Machine Learning and Motion Coordination for Battery Electric Vehicle

Safe, Efficient and Smart Learning Framework for Online Learning

Jacopo Dallafior

Supervisor: Esteban Gelso, Maliheh Sadeghi Kati, Amir Daghestani

Examiner: Mikael Johannson

OVERVIEW

Problem statement

Research Question

Framework Overview

Main Blocks Description

Results

PROBLEM

Electric truck suffer **Limited driving range**.

Electric trailer can extend the range.

Truck Manufactures relies on third-party resulting in **restricted access** to trailer data.

To enable optimal torque distribution is essential to **understand the trailer's behavior**.

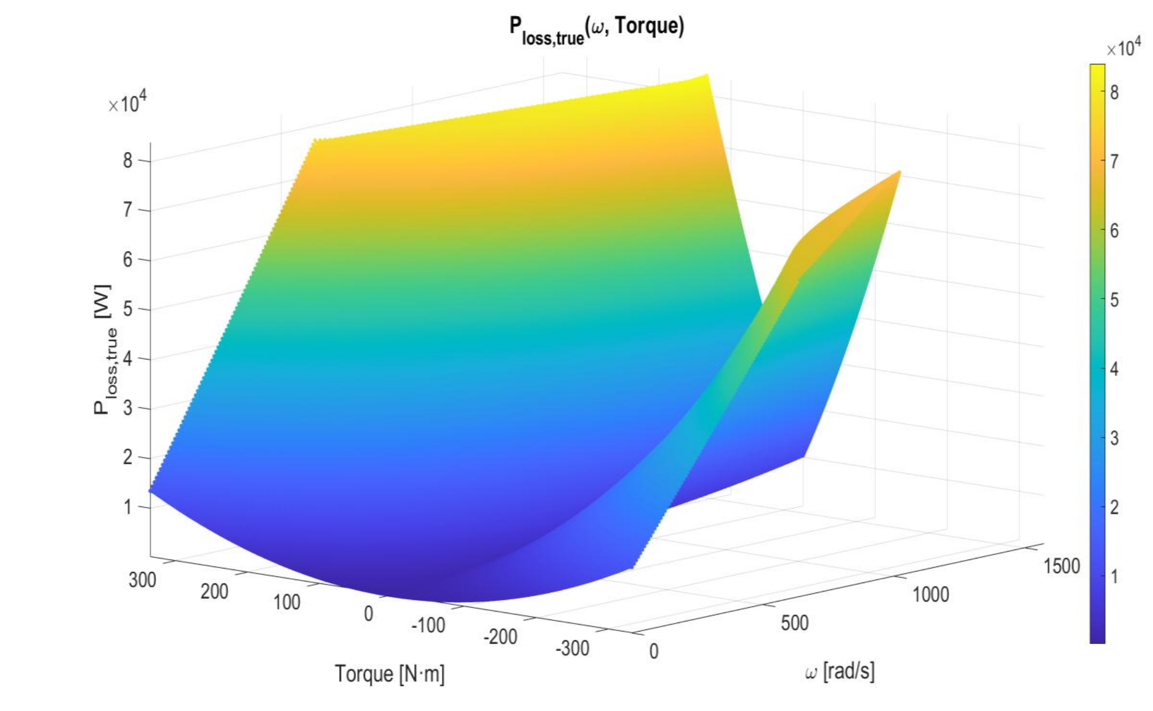
THE POWER LOSS

Power Loss Surface is a function of torque and angular speed.

Go to method in the literature for the energy optimization due to its near quadratic behavior:

$$P_{loss}(T,\omega) \approx a(\omega)T^2 + b(\omega)T + c(\omega)$$

Calculated under the assumptions of full sensor measurements avaiable



RESEARCH QUESTION

How can an articulated electric vehicle learn the power loss characteristics of an unknown trailer from limited data, and exploit this knowledge to achieve safe and efficient torque allocation?

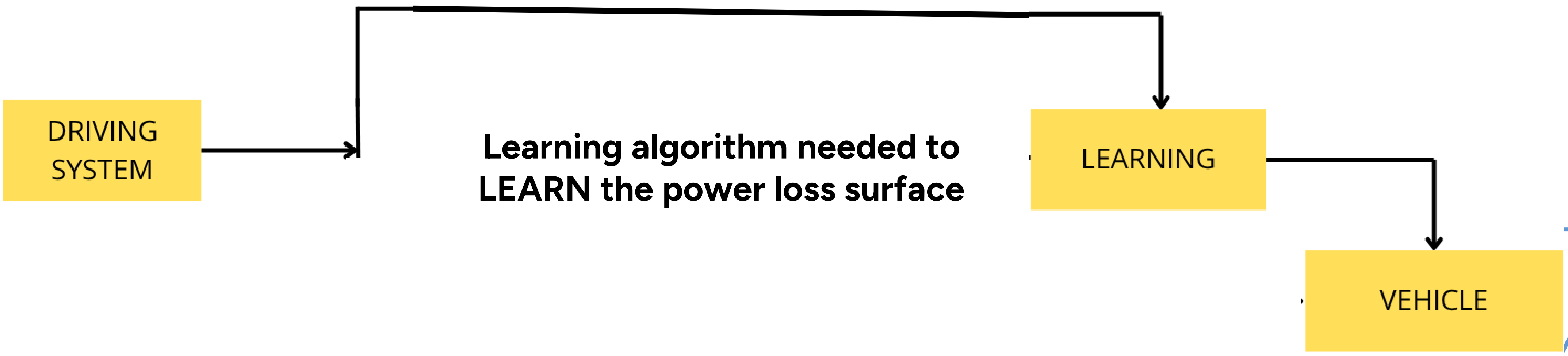
THE FRAMEWORK

DRIVING SYSTEM

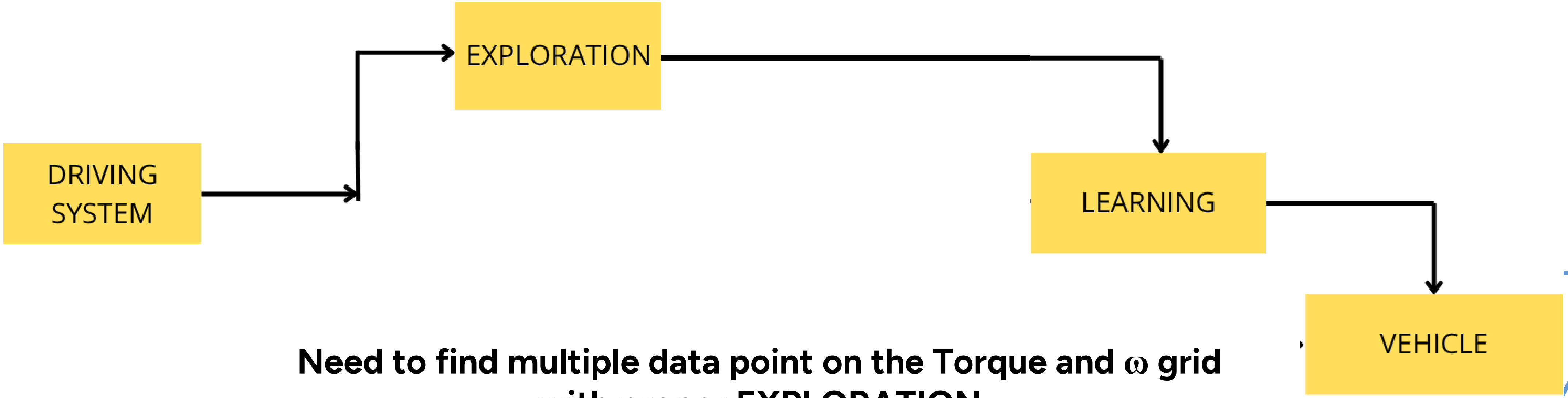
From a driving request to an optimized torque split to the articulated vehicle

VEHICLE

THE FRAMEWORK

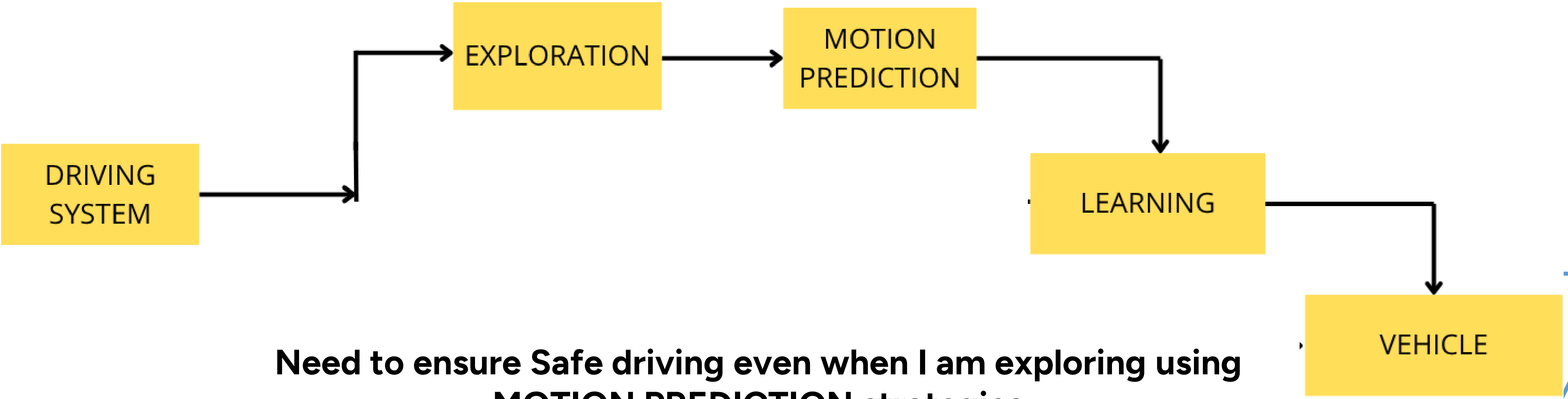


THE FRAMEWORK



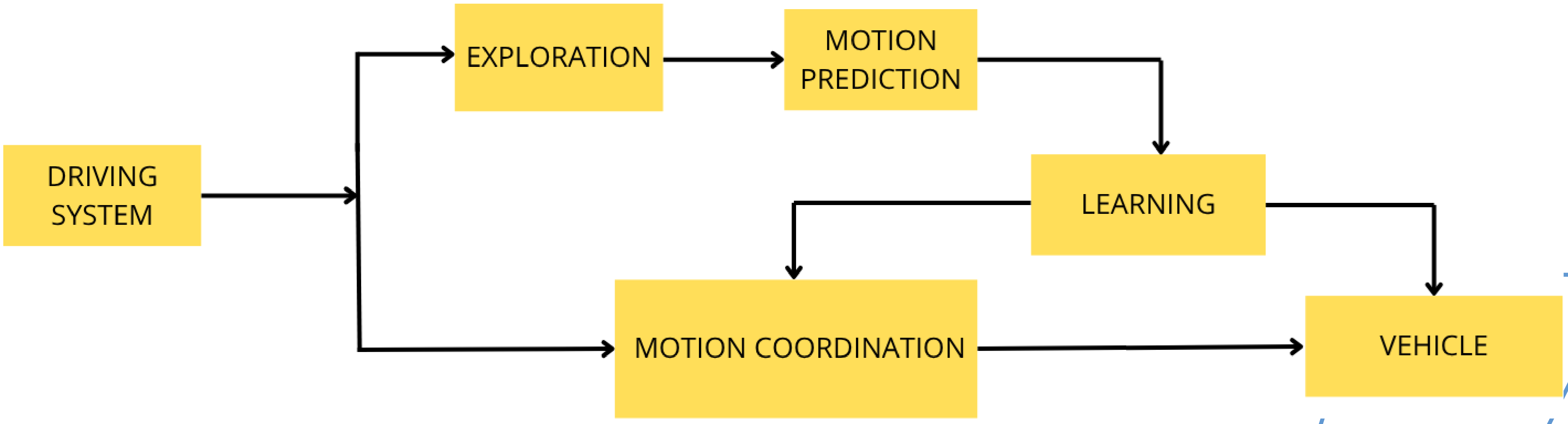
Need to find multiple data point on the Torque and ω grid with proper EXPLORATION

THE FRAMEWORK



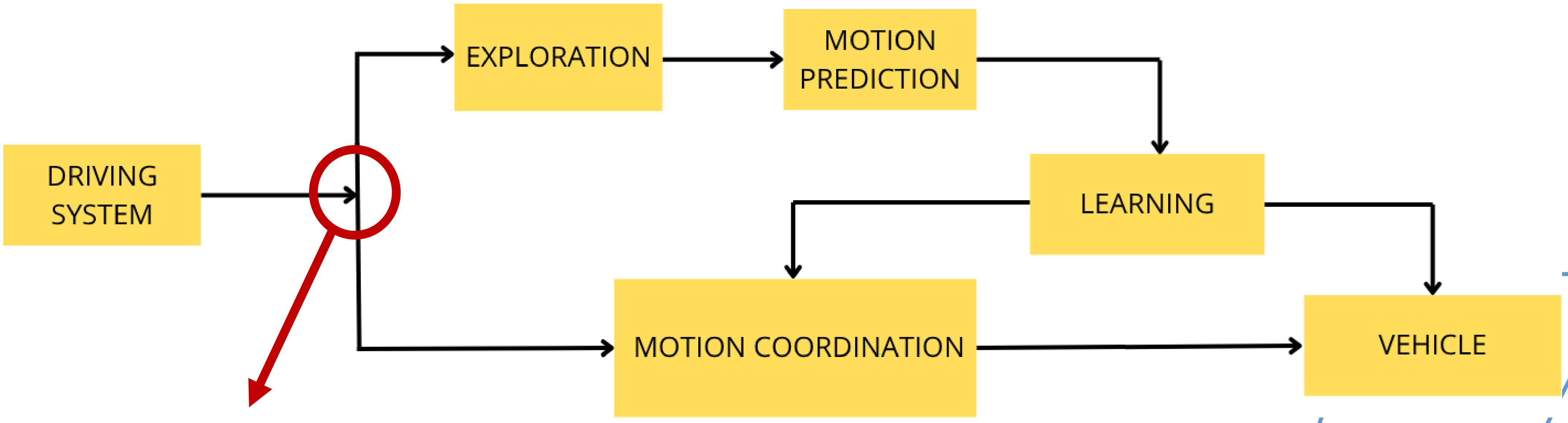
Need to ensure Safe driving even when I am exploring using MOTION PREDICTION strategies

THE FRAMEWORK



After a proper safe learning we need to EXPLOIT the knowledge to optimize the torque split

THE FRAMEWORK



Following a RL approach, an ε −greedy inspired strategy is used to balance the exploration and exploitation

MAIN BLOCKS

LEARNING EXPLORATION EXPLOITATION MOTION COORDINATION

LEARNING

In order to understand how the power loss behave we need a LEARNING algorithm
2 main strategies are adopted:

- under the polynomial assumptions we use system identification with LEAST SQUARES method

- using non parametric approach to have more freedom with MACHINE LEARNING

SYSTEM IDENTIFICATION METHOD

- Previous research used the parametric approach with ORDINARY LEAST SQUARES

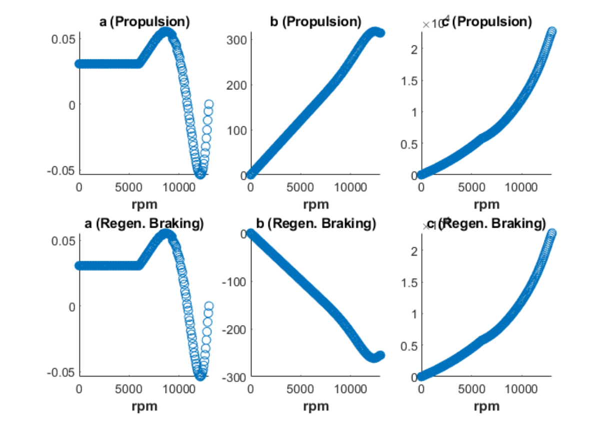
$$\theta^* = \arg\min_\theta \sum_{i=1}^{n} (y_i - \phi(x_i)^\top \theta)^2 \longrightarrow \theta^* = (\Phi^\top \Phi)^{-1} \Phi^\top y,$$

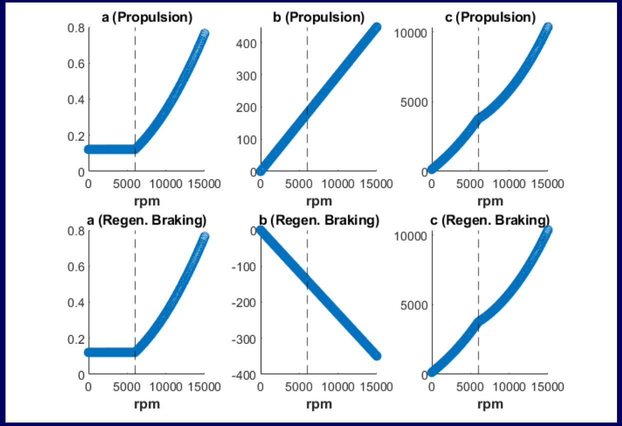
- Here enanched for online usage with RECURSIVE LEAST SQUARES

$$\theta^* = \arg\min_\theta \sum_{i=1}^{k} \lambda^{k-i} (y_i - \phi(x_i)^\top \theta)^2 \longrightarrow \theta_k = \theta_{k-1} + K_k(y_k - \phi(x_k)^\top \theta_{k-1})$$

Need to find the most scalable polynomial for the power loss coefficients

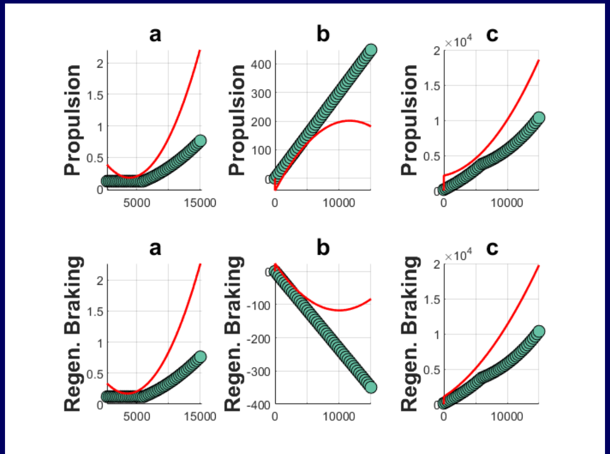
PMSM



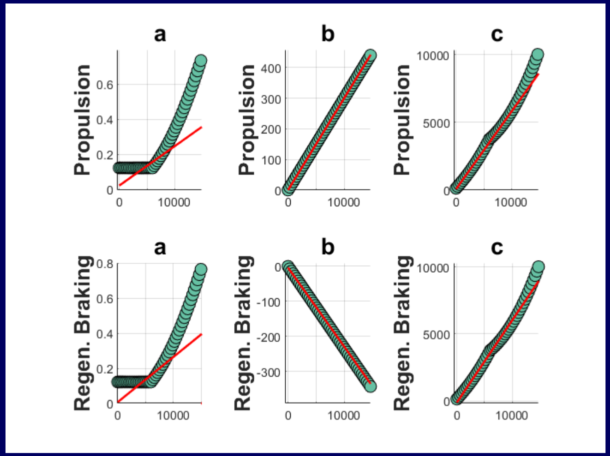


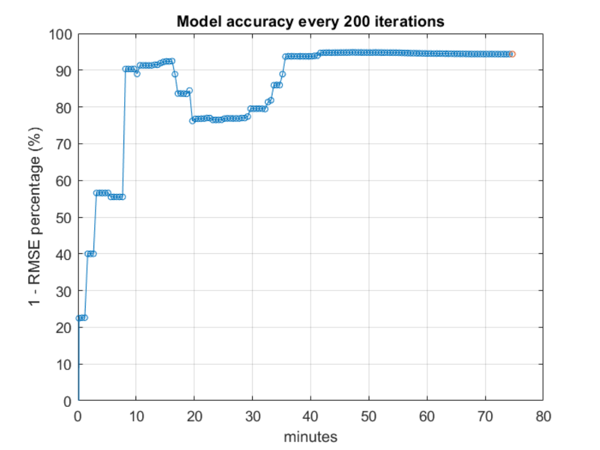
INDUCTION

QUADRATIC



LINEAR





Showing highest accuracy overall the LINEAR fitting

LEARNING | EXPLORATION | EXPLOITATION | MOTION COORDINATION

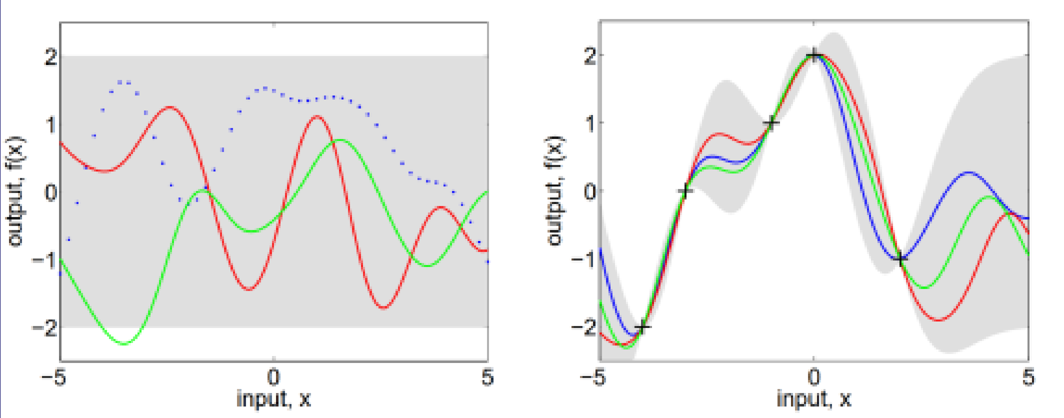
GAUSSIAN PROCESS REGRESSION METHOD

Non-parametric regression approach, flexible and data-driven, adapting the model complexity to the observed data.

$$f(x) \sim \mathcal{GP}\big(m(x), k(x, x')\big)$$

$m(x) = E[f(x)]$ the mean function

$k(x, x') = E[(f(x) - m(x))\,(f(x') - m(x'))]$ the kernel



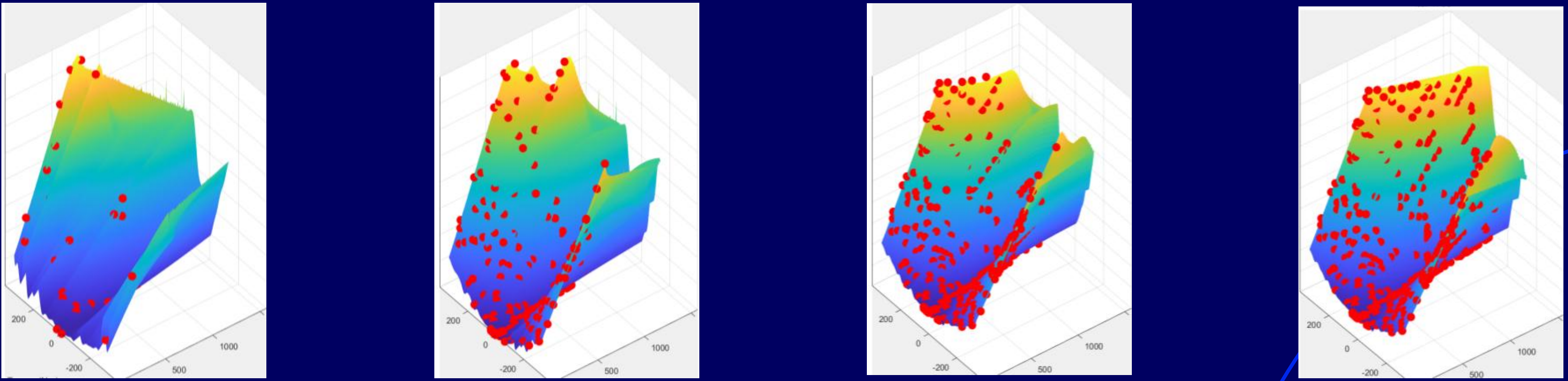
(a), prior (b), posterior

The **GPR** allows:

Better generalization in regions with limited training data

Uncertainty quantification, providing a confidence interval alongside predictions

Model the sensor noise using a heteroscedastic approach



TIME EVOLUTION

2 min | 12 min | 20 min | 28 min

LEARNING → EXPLORATION → EXPLOITATION → MOTION COORDINATION

EXPLORATION

To speed up the learning phase we need to find the most significative point to cover as much as possible the Torque and ω grid.

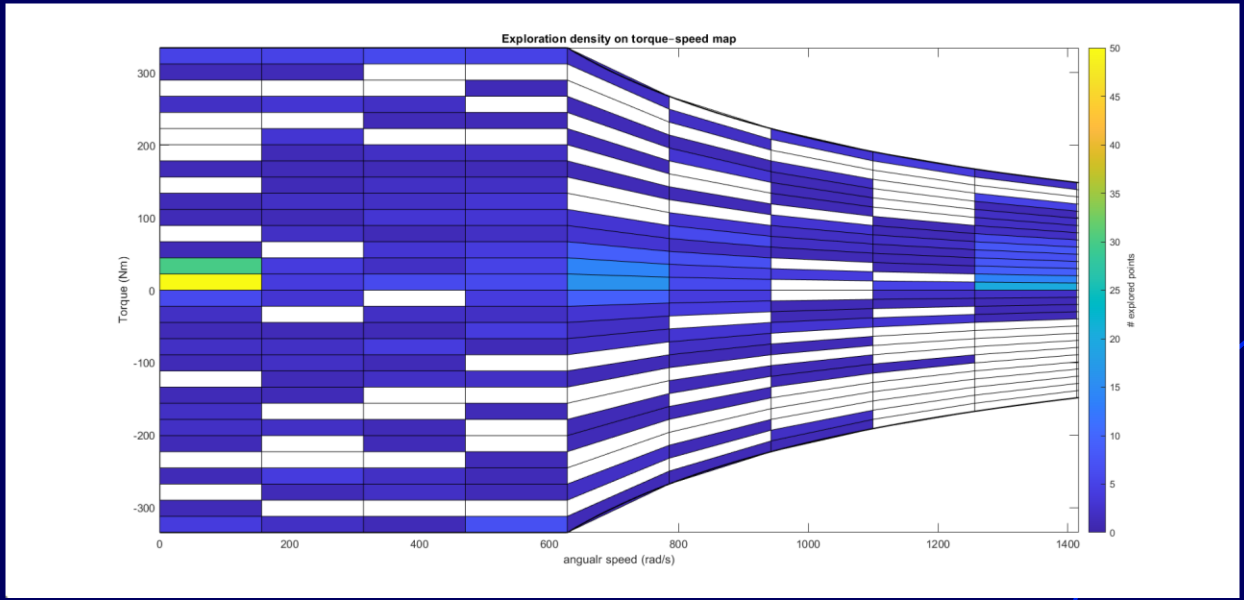
2 differents strategies adopted based on the type of learning algorithm:

- HEURISTIC GRID SEARCH for the RLS

- UPPER CONFIDENCE BOUND for the GPR

HEURISTIC GRID METHOD

- The heuristic grid method divides the torque–speed space into cells and tracks visit counts. Unexplored or under-sampled cells are then targeted, ensuring even coverage.



LEARNING | EXPLORATION | EXPLOITATION | MOTION COORDINATION

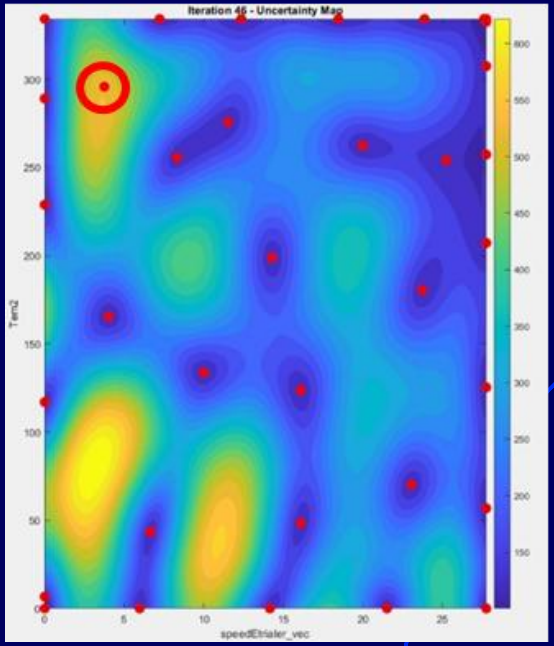
UPPER CONFIDENCE BOUND

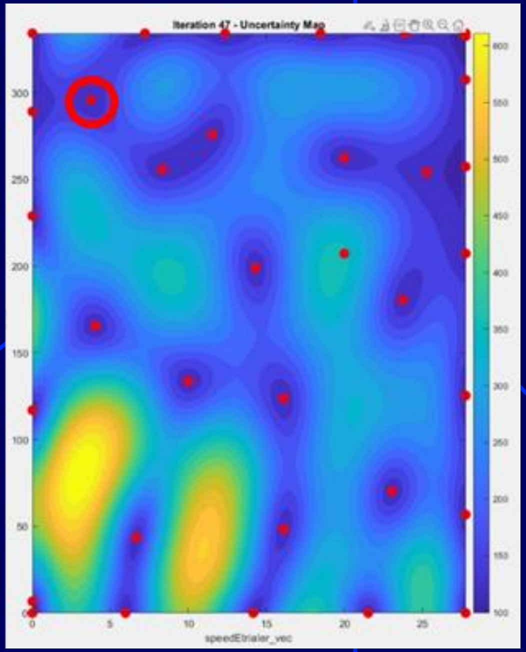
- Following Bayesian Optimization idea, applied to actively explore regions with the highest model uncertainty, as estimated by the Gaussian Process.

- Constrained version (C-UCB) for motor limits used in the framework

$$\max_{T_2} \quad \mu(T_2) + \beta \cdot \sigma(T_2)$$

$$\begin{aligned} s.t. \quad & T_{\min} \leq T_1 \leq T_{\max} \\ & T_{\min} \leq T_2 \leq T_{\max} \\ & |T_2 - T_{2,\text{previous}}| \leq \Delta T_{\max} \end{aligned}$$





EXPLOITATION and CONTROL

After and during the learning we want to exploit the learned power loss surface

- we want to minimize losses:

$$\min_{T_{\text{truck}},T_{\text{trailer}}} \quad P_{\text{loss},truck}(T_{truck}) + P_{\text{loss},trailer}(T_{trailer})$$

$$\begin{aligned} s.t. \quad & T_{min,trailer} \leq T_{trailer} \leq T_{max,trailer} \\ & Bu \; = \; v \\ & sign(T_{truck}) = sign(T_{trailer}) \\ & T_{min,truck} \leq T_{truck} \leq T_{max,truck} \end{aligned}$$

LEARNING | EXPLORATION | EXPLOITATION | MOTION COORDINATION

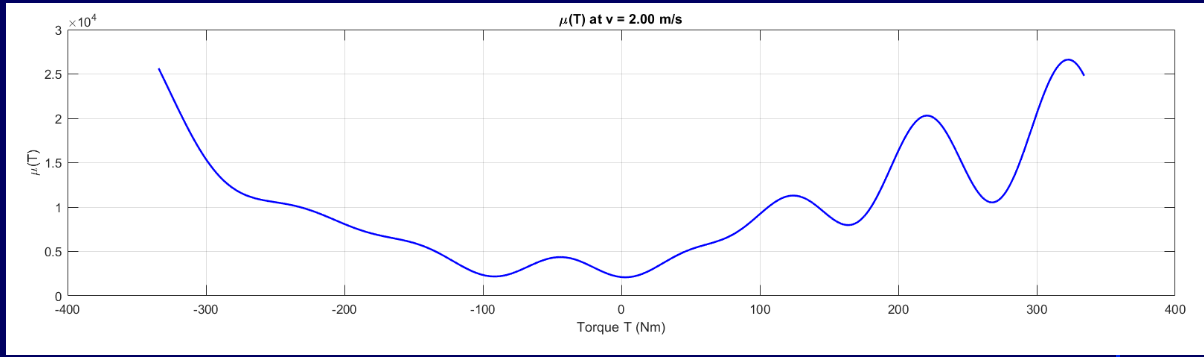
Two main technique used to solve it:

QUADRATIC PROGRAMMING for the RLS learning method

$$\min_{x \in R^n} \frac{1}{2} x^\top H x + g^\top x \longrightarrow \frac{1}{2} u^\top \begin{bmatrix} 2\, a_{truck} & 0 \\ 0 & 2\, a_{trailer} \end{bmatrix} u + \begin{bmatrix} b_{truck} \\ b_{trailer} \end{bmatrix} u$$

SEQUENTIAL QUADRATIC PROGRAMMING for the GPR with multistart

- GPR learning during initial phase is highly non-convex



LEARNING → EXPLORATION → **EXPLOITATION** → MOTION COORDINATION

Generic formulation

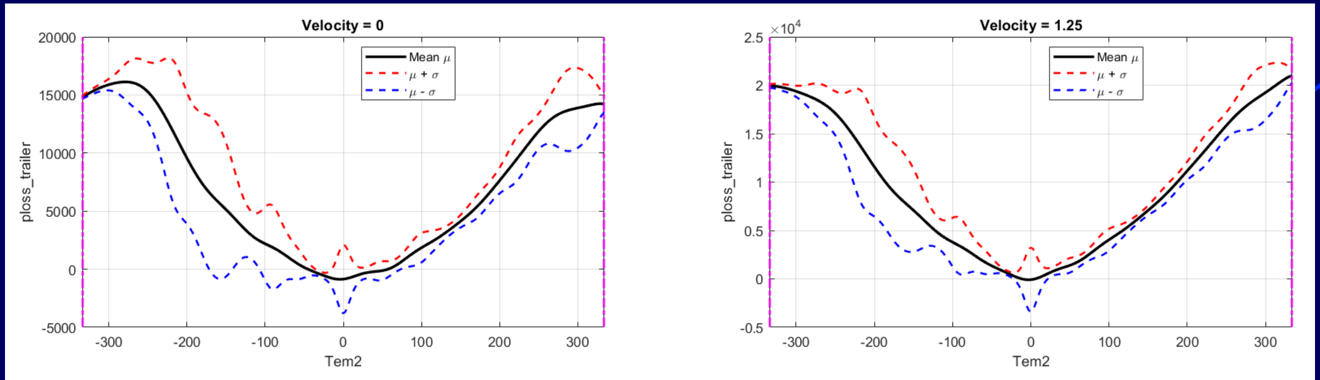
$$\min_{x \in R^n} f(x) \quad \text{s.t.} \quad g_i(x) \leq 0,\ h_j(x) = 0$$

at each iteration k

$$\min_{d \in R^n} \frac{1}{2} d^\top H_k d \ + \nabla f(x_k)^\top d \ \ s.t. \ \ \nabla h(x_k) d + h(x_k) = 0 \quad \nabla g(x_k) d + g(x_k) \leq 0$$

Using SQP consent to introduce also the **uncertainty term** of the GPR into the cost

$$\min_{T_{\text{truck}}, T_{\text{trailer}}} P_{\text{loss,truck}}(T_{\text{truck}}) \ + \ \mu_{\text{trailer}}(T_{\text{trailer}}) \ + \alpha\, \sigma_{trailer}(T_{trailer})$$



LEARNING EXPLORATION EXPLOITATION MOTION COORDINATION

MOTION PREDICTION and SAFETY

In order to maintain safety during the travel two diffrent strategies adopted:

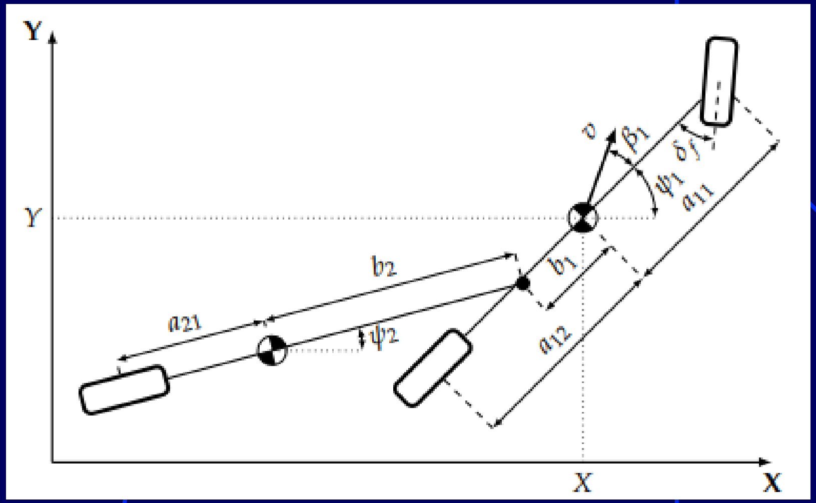
- **ROLLOUT SIMULATION** during EXPLORATION

- **CONTROL BARRIER FUNCTION** during EXPLOITATION

The system used is a single-track model

INPUT: torque for both the two units and steering angle

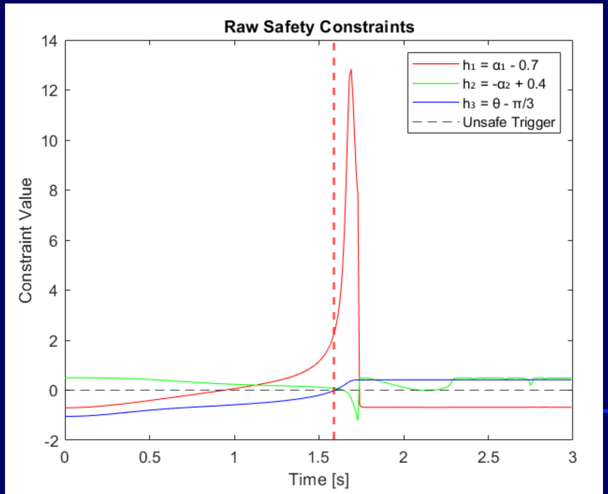
OUTPUT: articulation angle, slip angle



LEARNING
EXPLORATION
EXPLOITATION
MOTION COORDINATION

Once all the constraints in the set are triggered ⟶ **UNSAFETY**

ROLLOUT SIMULATIONS intervene as reactive detection

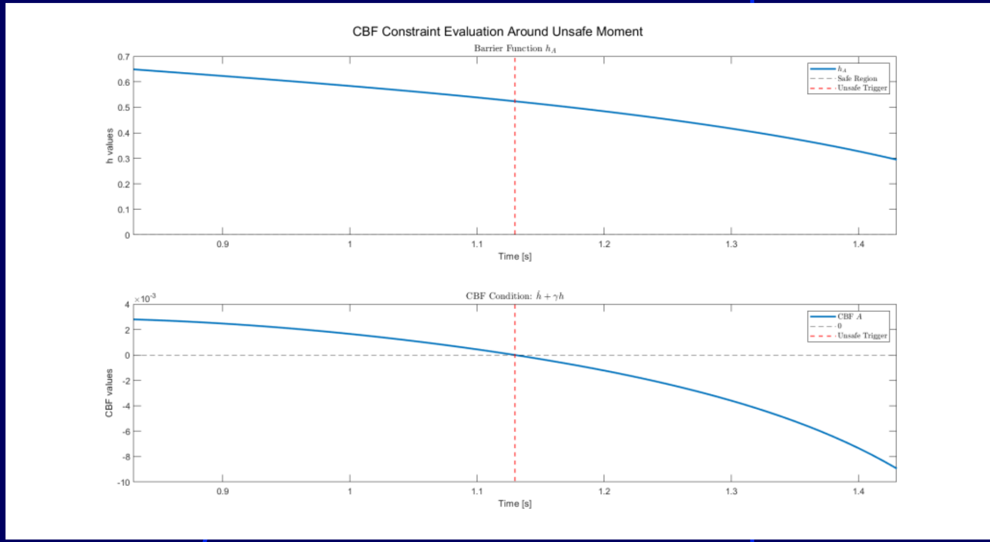


CONTROL BARRIER FUNCTION

considering the safe set $C = \{x \in R^n \mid h(x) \geq 0\}$

CBF condition act as a filter when the system risks leaving the safe set enforcing invariance

$$\dot{h}(x,u) + \alpha(h(x)) \geq 0$$



LEARNING → EXPLORATION → EXPLOITATION → **MOTION COORDINATION**

RESULTS

LEARNING EXPLORATION EXPLOITATION MOTION COORDINATION

LEARNING ACCURACY RESULTS

RLS learning method lead to the following accuaycy results for different motors

Motor Type	Polynomial Type	Accuracy (%)
PMSM	Linear	93.75
PMSM	Quadratic	92.00
Induction Motor	Linear	82.60
Induction Motor	Quadratic	81.00

GPR method achive the same accuracy over the different motors, showing higher accuracy with the usage of a proper **KERNEL**

Kernel Type	Accuracy (%)
ARD Squared Exponential Kernel	92.80
Isotropic Squared Exponential Kernel	97.46
Matérn 5/2 ARD Kernel	79.80
Cubic Polynomial ARD Kernel	74.00

Accuracy variability across runs (different seeds): **±1.1% (ISO CASE) for GPR**

ENERGY CONSUMPTION ANALYSIS

To compare the final energy saving, two baseline are used on different drive cycle.

Strategy	Göteborg–Borås cycle
Rule-based baseline	406.00
Full-knowledge (oracle)	389.27

Considering the scenario where the system start learning and then exploiting

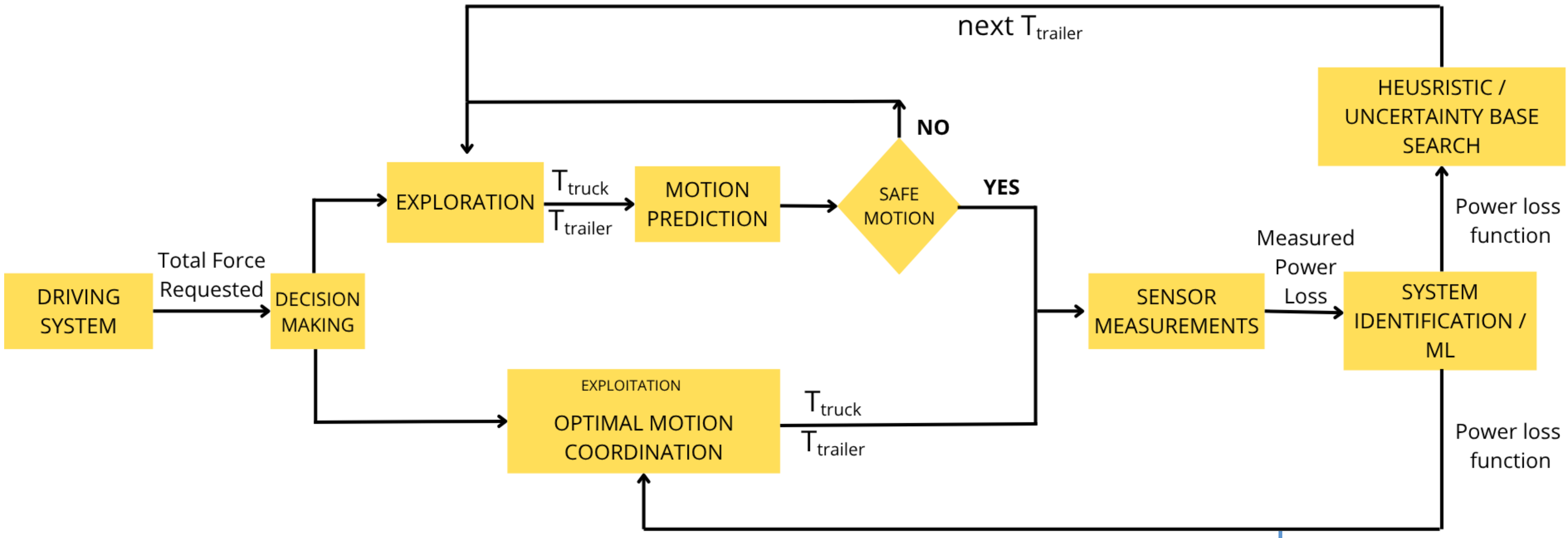
strategy	energy (kJ)	ratio vs. rule-based	ratio vs. full-knowledge
RLS	415.19	102.2%	106.7%
GP	410.46	101.1%	105.4%

for long drive, once the power loss is learned, the GP shows superior performance

strategy	energy (kJ)	ratio vs. rule-based	ratio vs. full-knowledge
RLS (post-learning)	397.52	97.9%	102.1%
GP (post-learning)	391.40	96.4%	100.5%

Almost same consumption as full knowledge

THE RESULTING FRAMEWORK



FINAL CONSIDERATION

Framework validated: safe online learning for articulated e-vehicles is feasible.

RLS baseline: simple and fast, but limited accuracy and higher energy consumptions.

GPR advantage: higher accuracy, stable across runs, robust to noise.

Vehicle-level gains: up to **4% energy saving**, operating within **1% of full knowledge** while ensuring safety.

THANK YOU

BACKUP SLIDES

LOSS ANALYSIS

The benefit of using the GPR can be seen with the **energy loss**

	Göteborg–Borås cycle
Rule-based baseline	178.04
Full-knowledge (oracle)	160.36

GPR shows lower losses than the rule-based baseline: exploration may cost more energy, but it improves the model, so the real benefit appears after learning during exploitation.

strategy	energy (kJ)	ratio vs. rule-based	ratio vs. full-knowledge
RLS	185.94	104.4%	116.0%
GP	177.00	99.4%	110.4%
RLS post-learning	168.52	94.7%	105.1%
GP post-learning	162.49	91.3%	101.3%

ENERGY ANALYSIS

The benefit of using the GPR can be seen with the **energy loss**

Strategy	WHVC cycle
Rule-based baseline	142.23
Full-knowledge (oracle)	137.01

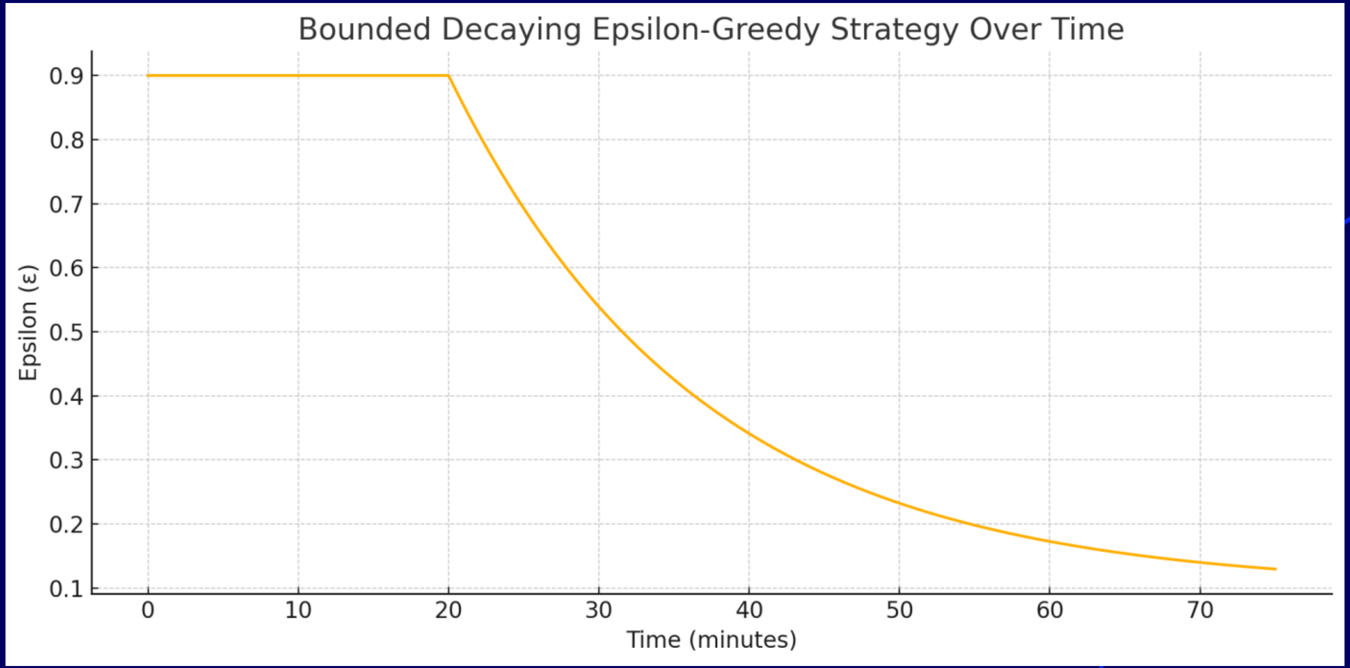
RLS explore at a lower cost during learning, but GP's exploration and retraining lead to better long-term performance closer to the full-knowledge optimum.

strategy	energy (kJ)	ratio vs. rule-based	ratio vs. full-knowledge
RLS	143.25	100.7%	104.5%
GP	143.59	100.9%	104.8%
RLS (post-learning)	140.10	98.5%	102.2%
GP (post-learning)	138.92	97.6%	101.3%

EXPLORATION RATE

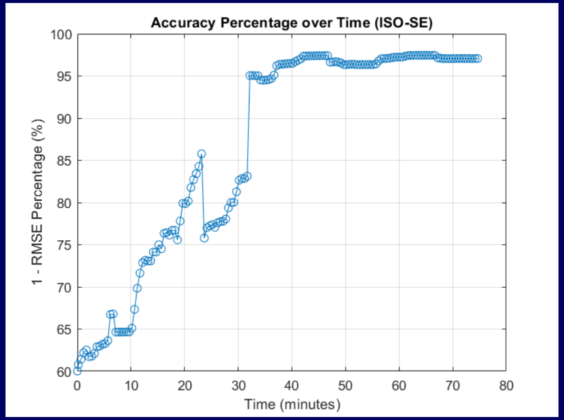
Simulation results indicated that atleast 20 minutes of high exploration activity are required to achievemeaningful coverage.
After this period, the exploration rate begins to decay over time according to:

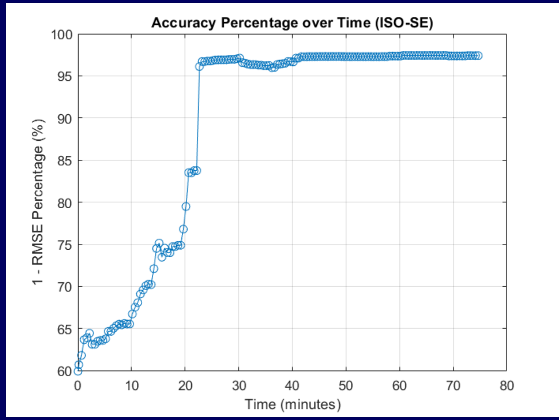
$$\varepsilon(t) = \varepsilon_{min} + (\varepsilon_{max} - \varepsilon_{min}) \cdot e^{-\lambda t}$$



GP ACCURACY

Across different motor the accuracy remain the same for the same kernel



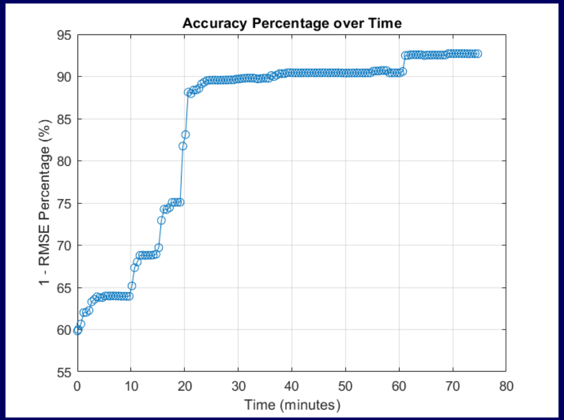


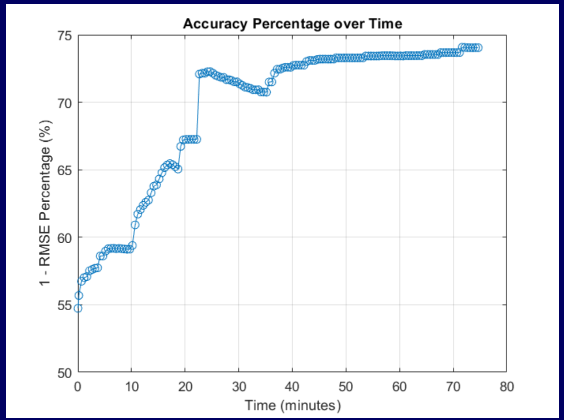
ISO - SE

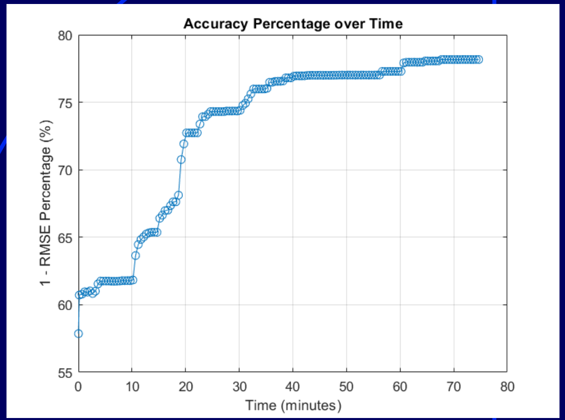
ARD - SE

POLYNOMIAL

MATERN ARD

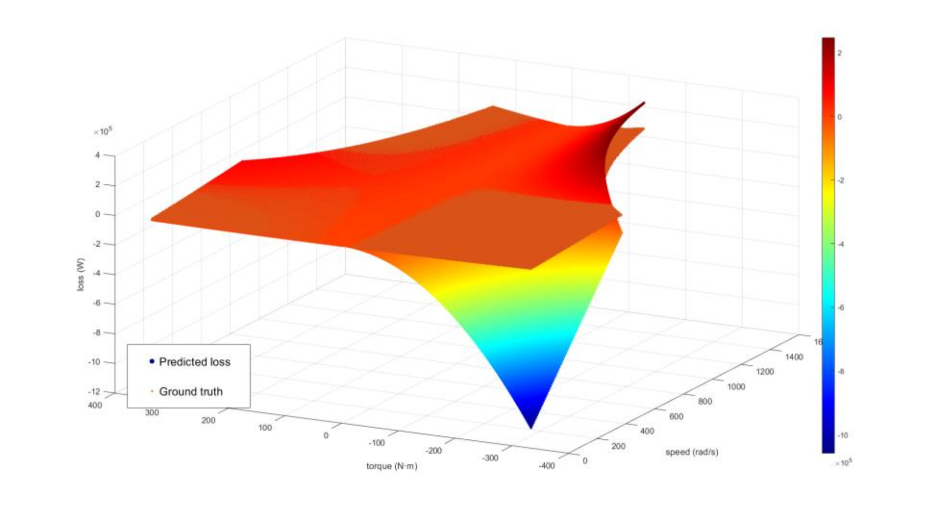


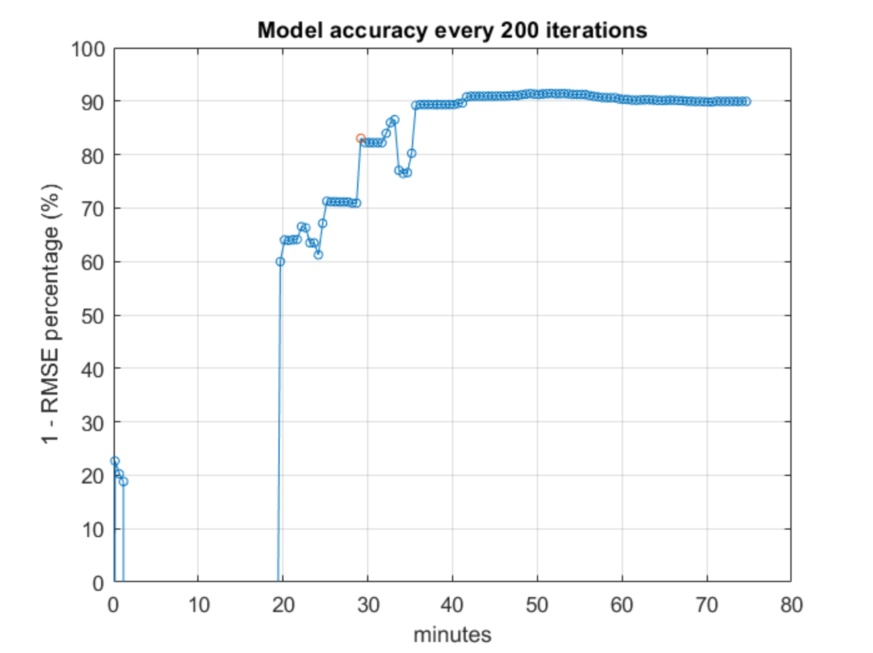




NOISE MODELING

RLS suffers while is fitting the noise that comes from the sensors, and in the initial phase can lead to big fitting errors





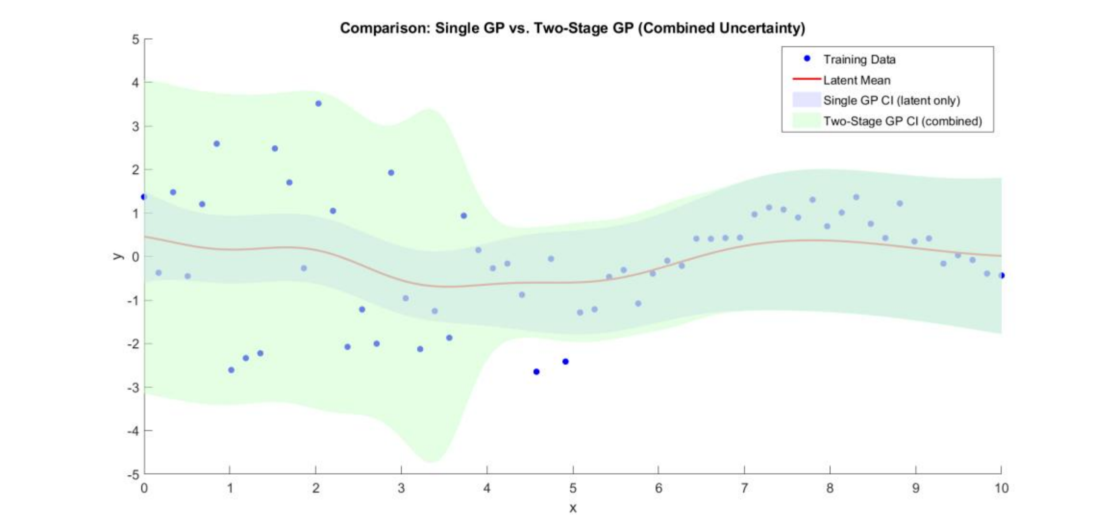
HETEROSCHEDASTIC NOISE

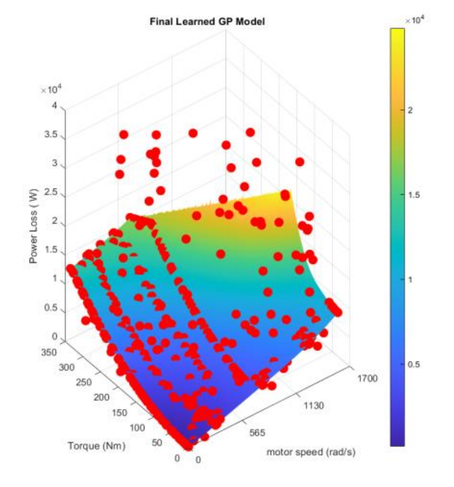
Considering the GPR that uses as training point noisy measurements.

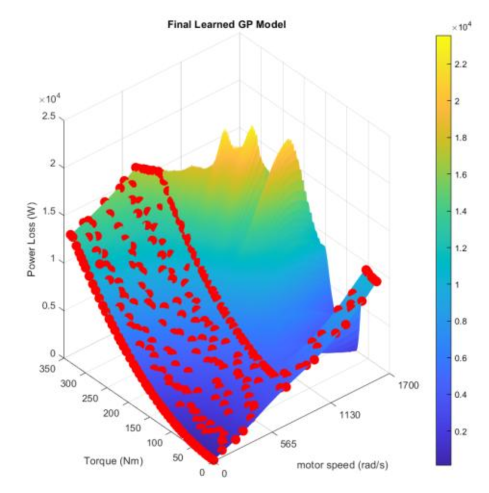
$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}\left(0, \sigma_n^2(x_i)\right)$$

This is incorporated into a composite kernel function

$$k_{\text{obs}}(x_i, x_j) = k_{\text{SE}}(x_i, x_j) + \delta_{ij}\sigma_n^2(x_i))$$

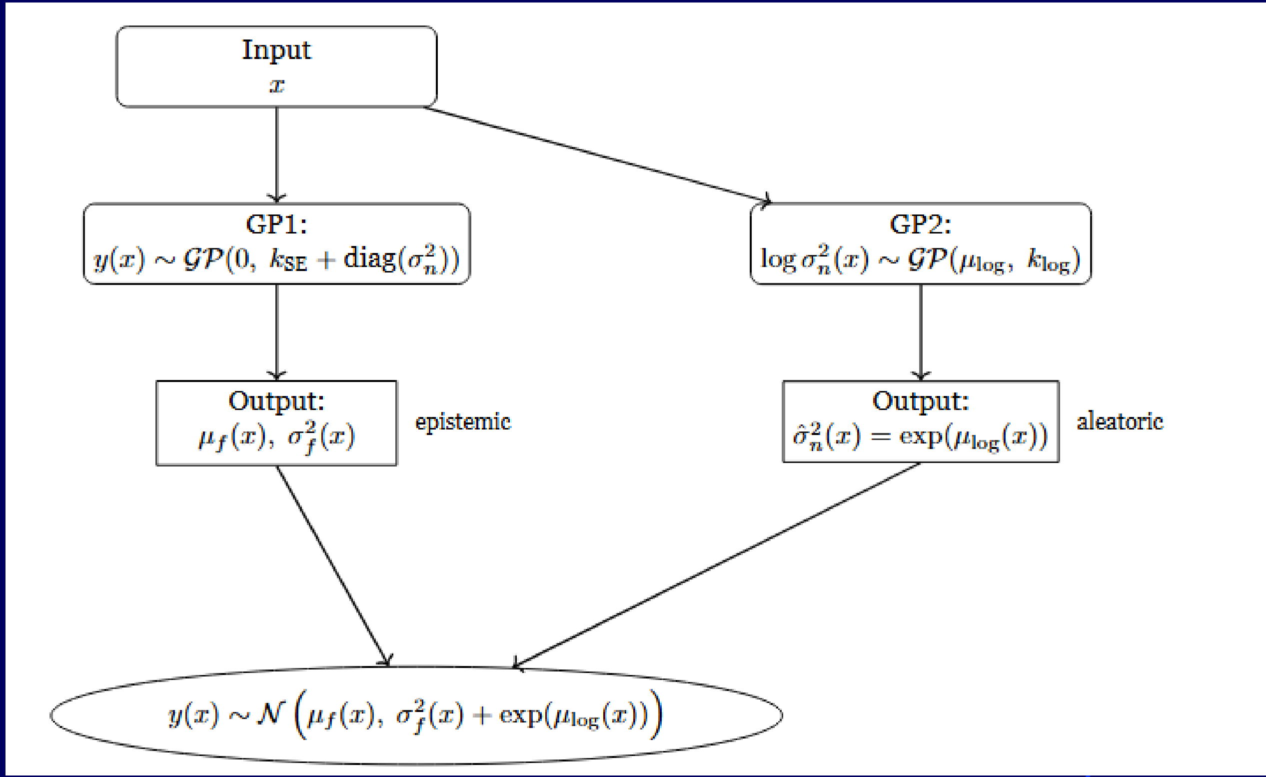




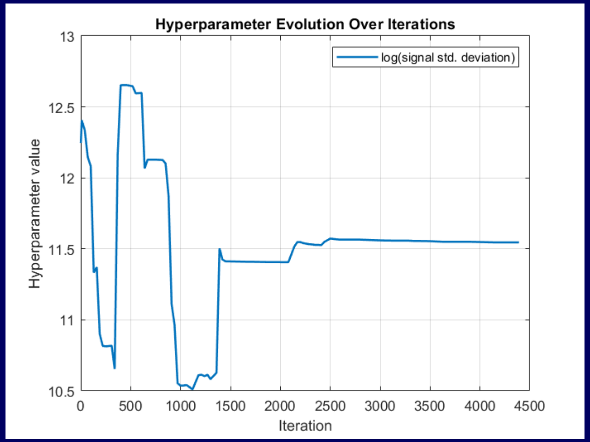


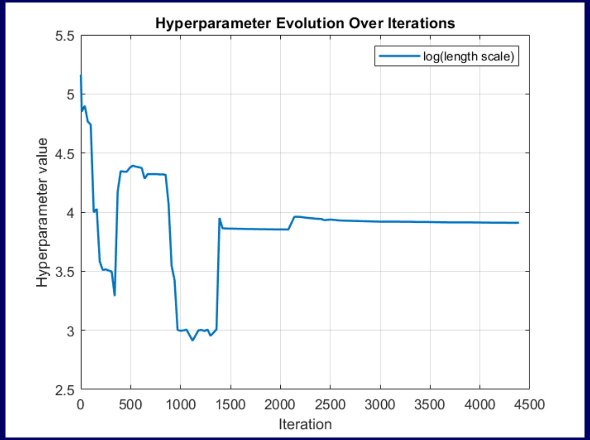
$$y(x_*) \sim \mathcal{N}\left(\mu_f(x_*),\ \underbrace{\sigma_f^2(x_*)}_{\text{epistemic}} + \underbrace{\widehat{\sigma_n^2}(x_*)}_{\text{aleatoric}}\right)$$

HETEROSCHEDASTIC NOISE



HYPERPARAMETERS OPTIMIZATION





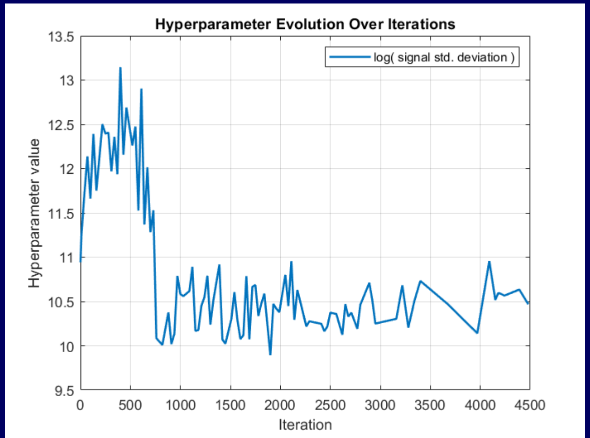
Good hyperparameter initialization lead to convergengence after approx. 25 mins.

Kernel and likelihood involve hyperparameters:
$(\theta = \{\sigma_f^2, \ell_j, \sigma_n^2\})$

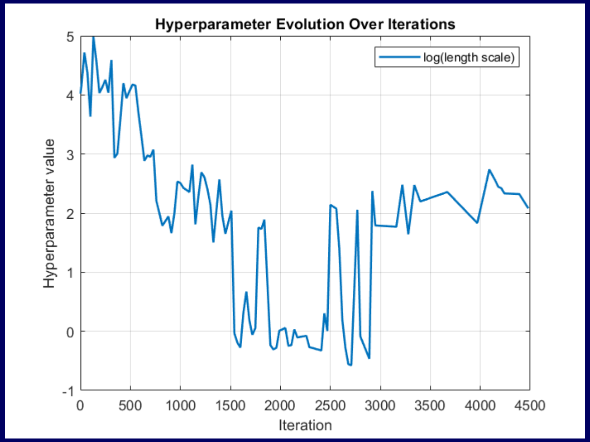
A principled approach is to optimize the log marginal likelihood:

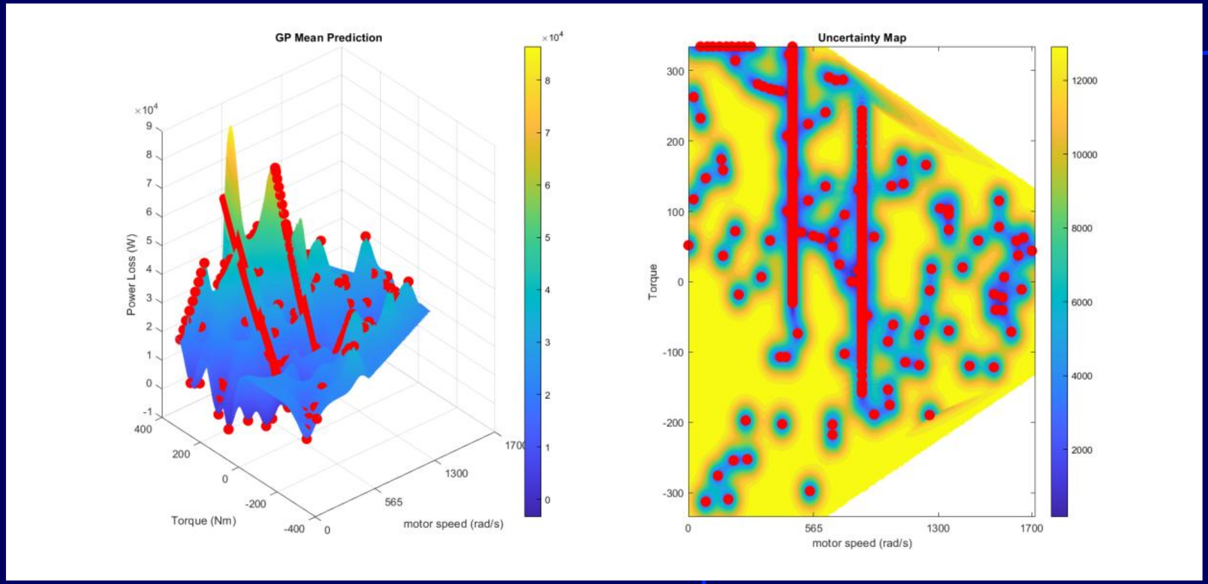
$$\log p\left(\mathbf{y}|\mathbf{X}, \theta\right) = -\frac{1}{2}\mathbf{y}^\top\left(\mathbf{K}_\theta + \sigma_n^2\mathbf{I}\right)^{-1} y - \frac{1}{2}\log\left|\mathbf{K}_\theta + \sigma_n^2\mathbf{I}\right|\frac{\mathrm{n}}{2}\log 2\pi$$

HYPERPARAMETERS OPTIMIZATION



Bad hyperparameter initialization lead to never reach a stable convergence leadings to poor fitting and in the worst case overfitting.



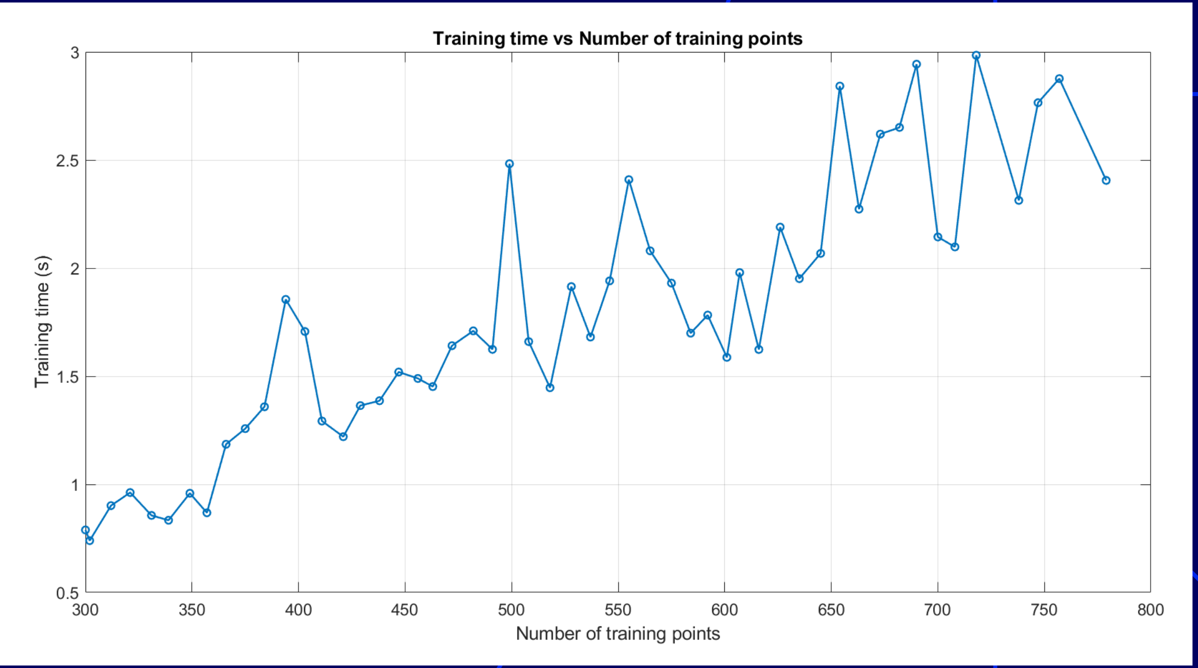


TRAINING TIME GP

The true bottolneck of the GP is the training time.

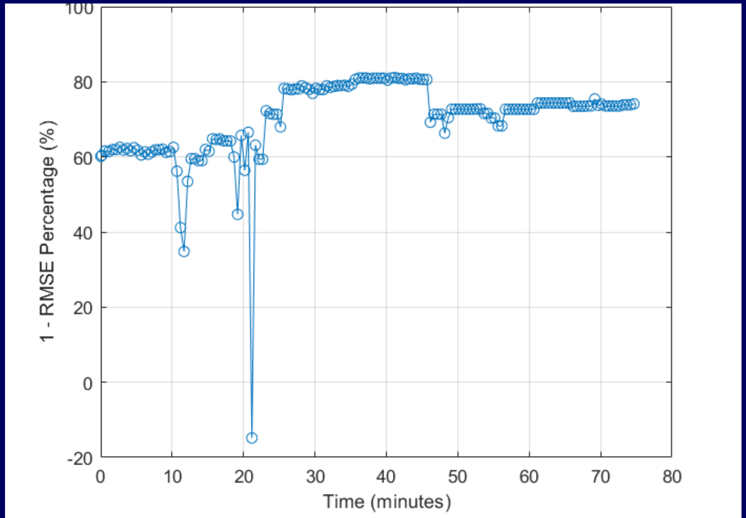
The highest time of training for 800 datapoints is almost 3 seconds.

The $O(n^3)$cost comes from the Cholesky factorization of the covariance matrix during training. Predictions are cheaper: $O(n)$for the mean and $O(n^2)$for the variance per test point.

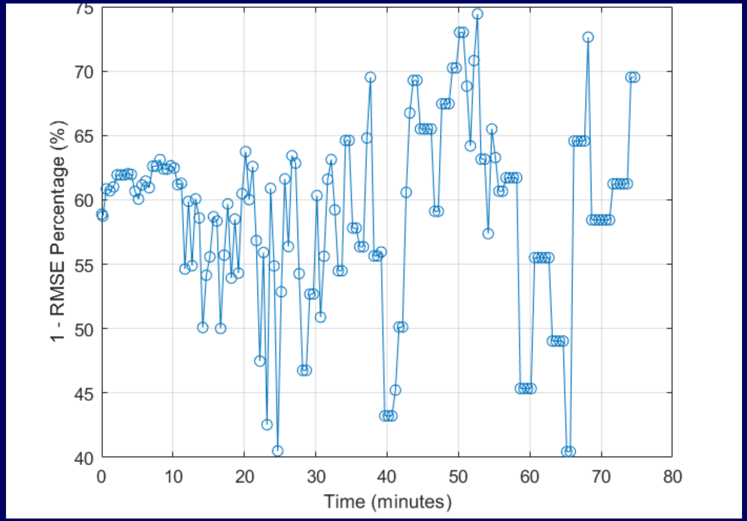


SPARSE SOLUTION

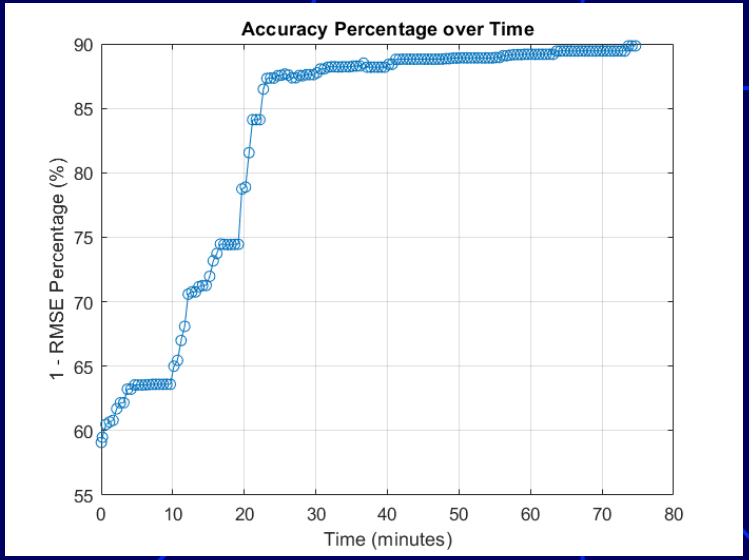
Farthest Point Sampling



k-Means

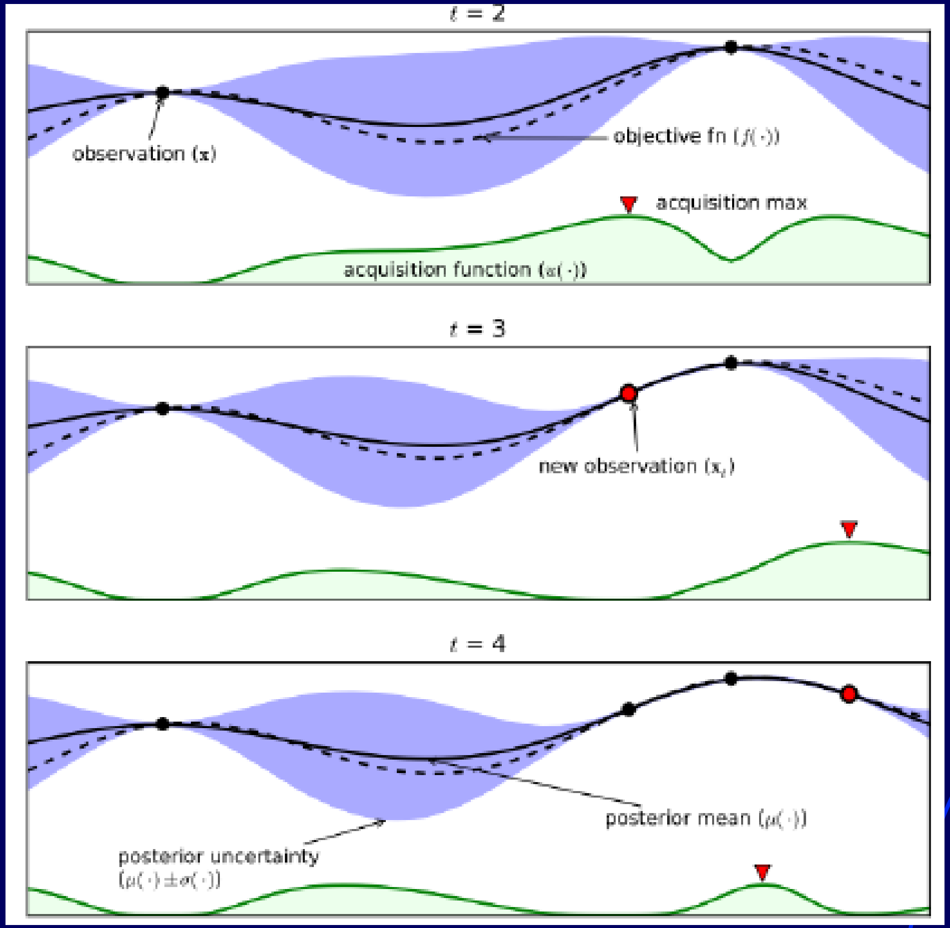


30% k-Means 70% FPS



Model Type	Maximum Accuracy (%)	Computational Complexity
Full Gaussian Process	97.46	$\mathcal{O}(n^3)$
Sparse Gaussian Process	89.70	$\mathcal{O}(nm^2)$

UPPER CONFIDENCE BOUND



For any set of inputs, the GP prior implies that the training outputs f and the function values at test inputs f^* are jointly Gaussian :

$$\begin{bmatrix} y \\ f^* \end{bmatrix} = \begin{bmatrix} K(X,X) + \sigma^2 * \mathrm{I} & K(X,X^*) \\ K(X^*,X) & K(X^*,X^*) \end{bmatrix}$$

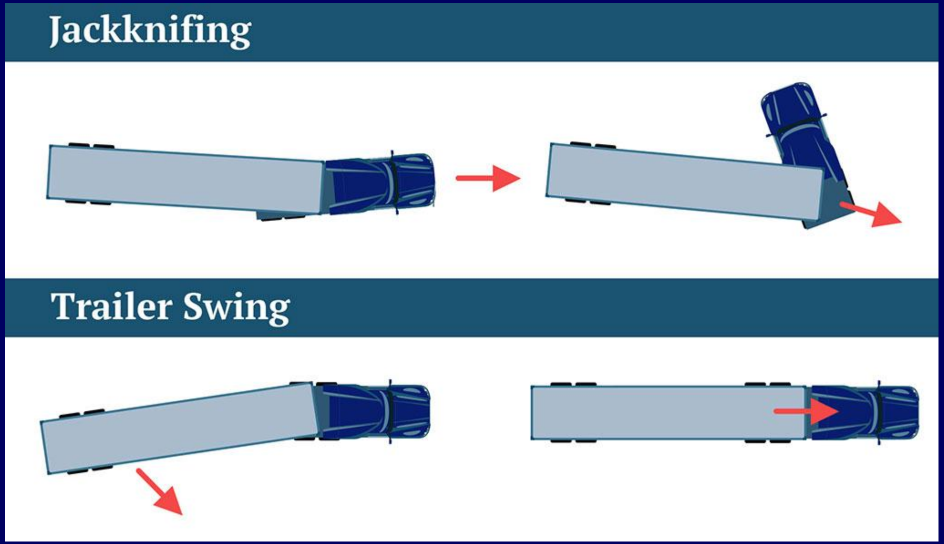
Conditioning this joint Gaussian on the observed training outputs yields the posterior distribution:

$$\mu(x_*) = K(X_*,X)(K(X,X) + \sigma_n^2 I)^{-1} y$$

$$\sigma^2(x_*) = K(X_*,X_*) - K(X_*,X)(K(X,X) + \sigma_n^2 I)^{-1} K(X,X_*)$$

UNSAFE MODES

Two main UNSAFE modes are considered during the whole study of the thesis:



The state treshold are base on a previous thesis that consider the following set of constraint:

$$\alpha_{12} = \left|\frac{v_{y,12}}{v_{x,12}}\right|, \quad \alpha_{22} = \left|\frac{v_{y,22}}{v_{x,22}}\right|$$

Behavior	Condition	Angle Threshold
Jackknifing	$\alpha_{12} > 0.8, \alpha_{22} < 0.4$	$\lvert\theta\rvert > 60°$
Trailer Swing	$\alpha_{12} < 0.4, \alpha_{22} > 0.8$	$\lvert\theta\rvert \geq 30°$

CBF SAFETY

Unsafe behavior is triggered only when all of a set of three conditions are simultaneously satisfied.
MATLAB's $fmincon$ solver does not support logical "AND" constraints directly, we approximate this behavior using a smooth soft-minimum

$$h(x_k) = \frac{1}{\kappa}\log\left(\sum_{i=1}^{m} \exp\big(-\kappa g_i(x_k)\big)\right),$$

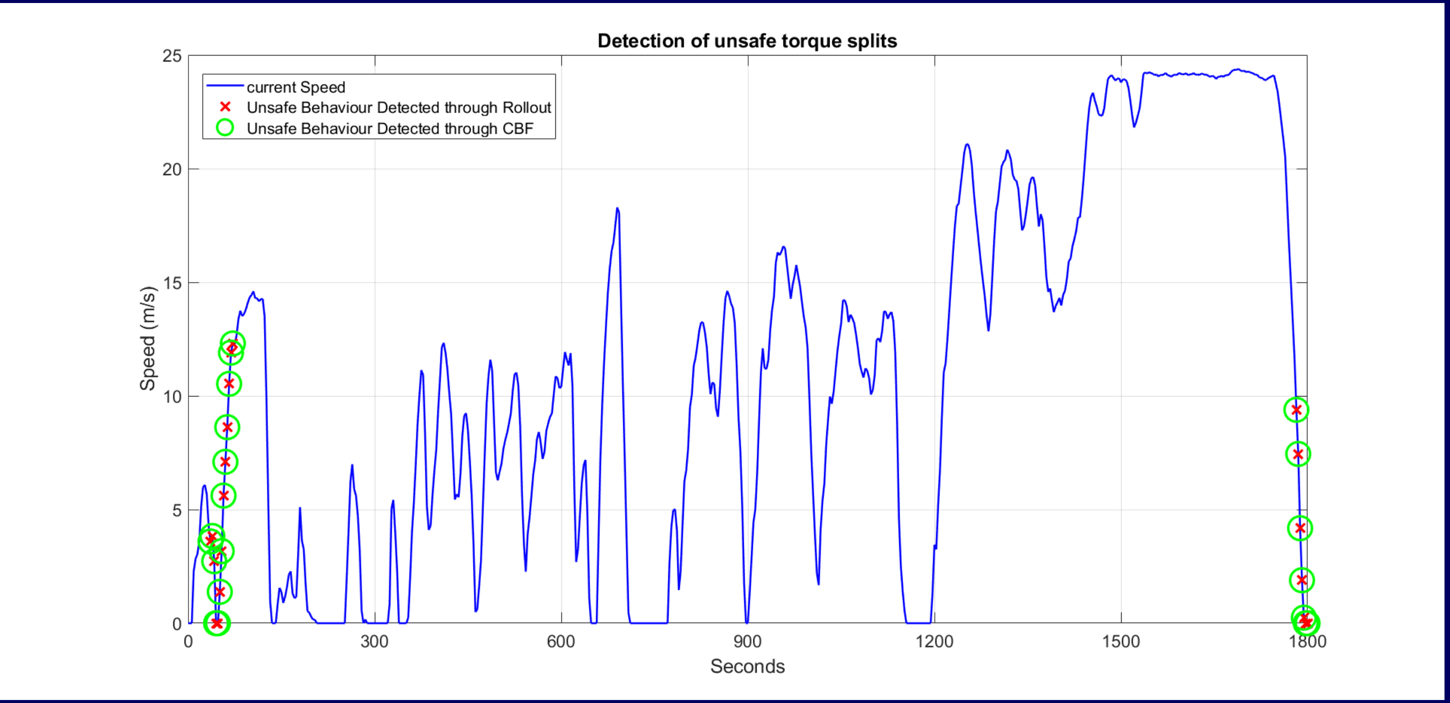
Considering the jack-knifing situation:

$$g_1(x_k) = \alpha_{12} - 0.8, \quad g_2(x_k) = -\alpha_{22} + 0.4, \quad g_3(x_k) = \theta_1 - \frac{\pi}{3}$$

That leads to the following CBF function

$$h^A(x_k) = \frac{1}{\kappa}\log(\exp(-\kappa g_1) + \exp(-\kappa g_2) + \exp(-\kappa g_3))$$

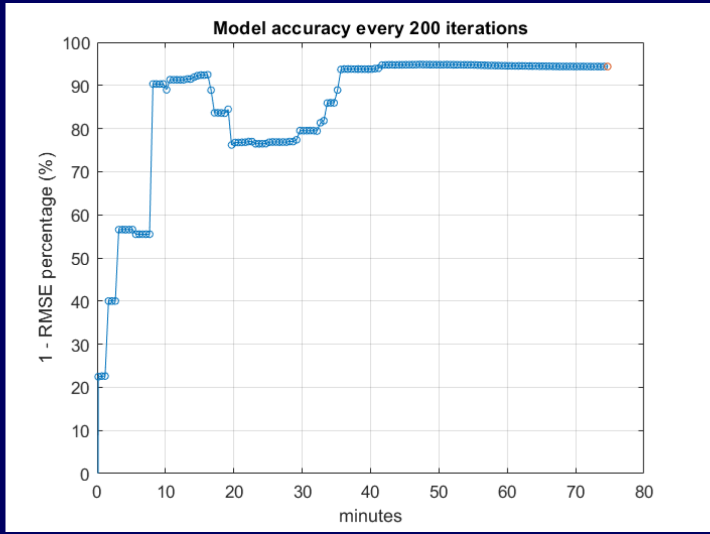
STRESS TEST FOR THE 2 METHODS



Stress test prove that both detect the same unsafe modes during the critical scenario

FORGETTING FACTOR

$\lambda \approx 1$



$\lambda = 0.95$