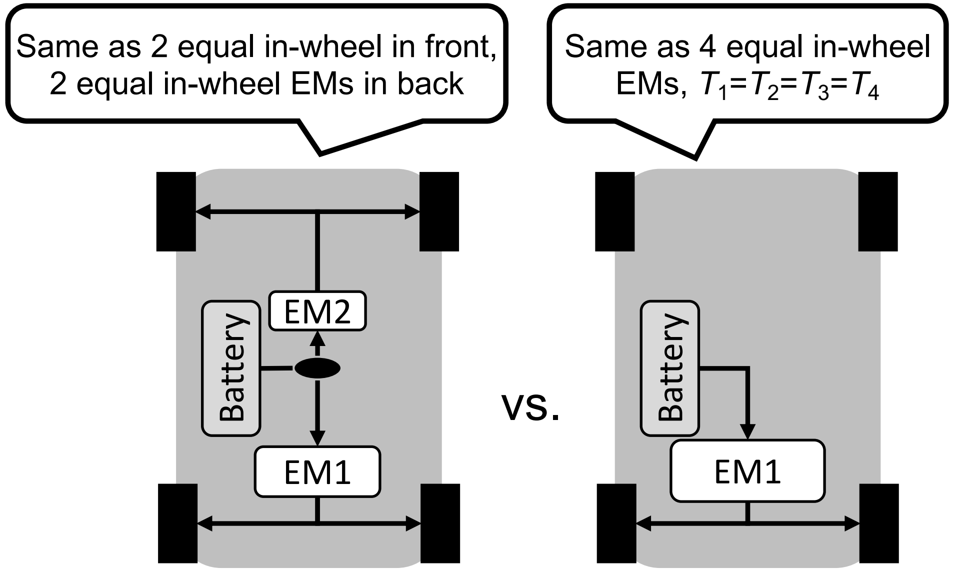
$L_{driving}(v) = v(F_{drag}(v) + F_{roll}(v))$

Design (Motor Size) Optimization

Fixed Vehicle Torque/Speed Demand from Urban Dyno Driving Schedule (UDDS) and US06

Motor 1 with max. power P1 = 200kW – P2
Motor 2 with max. power P2
➞ P2 = 0 implies EV with 1 motor





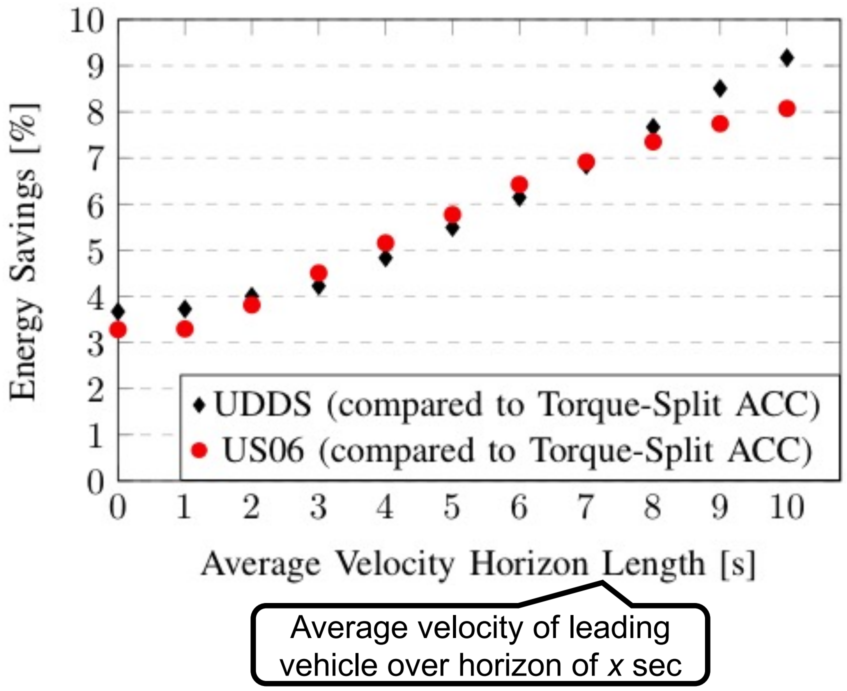
Torque-split Control (fixed gear)

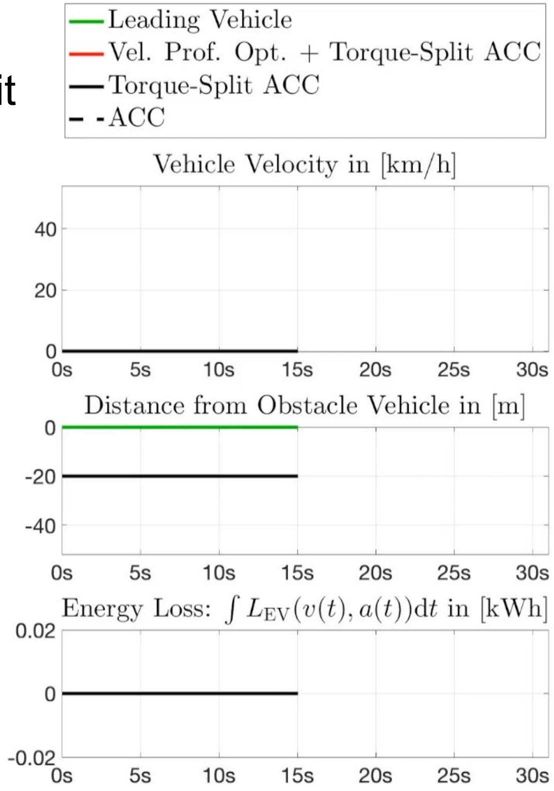
$$T_{vehicle} = T_{EM1} + T_{EM2}$$

Energy Savings of Adaptive Cruise Control (ACC)

Savings of 2-motor (170kW, 30kW)
compared to 1-motor (200kW)

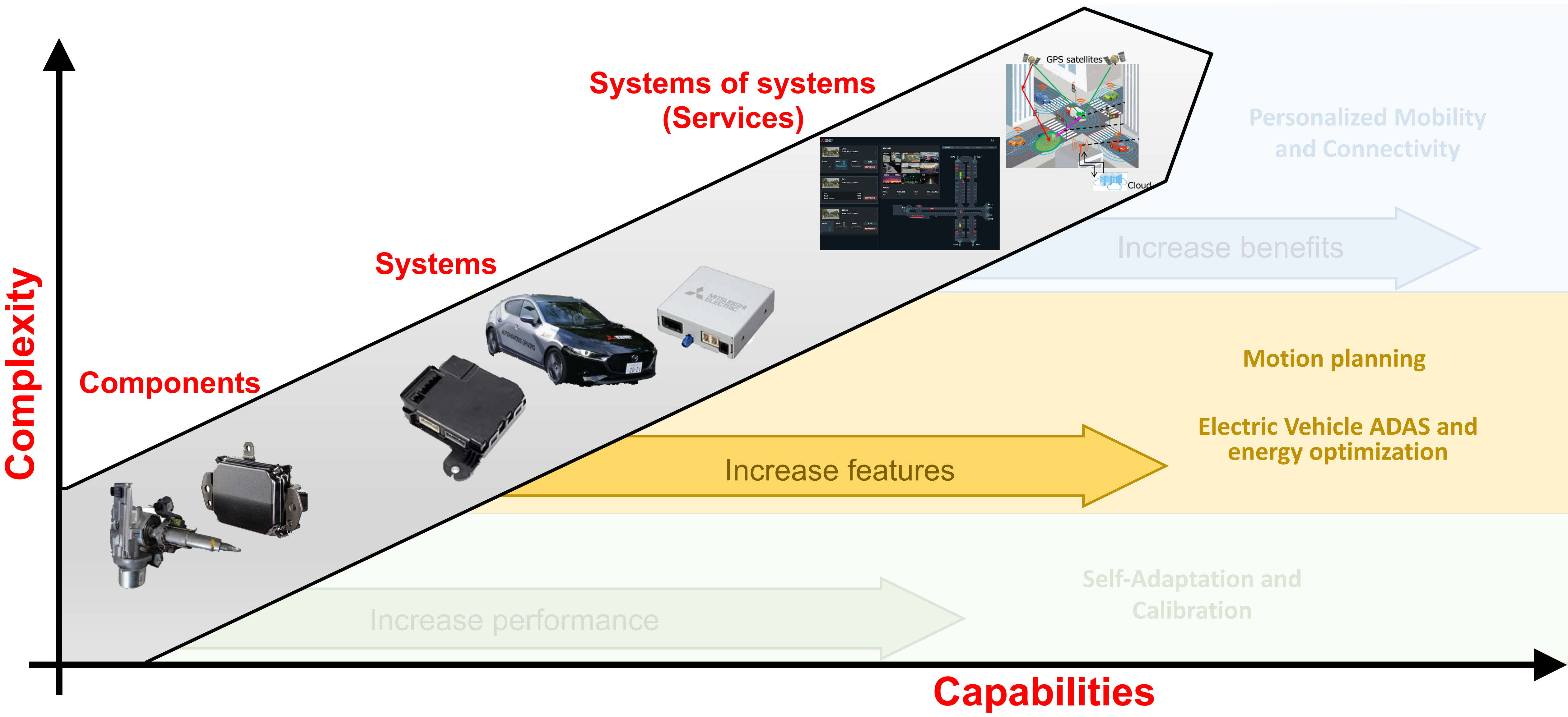
- 30%/18% for UDDS/US06 with torque split
- Additional 3–10% with velocity profile optimization, e.g., for ACC





Energy-Optimized Adaptive Cruise Control (ACC)

Overview of Research Directions

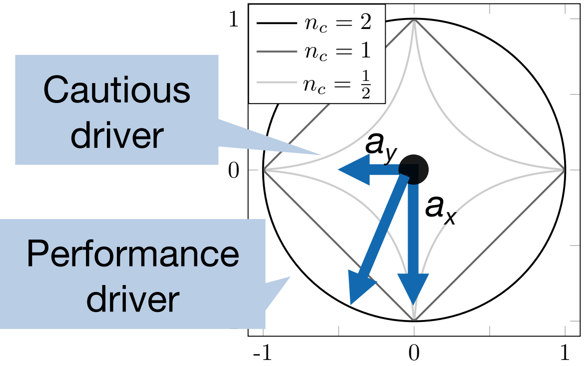


Human-Adaptive Motion Planning

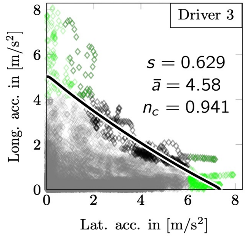
Personalizing Driving Experience of Autonomous Vehicle

Model-based Learning: Keep Safety Guarantees, Learn Performance Parameters

- Fixed: Motion model, driving limits, computational structure
- Parametrized: Driving objectives, relative importance of objectives

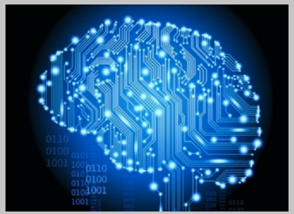


Personal feeling of comfort



Driving Data

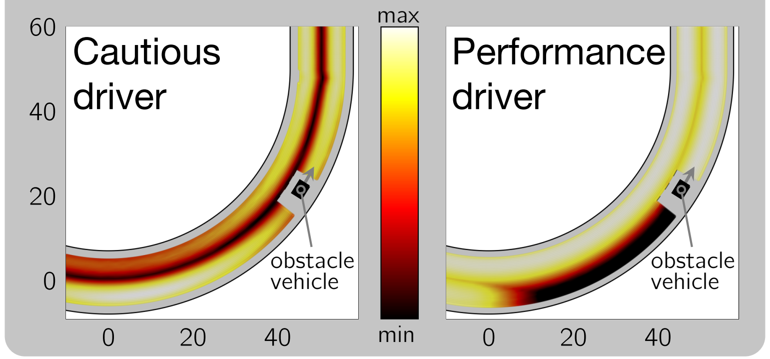
Learning Algorithm (offline calibration)



$$\max_{\theta} \quad p(\theta|\text{data})$$
$$\text{s.t.} \quad \theta \in C_{\theta}$$

Parameters

Motion Planner



Motion Plan

Human-Adaptive Motion Planning

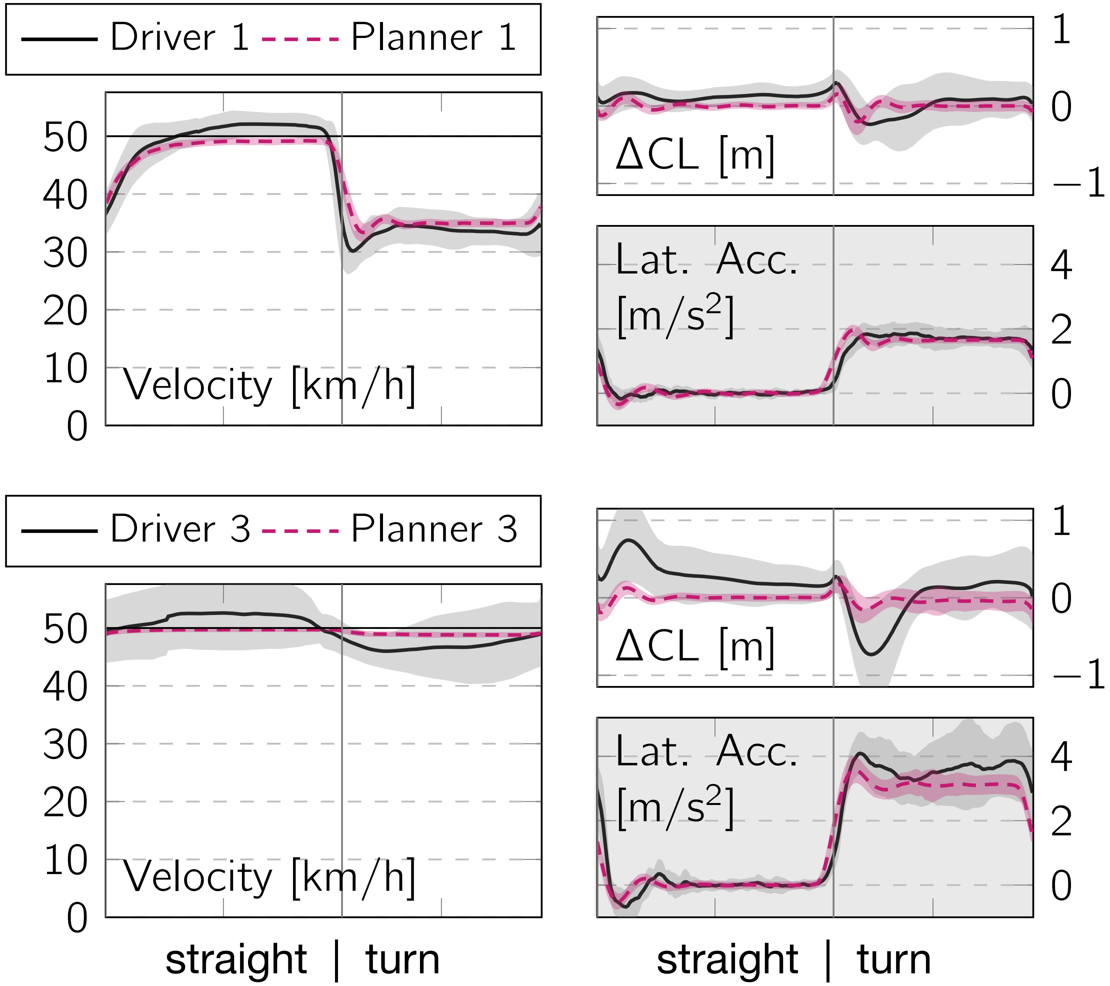
Personalizing Driving Experience of Autonomous Vehicle

Parametrized driving objectives
→ similarities of planner & driver

- Lateral accelerations and velocities of planners match drivers

Fixed model-based algorithm
→ safety properties

- Planners avoid exceeding speed limit (drivers often exceeded speed limit)
- Planners track centerline more closely than drivers



Human-Adaptive Motion Planning

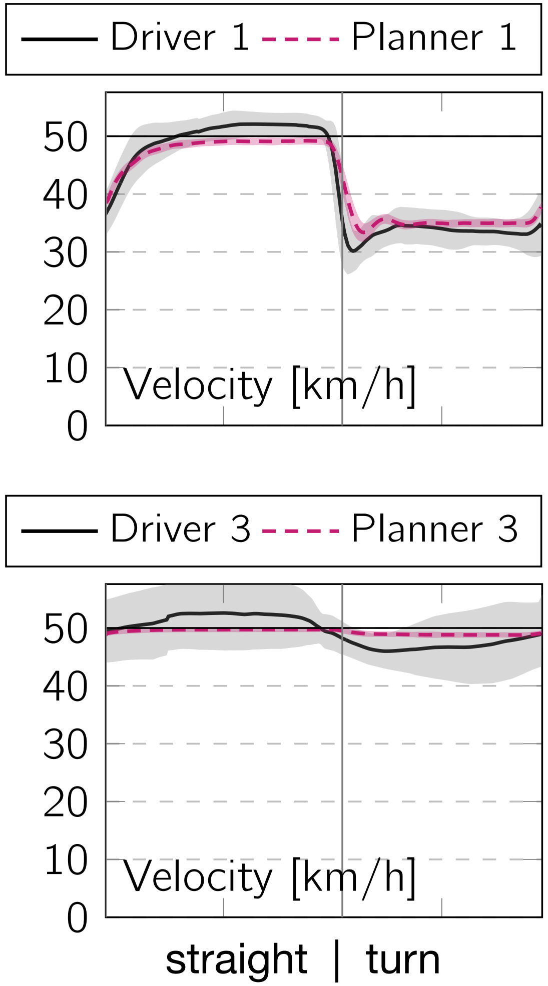
Personalizing Driving Experience of Autonomous Vehicle

Parametrized driving objectives
➞ similarities of planner & driver

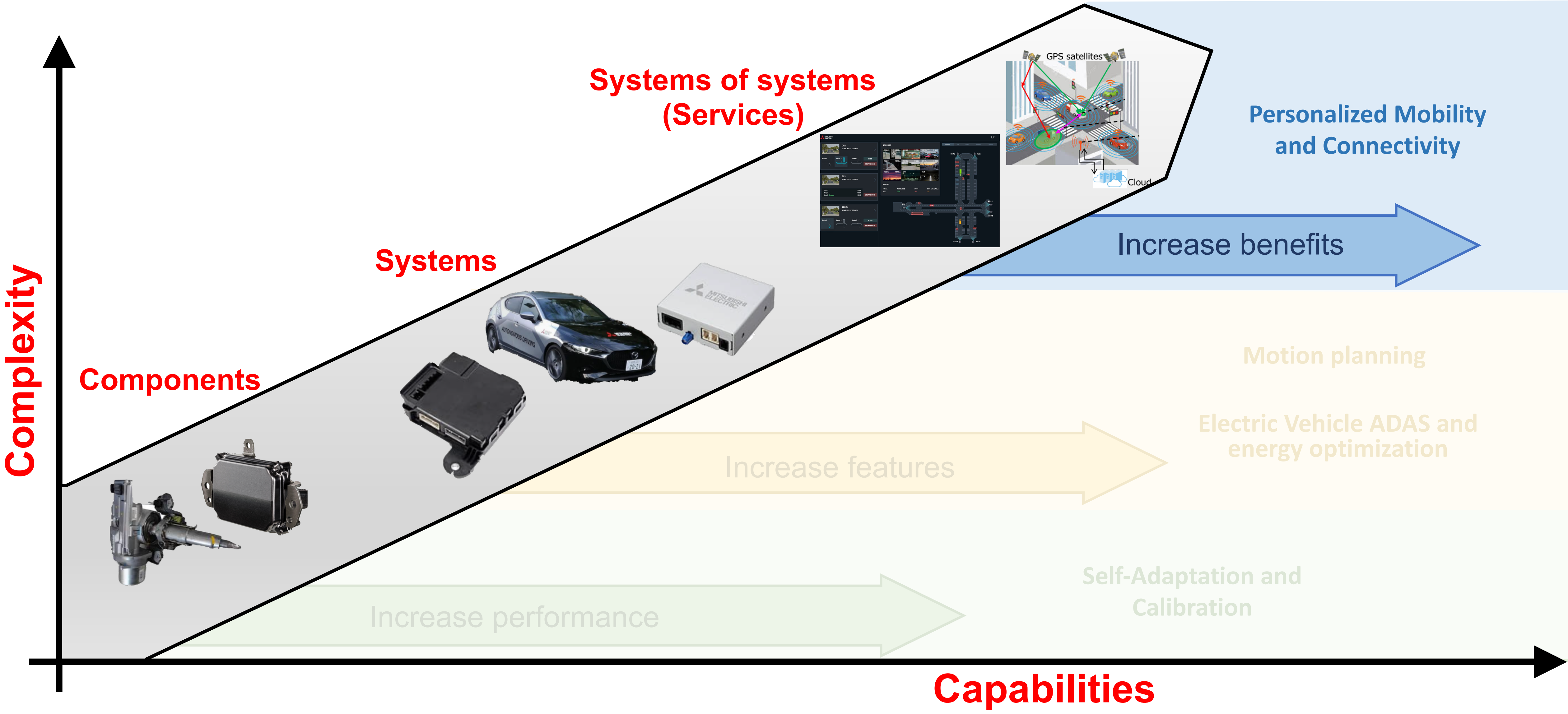
- Lateral accelerations and velocities of planners match drivers

Fixed model-based algorithm
➞ safety properties

- Planners avoid exceeding speed limit (drivers often exceeded speed limit)
- Planners track centerline more closely than drivers



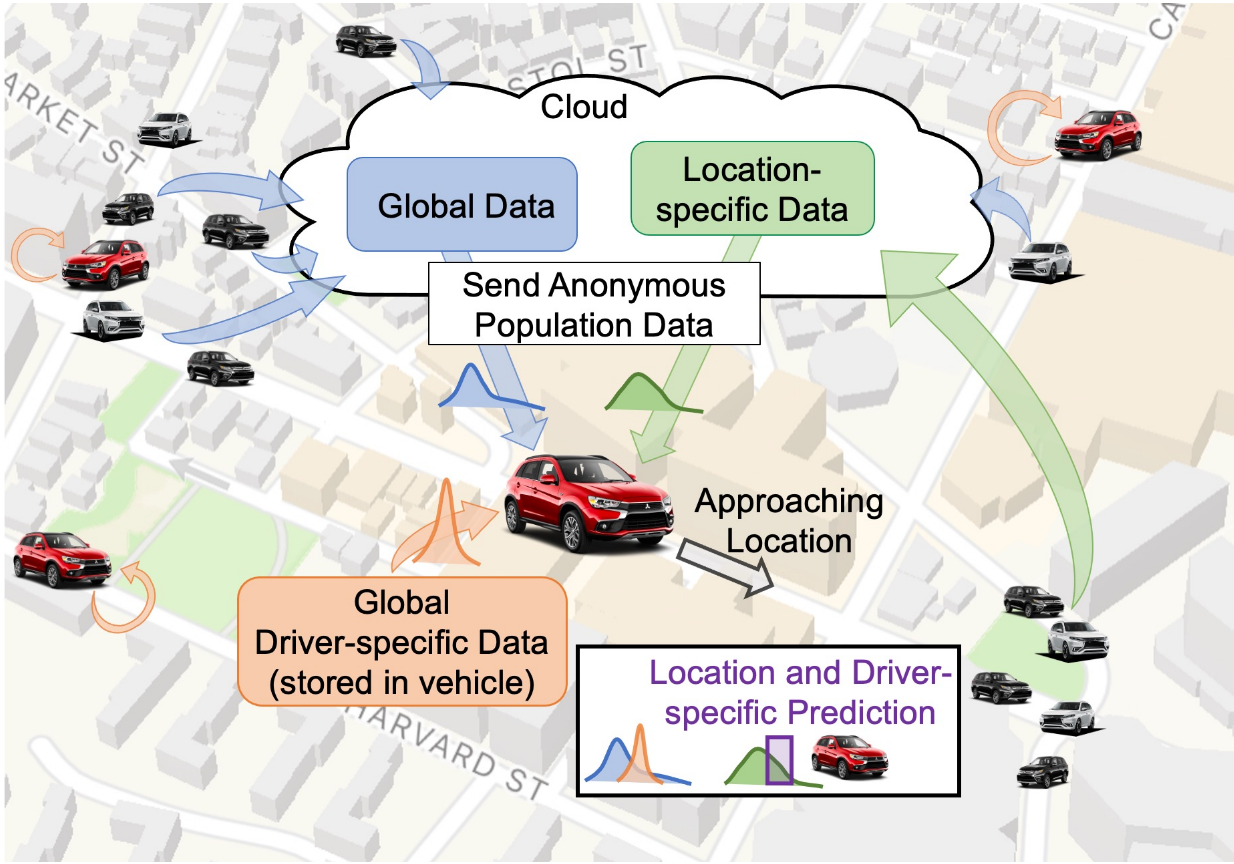
Overview of Research Directions



Vehicle Calibration from Population Data (Crowdsourcing)

Adapting ADAS using Vehicle Data and Crowdsourced Data

- Vehicle data for driver-specific adaptation
- Crowdsourced data for environment/location/time-specific adaptation ➞ driving scenarios that impact all drivers



Fundamental Idea

Exploit Population Data and Vehicle Data

- Use data set for population in all locations *e.g., stored in cloud (unlabeled/anonymous)*
- Use data set for population in specific location *e.g., stored in cloud (unlabeled/anonymous)*
- Use data set for driver in all locations *stored only in vehicle (labeled)*
- Predict behavior for driver in specific location and use for adapting ADAS

Prediction using Empirical Cumulative Density Functions

Prediction-making using percentiles and ranking assumption

For global data

Compare rank of driver, $\tilde{p}$, w.r.t. population, $\tilde{p}_X$

$$\tilde{x}_{X|D} = F_{X|D}^{-1}(\tilde{p})$$
$$\tilde{p}_X = F_X\left(\tilde{x}_{X|D}\right)$$

For specific location

Compare rank of driver, $\tilde{p}_{X|D,L}$, w.r.t. population, $\tilde{p}_{X|L}$

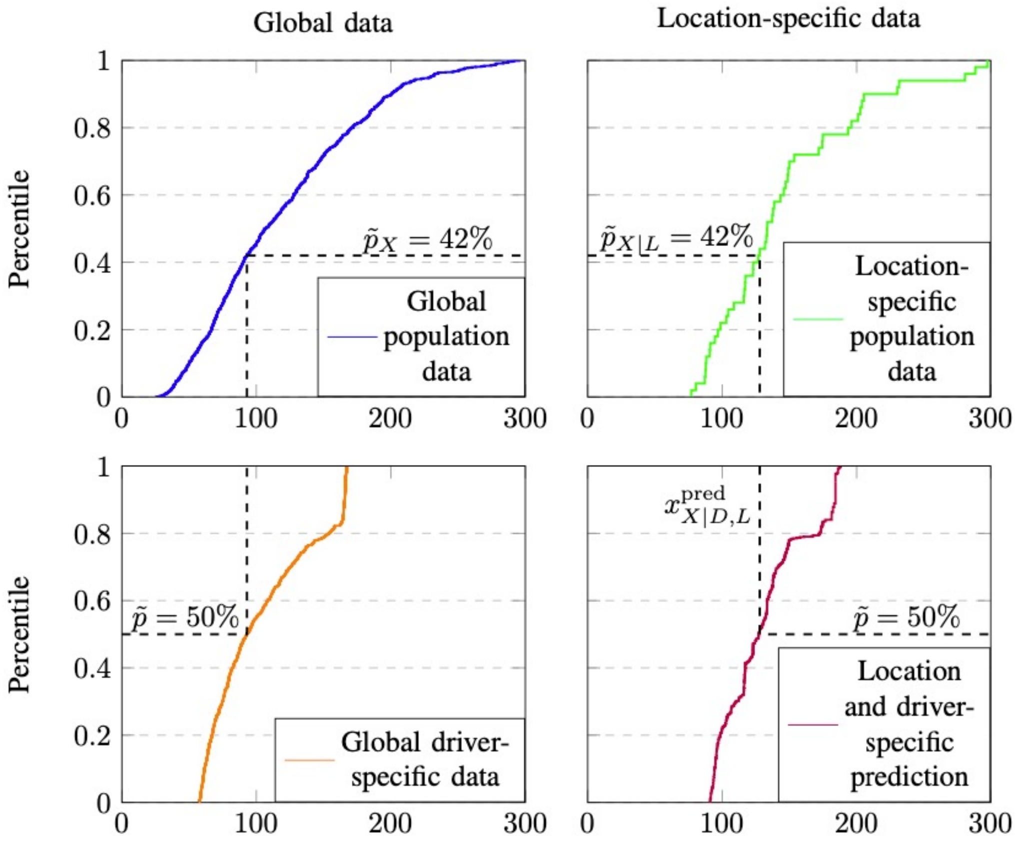
$$\tilde{x}_{X|D,L} = F_{X|D,L}^{-1}(\tilde{p})$$
$$\tilde{p}_{X|L} = F_{X|L}\left(\tilde{x}_{X|D,L}\right)$$

$F_{X|D,L}$ is unknown

To predict $F_{X|D,L}$, we use $\tilde{p}_X = \tilde{p}_{X|L}$ (ranking assumption)

$$\rightarrow \quad x_{X|D,L}^{\text{pred}} = F_{X|L}^{-1}\left(F_X\left(F_{X|D}^{-1}(\tilde{p})\right)\right)$$

Repeat prediction step for $\tilde{p} = 1\%, \dots, 99\%$



Simulation Results using SUMO

Calibration of Automated Emergency Braking

- Based on driving style of drivers
- Based on road surface condition

Simulation Setup

Given: F_X, $F_{X|D}$
Collected over time: $F_{X|L}$ (data samples collected is *x* axis)

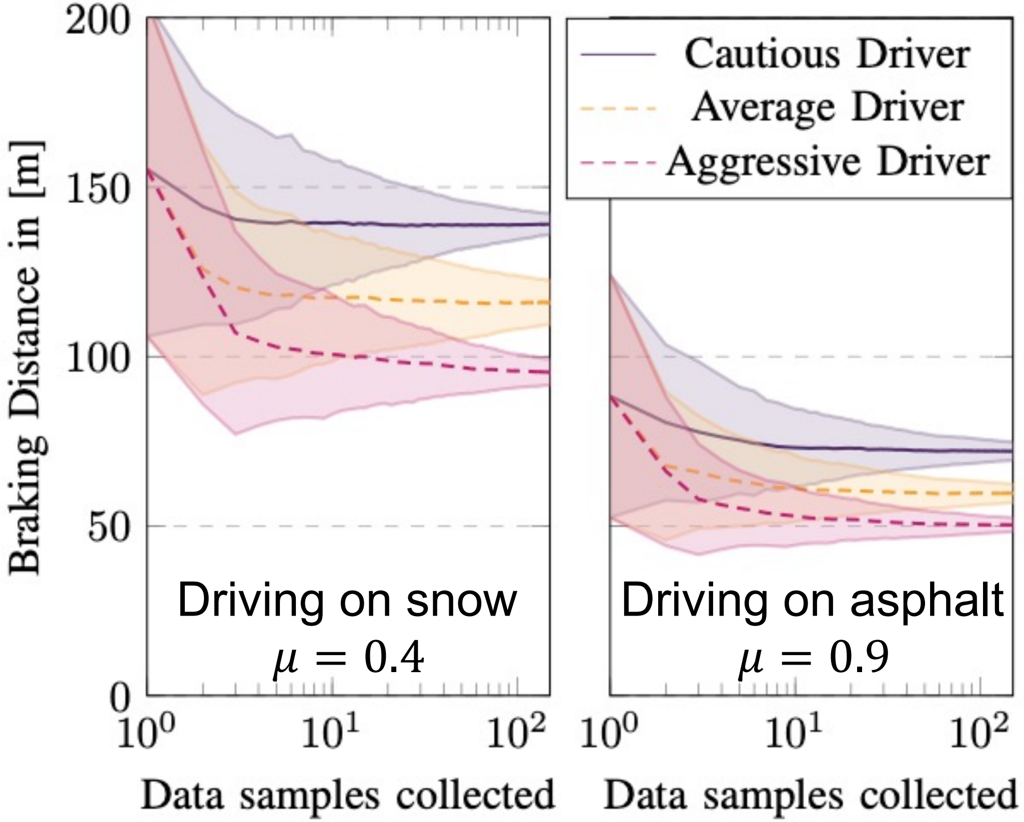
Results

- Fast separation of different environmental conditions (e.g., icy, dry, rainy) with less than 10 data points
- Fast separation of driving styles (e.g., cautious, average, aggressive) with as little as 10 data points
- Robust separation of environment and driving styles within 100 data points

Interpretation of ranking assumption in this study

- Cautious driver on asphalt is predicted to be cautious on snow
- Aggressive driver on asphalt is predicted to be aggressive on snow

How quickly can we make accurate predictions?



Overview of Research Directions