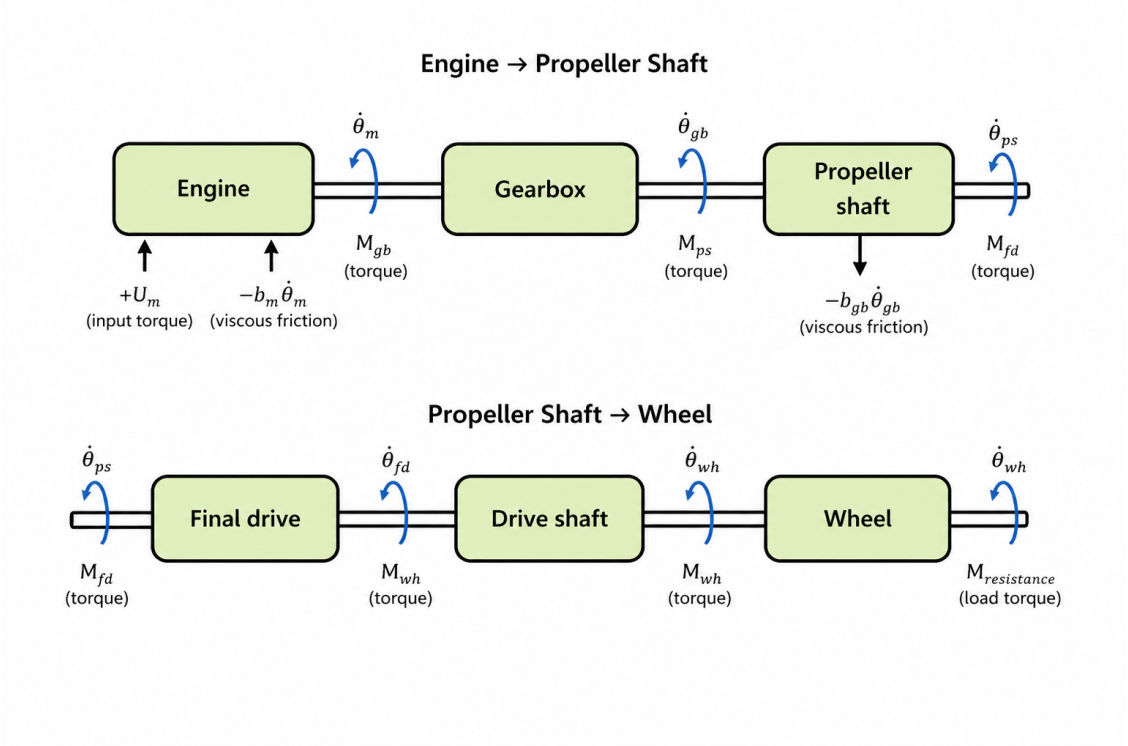
CHALMERS
UNIVERSITY OF TECHNOLOGY



Robust Vehicle Acceleration Control

Model-Based Control for Robust Longitudinal Acceleration Tracking

Master's thesis in Systems, Control and Mechatronics

ALEXANDER KONGSTAD
FABIAN KARLSSON

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Robust Vehicle Acceleration Control

ALEXANDER KONGSTAD
FABIAN KARLSSON



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
Gothenburg, Sweden 2026

Robust Vehicle Acceleration Control
ALEXANDER KONGSTAD
FABIAN KARLSSON

© ALEXANDER KONGSTAD & FABIAN KARLSSON, 2026.

Examiner: Jonas Fredriksson, Department of Electrical Engineering

Master's Thesis 2026
Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Schematic diagram of the driveline components and their interactions. Read more in chapter 2.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Abstract

This thesis investigates robust acceleration control for heavy-duty vehicles to improve longitudinal control and driver-assist functions. Despite advances in longitudinal vehicle control, important limitations remain. Existing control strategies may be sensitive to measurement delays, perform poorly for small acceleration errors, and insufficiently reject oscillatory disturbances from driveline windup and tire dynamics. These challenges motivate a model-based control approach that explicitly accounts for the relevant system dynamics to improve tracking performance and disturbance rejection.

A third-order state-space driveline model was developed to capture the system's dominant dynamics, including flexible driveshaft behavior modeled as a spring-damper system. The model was tuned and validated against real-world measurements from a battery electric vehicle (BEV), demonstrating close agreement in both speed and acceleration responses across multiple driving scenarios and gear configurations.

A Smith predictor was subsequently designed and evaluated to compensate for the system's inherent dead-time delays. Simulation testing showed that the Smith predictor maintains stability under model and delay mismatches of up to $\pm 50\%$, while also enabling higher controller gains and faster response times. To further improve performance, the Smith predictor was extended to include measurement updates.

An LQG controller with integral action, focusing on robustness properties, was proposed to address driveline oscillation damping and steady-state tracking accuracy. Simulation results show that, given an accurate model, the proposed controller reduces oscillations and improvesmproved steady-state tracking compared to the current controller.

Using logged vehicle data, the simulation validated the proposed controllers, showing that model-based control strategies can improve acceleration-tracking performance and robustness compared to the existing controller.

Acknowledgements

We would like to sincerely thank our supervisor and examiner, Jonas Fredriksson at the Department of Electrical Engineering, Chalmers University of Technology, for his guidance, support, and valuable feedback throughout the course of this thesis.

We would also like to thank Volvo Group, and especially the VMP team, for their support and for giving us the opportunity to carry out our thesis work within the team. We greatly appreciate their help, expertise, and willingness to share their knowledge throughout the project.

Fabian Karlsson & Alexander Kongstad, Gothenburg, August 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

BEV	Battery Electric Vehicle
ECU	Electrical Control Unit
GR	Gear Ratio
ICE	Internal Combustion Engine
PI-FF	PI controller with feedforward action
PI-FF-SP	PI controller with feedforward action + Smith Predictor
PI-FF-MUSP	PI controller with feedforward action + Measurement-Updated Smith Predictor
ILQG	LQG controller with integral action
LTR	Loop Transfer Recover
SISO	Single Input Single Output
MISO	Multiple Input Single Output

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

$_k$	Index for time step in discrete time

Driveline Parameters

b_m	Engine viscous friction coefficient
b_{gb}	Gearbox viscous friction coefficient
b_{wh}	Wheel viscous friction coefficient
i_{gb}	Gearbox ratio
i_{fd}	Final drive ratio
J_m	Engine moment of inertia
J_{gb}	Gearbox moment of inertia
J_{fd}	Final drive moment of inertia
$m_{vehicle}$	Vehicle mass
k	stiffness coefficient
c	damping coefficient
r_{wh}	Wheel radius
C_a	Drag coefficient
S_a	Cab surface
ρ_a	Air density
g	Gravity acceleration
C_{rr}	Rolling resistance
T_s	Timestep

Driveline variables

M_{gb}	Gearbox torque
M_{ps}	Propeller shaft torque
M_{fd}	Final drive torque
M_{ds}	Drive shaft torque
M_{wh}	Wheel drive torque
$M_{resistance}$	External load torque
$\dot{\theta}_m$	Angular engine speed
$\dot{\theta}_{gb}$	Angular gearbox speed
$\dot{\theta}_{ps}$	Angular propeller shaft engine speed
$\dot{\theta}_{fd}$	Angular final drive speed
$\dot{\theta}_{fd}$	Angular wheel speed
U_m	Engine input torque
$v_{vehicle}$	Vehicle speed in m/s
F_{aero}	Air drag force
F_{roll}	Roll resistance force
α_{road}	Road angle in radians
p	Road angle in %

State pace variables

x	State vector
$\hat{x}$	Predicted state vector
x_1	State one, $= (\frac{\theta_m}{i_{fd} i_{gb}} - \theta_{wh})$
x_2	State two, $=\dot{\theta}_m$
x_3	State three, $=\dot{\theta}_{wh}$
y	output, $=v_{vehicle}$
u	input, $=U_m+$ applied service break force
l	external load

State pace Matrices

A	Continuous system matrix
A_d	Discrete system matrix

B	Continuous input matrix
B_d	Discrete input matrix
H	Continuous load matrix
H_d	Discrete load matrix
C	Continuous output matrix
C_d	Discrete output matrix

Transfer function Variables

G	Actual plant transfer function
G	Model plant transfer function
Z	Controller transfer function
T_F	Closed loop transfer function

Kalman Variables

L	Kalman gain
P	Solution of the algebraic Riccati equation (Kalman)
v_1	Process noise term
v_2	Measurement noise term
R_1	Process noise intensity
R_2	measurement noise intensity
N	Process noise input matrix
$C_{d,kalman}$	Output matrix (Kalman)
$C_{d,engspeed}$	Engine output matrix

ILQG Variables

K_x	LQ state gain matrix
K_i	LQ integral gain matrix
K_{aug}	LQ Augmented gain matrix
Q	State quadratic weighting matrix
R	Input quadratic weighting matrix
R	Input quadratic weighting matrix

w_2	Measurement noise term (ILQG)
r	Reference
P_{aug}	Solution of the algebraic Riccati equation (ILQG)
x_{aug}	Augmented state vector
x_I	Integral state

ILQG State-Space Matrices

A_{aug}	Augmented system matrix
B_{aug}	Augmented input matrix
C_{aug}	Augmented output matrix
A_c	Controller system matrix
B_c	Controller measurement input matrix
B_r	Controller reference input matrix
C_c	Controller output matrix
D_c	Controller feedthrough matrix

Contents

List of Acronyms viii

Nomenclature xi

List of Figures xvii

List of Tables xix

1 Introduction and Background 1
- 1.1 Objective and aim . . . 1
- 1.2 Research questions . . . 2
- 1.3 Structure of work . . . 2
- 1.4 Limitations . . . 2
- 1.5 Ethical and sustainability aspects . . . 2

2 Driveline modeling 3
- 2.1 Driveline equations . . . 3
  - 2.1.1 Engine . . . 4
  - 2.1.2 Gearbox . . . 4
  - 2.1.3 Propeller shaft . . . 5
  - 2.1.4 Final drive . . . 5
  - 2.1.5 Drive shaft . . . 6
  - 2.1.6 Wheel . . . 6
- 2.2 State space model . . . 7
- 2.3 Model tuning and validation . . . 8
  - 2.3.1 Model performance between different gears . . . 9

3 Analysis 15
- 3.1 Bode . . . 15
- 3.2 Nyquist . . . 16
- 3.3 Step response . . . 16
- 3.4 Plant . . . 16
- 3.5 Current Controller . . . 17
- 3.6 Delay estimation . . . 20

4 Smith predictor **23**
4.1 Robustness to model and delay mismatch . . . 24
4.2 Model mismatch . . . 25
4.3 Delay mismatch . . . 25
4.4 Combined mismatch . . . 26
4.5 Controller tuning with Smith prediction . . . 27
4.6 Validation . . . 27
4.7 Measurement-updated Smith predictor . . . 28

5 LQG and integration control **31**
5.1 Motivation for LQG Control . . . 31
5.1.1 State Estimation and LQG Control . . . 31
5.1.2 Robustness Considerations . . . 32
5.1.3 Integral Action . . . 32
5.1.4 Practical Considerations . . . 32
5.2 Kalman Filter . . . 33
5.3 Gains . . . 33
5.4 Controller structure . . . 35
5.5 Stability and Robustness Tuning . . . 35
5.5.1 Open-Loop Transfer Function and Margin Analysis . . . 36
5.5.2 Tuning for robustness . . . 37

6 Results **39**
6.1 Research Question 1 . . . 39
6.1.1 Model . . . 39
6.2 Research Question 2 . . . 40
6.2.1 Smith predictor & measurement-updated Smith predictor . . . 44
6.2.2 LQG . . . 47
6.2.3 Model mismatch . . . 49

7 Discussion **51**
7.1 Model . . . 51
7.2 Smith predictor . . . 52
7.3 LQG . . . 53
7.4 Comparison of Control Strategies . . . 54
7.4.1 PI-FF-SP . . . 55
7.4.2 PI-FF-MUSP . . . 55
7.4.3 ILQG . . . 55
7.4.4 Recommended Control Strategy . . . 56

8 Conclusion **57**

List of Figures

2.1 Schematic diagram of the driveline components and their interactions. 4
2.2 Simulink representation of state space model 9
2.3 Comparison between measured and simulated data. Gear shifts are present, and the model does not adapt to them. 11
2.4 Comparison between measured and simulated data. Gear shifts are present, and the model adapts to them. 12
2.5 Comparison between simulated and measured acceleration. Here, the model adapts to the gear shift, keeps a constant acceleration during a gear, and maintains constant acceleration during the shift. 13

3.1 Bode plot of plant model 17
3.2 Open-loop Bode plot for different gear ratios 17
3.3 Nyquist plot 18
3.4 Closed-loop step response 19
3.5 Step-response with feedforward 20
3.6 Delay estimation. The dotted red line shows $\tau = 0.36$, indicating an estimated time delay of 0.36s. 20

4.1 System without delay 23
4.2 Smith predictor block diagram 24
4.3 Comparison of model mismatch 25
4.4 Comparison of delay mismatch 26
4.5 Comparison of worst cases with model and delay mismatch 26
4.6 Comparison of different PI tunes 27
4.7 Step response of PI-FF-SP during increasing gain 28
4.8 Simple example of how the measurement-updated Smith can maintain a more accurate performance compared to the faulty model not updated with measurements. The horizontal axis represents time, while the vertical axis represents the model output. 29
4.9 Block diagram of measurement-updated Smith predictor structure. . 30

5.1 Block diagram of ILQG controller structure 35
5.2 LQG controller tuned for accuracy and fast settling time 37
5.3 LQG tuning with respect to robustness 38

6.1 Comparison between measured and simulated data during driving opportunity 1. 41
6.2 Comparison between measured and simulated data during driving opportunity 2 42
6.3 Comparison between measured and simulated data during driving opportunity 3 43
6.4 Comparison of PI-FF vs PI-FF-SP 44
6.5 In-vehicle testing: PI-FF vs PI-FF-SP Smith predictor 45
6.6 In-vehicle testing: PI-FF vs PI-FF-MUSP Measurement-updated Smith predictor 46
6.7 Step response tracking comparison between PI-FF and ILQG controllers 48
6.8 Reference tracking comparison between PI-FF and ILQG controllers . 48
6.9 Step response with altered model parameters to demonstrate model mismatch. The parameters are faulty by a 10 percent increase and decrease. 49

List of Tables

3.1 Stability for different gear ratios 18
3.2 Open-loop poles 19
3.3 Open-loop zeros 19

5.1 Performance measurements for different controller tuning 38
5.2 T_{max} for different controller tuning 38

6.1 Cross-correlation-based comparison of the standard and measurement-updated Smith predictors. 46
6.2 Step response performance measurements for PI-FF and ILQG 48
6.3 Root mean square error for PI-FF and LGQ+I 49

1

Introduction and Background

As the market for heavy-duty vehicles continues to evolve, the importance of advanced control and driver-assist functions has increased. This is particularly evident in the high-end automotive market, where these functionalities are essential to remain competitive. Robust, precise acceleration control is therefore fundamental to enabling safe, accurate operation of driver-assist and control systems.

Most longitudinal vehicle control systems rely directly or indirectly on accurate acceleration control. Inaccurate acceleration control can lead to several issues, including oscillatory behavior and incorrect speed estimation. Currently, PID controllers are used, and despite their simple design, they achieve a lot. Unfortunately, they have weaknesses that need to be addressed. These include sensitivity to measurement delays, insufficient handling of small acceleration errors, and limited robustness to unmodeled disturbances from driveline windup and tire dynamics.

Over the years, researchers have proposed numerous control structures to improve accuracy, robustness, and damping of oscillatory behavior. Many of these approaches are model-based. However, industrial adoption of these methods has been relatively slow, as industry tends to favor simpler, more interpretable control strategies. As a result, PI and PID controllers remain widely used due to their simplicity, robustness, and adequate performance in many applications.

1.1 Objective and aim

This thesis investigates and analyzes the strengths and weaknesses of a state-of-the-art acceleration controller, then evaluates its robustness and stability to identify potential solutions for improving performance.

The objective is to investigate how model-based control can improve longitudinal acceleration control in heavy-duty vehicles, with particular focus on small acceleration errors, unmodeled disturbances, and robustness to measurement noise and delays. The study also aims to identify which drivetrain dynamics are essential to capture in order to achieve a sufficiently accurate yet practical model.

1.2 Research questions

The following research questions guide and focus the work.

Research Question 1. What driveline factors and relationships must be considered to create a model that accurately captures vehicle dynamics, including oscillations stemming from driveline windup?

Research Question 2. What impact does introducing model-based control have on acceleration-tracking performance and robustness compared to the current solution?

1.3 Structure of work

The work is divided into several parts. First, Chapter 2 develops a model of the vehicle driveline, which serves as the foundation for the subsequent chapters. Chapter 3 then uses this model to analyze the current system's properties and limitations. Based on this analysis, two model-based solutions are proposed. The first solution, presented in Chapter 4, uses a Smith predictor to supplement the existing controller and improve performance. The second solution, presented in Chapter 5, introduces a new control strategy based on an LQG controller with integral action. These two approaches are investigated to determine whether the weaknesses identified during the system analysis can be mitigated through an alternative, model-based control design philosophy. Chapter 6 presents the evaluation of the developed model and controllers, while Chapter 7 discusses the results and their implications.

1.4 Limitations

To limit the scope of the project, no hardware modifications are performed. Instead, the focus is entirely on implementing a software-based solution that uses current sensors and signals already used in the vehicles. Moreover, the controller design is expected to be the same across all driveline configurations but may differ in tuning between ICE and BEV units due to the significant difference in actuator response. The developed algorithm should also be lightweight enough to fit within the given ECU limits.

1.5 Ethical and sustainability aspects

The thesis aims to improve driveline acceleration control, thereby reducing energy use and, in turn, mitigating negative environmental impacts such as emissions. The thesis also aims to develop a controller that works for both BEV and ICE drive lines without any external or new parts, which is great from a sustainability standpoint, given the costs of materials and production line changes when introducing new parts.

2
Driveline modeling

To capture the relationship between torque demand from an acceleration controller and the resulting wheel speed, a driveline model is required. With a driveline model, it is possible to investigate how different scenarios and controller configurations affect system behavior through simulation. It is also possible to utilize a model of the driveline for model-based control, which, as shown in [1], can be the optimal control solution under the right circumstances.

The aim of the control task is to be precise and handle disturbances, therefore, the driveline model not only has to capture the relationship between torque demand and resulting wheel speed but also describe disturbances such as oscillations occurring in the driveline. Another key aspect is the model's simplicity and modularity. The controller design cannot change between different driveline configurations, only tuning can be done. Therefore, the model is modular and simple enough to allow simple and easy tuning across different driveline configurations.

In [2], a third-order model for a rear-driven heavy truck driveline is proposed, consisting of engine, clutch, transmission, propeller shaft, final drive, drive shafts, and wheels. In this model, oscillatory disturbances are modeled as a spring-damper system in the driveshaft. Further, in [3] it is shown and concluded that a third-order model is sufficient to capture the main characteristics of a driveline, suggesting that the driveline oscillations originate mainly from the driveshafts due to the driveshaft being exposed to the largest torque in relation to stiffness and will therefore generate the lowest frequency of oscillation. Given the project's aim of capturing the vehicle's dynamics, including possible driveline oscillations, the model proposed by [2] is chosen as the template for the model developed in this project.

2.1 Driveline equations

The driveline equations are heavily inspired by the third-order model presented in [2]. However, unlike [2], this project models the clutch as stiff. This is mainly to reduce model complexity while maintaining satisfactory performance, but also because real-world testing and simulation validation will be conducted primarily using a BEV, which does not use a clutch as a standard ICE driveline configuration does. The propeller shaft is also modeled as stiff, following [2], which states that the dominant driveline oscillations originate primarily from the drive shaft, indicating that the propeller shaft can be modeled as stiff to reduce model complexity.

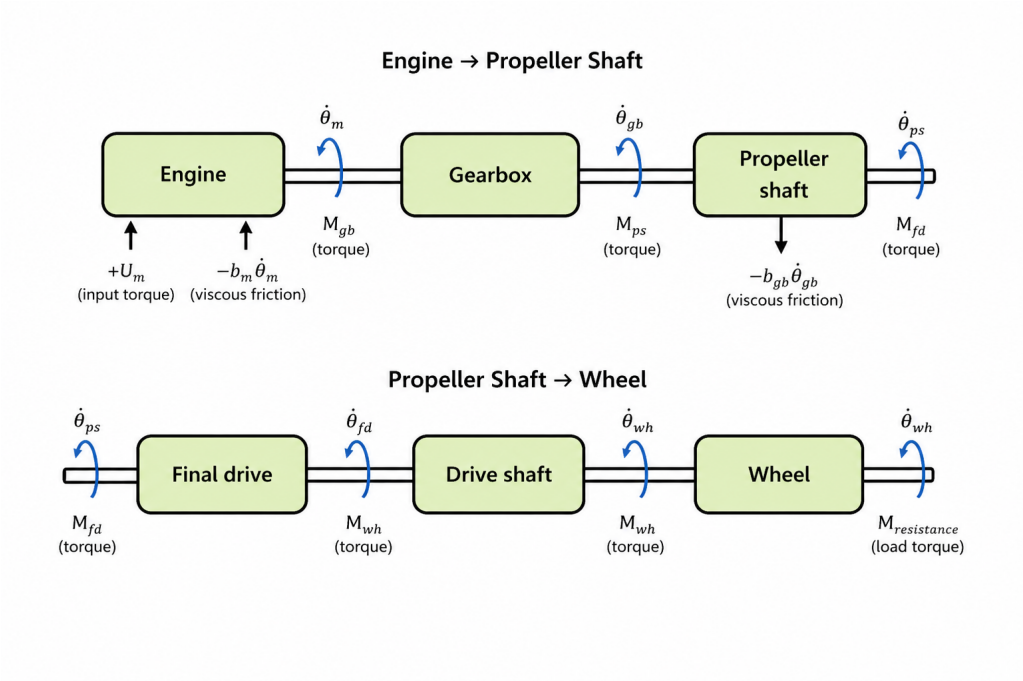


Figure 2.1: Schematic diagram of the driveline components and their interactions.

The project aims to improve acceleration tracking, and it is therefore intuitive to choose vehicle acceleration as the model output. However, in this project, the model output is vehicle speed. This is because the current real vehicle system front-axle speed measurement is the most accurate available, and acceleration is derived from the vehicle speed between sampling periods.

In the model, the upstream driveline is assumed to be rigid from the transmission output shaft to the engine, meaning no torsional deformation between engine components and the transmission output shaft. Using the driveline equations, Newton's second law is used to derive the torque–angular acceleration relationship. Figure 2.1 shows an overview of the model components.

2.1.1 Engine

For the engine, U_m denotes the commanded input torque, which may originate from either a combustion or an electric power unit. The engine is characterized by a viscous friction coefficient b_m, a moment of inertia J_m, and an output shaft angle θ_m. The rotational dynamics of the engine are described by

$$J_m\ddot{\theta}_m = U_m - b_m\dot{\theta}_m - M_{gb}, \tag{2.1}$$

where M_{gb} represents the torque transmitted to the gearbox.

2.1.2 Gearbox

The gearbox ratio at the output shaft is denoted by i_{gb}, which varies with the selected gear. Like the engine, the gearbox is modeled with an equivalent inertia J_{gb}

and subject to viscous friction characterized by the coefficient b_{gb}. The rotational dynamics of the gearbox are therefore described by

$$J_{gb}\ddot{\theta}_{gb} = M_{gb}i_{gb} - b_{gb}\dot{\theta}_{gb} - M_{ps}, \tag{2.2}$$

where M_{ps} denotes the torque transmitted to the downstream driveline. The kinematic relationship between the engine and gearbox shafts is given by

$$\theta_m = \theta_{gb}i_{gb}. \tag{2.3}$$

Using this relation, we can reformulate the gearbox dynamics in (2.2) in terms of the engine angular variables. This yields

$$J_{gb}\ddot{\theta}_m = M_{gb}i_{gb}^2 - b_{gb}\dot{\theta}_m - M_{ps}i_gb, \tag{2.4}$$

which explicitly expresses the gearbox dynamics as a function of the engine angular speed.

2.1.3 Propeller shaft

As mentioned, the propeller shaft is modeled as stiff and therefore carries only angle and torque.

$$\begin{aligned} M_{ps} &= M_{fd} \\ \theta_{gb} &= \theta_{ps} \end{aligned} \tag{2.5}$$

2.1.4 Final drive

The final drive is modeled as the engine and gearbox, with an inertia J_{fd}.

$$J_{fd}\ddot{\theta}_{fd} = M_{fd}i_{fd} - M_{ds} \tag{2.6}$$

This can then be expanded to express it in terms of angular engine speed by using the relation between the angles.

$$\theta_{ps} = \theta_{fd}i_{fd} = \theta_{gb} \tag{2.7}$$

$$J_{fd}\ddot{\theta}_{gb} = M_{ps}i_{fd}^2 - M_{ds}i_{fd} \tag{2.8}$$

$$J_{fd}\ddot{\theta}_m = M_{ps}i_{fd}^2 i_{gb} - M_{ds}i_{fd}i_{gb} \tag{2.9}$$

Finally from (2.2) it is possible to substitute M_{ps}

$$J_{fd}\ddot{\theta}_m = (M_{gb}i_{gb}^2 - J_{gb}\ddot{\theta}_m - b_{gb}\dot{\theta}_m)i_{fd}^2 - M_{ds}i_{fd}i_{gb}, \tag{2.10}$$

resulting in a final drive model based solely on engine angular speed:

$$(J_{gb}i_{fd}^2 + J_{fd})\ddot{\theta}_m = M_{gb}i_{gb}^2 i_{fd}^2 - b_{gb}\dot{\theta}_m i_{fd}^2 - M_{ds}i_{fd}i_{gb} \tag{2.11}$$

2.1.5 Drive shaft

The drive shaft is modeled as flexible. Here, it is represented by a spring-damper system with stiffness k and damping c:

$$M_{wh} = M_{ds} = k(\theta_{fd} - \theta_{wh}) + c(\dot{\theta}_{fd} - \dot{\theta}_{wh}) = k(\frac{\theta_m}{i_{fd}i_{gb}} - \theta_{wh}) + c(\frac{\dot{\theta}_m}{i_{fd}i_{gb}} - \dot{\theta}_{wh}) \quad (2.12)$$

Substituting M_{ds} in (2.11), gives

$$(J_{gb}i_{fd}^2 + J_{fd})\ddot{\theta}_m = M_{gb}i_{gb}^2 i_{fd}^2 - b_{gb}\dot{\theta}_m i_{fd}^2 - k(\frac{\theta_m}{i_{fd}i_{gb}} - \theta_{wh})i_{fd}i_{gb} - c(\frac{\dot{\theta}_m}{i_{fd}i_{gb}} - \dot{\theta}_{wh})i_{fd}i_{gb}. \quad (2.13)$$

Inserting (2.13) in (2.1) gives:

$$J_m\ddot{\theta}_m = U_m - b_m\dot{\theta}_m - (\frac{J_{gb}}{i_{gb}^2} + \frac{J_{fd}}{i_{gb}^2 i_{fd}^2})\ddot{\theta}_m + \frac{b_{bg}\dot{\theta}_m}{i_{gb}^2} + \frac{k}{i_{gb}i_{fd}}(\frac{\theta_m}{i_{fd}i_{gb}} - \theta_{wh}) + \frac{c}{i_{gb}i_{fd}}(\frac{\dot{\theta}_m}{i_{fd}i_{gb}} - \dot{\theta}_{wh}), \quad (2.14)$$

which describes the engine angular acceleration as a function of the engine speed, the angular displacement between the engine and the wheel, and their relative angular speed.

2.1.6 Wheel

The forces acting on the wheels are described by:

$$(J_{wh} + m_{vehicle}r_{wh}^2)\ddot{\theta}_{wh} = M_{wh} - b_{wh}\dot{\theta}_{wh} - M_{resistance} \quad (2.15)$$

where the friction torque is represented by a viscous friction coefficient b_{wh} and the external load torque applied to the vehicle is represented by $M_{resistance}$. The load torque consists of rolling resistance force, air drag force, and the gravitational force acting on the vehicle on the same axle as the forces pushing the vehicle in its driving direction

$$M_{resistance} = (F_{roll} + F_{aero} + m_{vehicle}gsin(\alpha_{road}))r_{wh}, \quad (2.16)$$

where forces F_{roll} and F_{aero} are calculated using

$$\begin{aligned} F_{aero} &= \frac{1}{2}C_a S_a \rho_a v_{vehicle}^2 \\ F_{roll} &= m_{vehicle}gC_{rr}cos(\alpha_{road}) \end{aligned} \quad (2.17)$$

where C_a denotes the aerodynamic drag coefficient, S_a is the frontal area of the cab, ρ_a is the air density, g is the gravitational acceleration, α_{road} is the road inclination angle, and C_{rr} denotes the rolling resistance coefficient. The road inclination angle is given as a percentage slope, so it has to be converted by

$$\alpha_{road} = \arctan(\frac{p}{100}) \quad (2.18)$$

where p is the slope in %.

Modeling M_{wh} as in (2.12), it is possible to finalize and describe the angular wheel acceleration as a function of the angular displacement between the engine and the wheel, and their relative angular speed, as well as the angular wheel speed:

$$(J_{wh}+m_{vehicle}r_{wh}^2)\ddot{\theta}_{wh} = k(\frac{\theta_m}{i_{fd}i_{gb}}-\theta_{wh})+c(\frac{\dot{\theta}_m}{i_{fd}i_{gb}}-\dot{\theta}_{wh})-b_{wh}\dot{\theta}_{wh}-M_{resistance}. \quad (2.19)$$

2.2 State space model

To apply model-based control and simulations to the vehicle model, a state-space model is formed. This is done by combining (2.14) and (2.19) to obtain a system of equations that describes the vehicle dynamics.

$$\begin{aligned}(J_m + \frac{J_{gb}}{i_{gb}^2} + \frac{J_{fd}}{i_{gb}^2 i_{fd}^2})\ddot{\theta}_m &= U_m - b_m\dot{\theta}_m + \frac{b_{bg}\dot{\theta}_m}{i_{gb}^2} + \frac{k}{i_{gb}i_{fd}}(\frac{\theta_m}{i_{fd}i_{gb}} - \theta_{wh}) + \frac{c}{i_{gb}i_{fd}}(\frac{\dot{\theta}_m}{i_{fd}i_{gb}} - \dot{\theta}_{wh}) \\ (J_{wh} + m_{vehicle}r_{wh}^2)\ddot{\theta}_{wh} &= k(\frac{\theta_m}{i_{fd}i_{gb}} - \theta_{wh}) + c(\frac{\dot{\theta}_m}{i_{fd}i_{gb}} - \dot{\theta}_{wh}) - b_{wh}\dot{\theta}_{wh} - M_{resistance}\end{aligned} \quad (2.20)$$

As the objective is to control wheel speed using engine speed and understanding the engine oscillation, it is intuitive to choose states as $x_1 = (\frac{\theta_m}{i_{fd}i_{gb}} - \theta_{wh})$ for driveline torsion, $x_2 = \dot{\theta}_m$ for angular engine speed and $x_3 = \dot{\theta}_{wh}$ for angular wheel speed. The load torque $M_{resistance}$ is possible to estimate from measured values of slope, speed, and vehicle mass. Thus, the load torque can be modeled as a disturbance input l with a disturbance input matrix H. The final state space model with load torque l and input torque u can be described by

$$\dot{x} = Ax + Bu + Hl \quad (2.21)$$

where

$$\begin{aligned}A &= \begin{bmatrix} 0 & \frac{1}{i} & -1 \\ -\frac{k}{iJ_1} & -\frac{b_1+\frac{c}{i^2}}{J_1} & \frac{c}{iJ_1} \\ \frac{k}{J_2} & \frac{c}{iJ_2} & -\frac{c+b_2}{J_2} \end{bmatrix} \\ B &= \begin{bmatrix} 0 \\ \frac{1}{J_1} \\ 0 \end{bmatrix} \\ H &= \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{J_2} \end{bmatrix}.\end{aligned} \quad (2.22)$$

In the state space model, $J_1,\ J_2,\ b_1,\ b_2,\ i$ are described as

$$\begin{aligned} J_1 &= J_m + \frac{J_{gb}}{i_{gb}^2} + \frac{J_{fd}}{i_{gb}^2 + i_{fd}^2} \\ J_2 &= J_{wh} + m_{vehicle} r_{wh}^2 \\ b_1 &= -b_m + \frac{b_{gb}}{i_{gb}^2} \\ b_2 &= b_{wh} \\ i &= i_{gb} i_{fd} \end{aligned} \tag{2.23}$$

.
The vehicle speed is the output signal from the model:

$$y = Cx = \begin{bmatrix} 0 & 0 & r_{wh} \end{bmatrix} x \tag{2.24}$$

Because all measurements in this project are sampled in discrete time, the continuous-time model, (2.21), is discretized using the zero-order hold method, assuming each sampled value is piecewise constant over each sample interval. The zero-order hold method used for this project is explained in [4]. The resulting discretized model is described as

$$\begin{aligned} x_{k+1} &= A_d x_k + B_d u_k + H_d l \\ y_k &= C_d x_k, \end{aligned} \tag{2.25}$$

with time step k and discrete state space matrices A_d, B_d and C_d.

2.3 Model tuning and validation

According to [1], a model cannot replicate reality exactly. However, it is possible to tune the model parameters, making the model behave as close to reality as possible. The tuning is done by simulating the model behavior in different scenarios and validating the result using real-world measurements from an actual vehicle. The vehicle measurements used as validation measurements are the recorded vehicle speed and engine speed. As mentioned, there are no measurements of acceleration available, the acceleration comparison requires the derivative of both the model-predicted speed and measured speed. The acceleration is the most important comparison during validation, as the final goal is to control it.

Although the model should be adaptable across driveline configurations, time and testing resources limit validation to a single driveline configuration, a Battery Electric Vehicle (BEV). To facilitate tuning, the validation measurements are collected during structured tests in which the vehicle starts from a standstill and then performs generic driving on regular roads. This ensures that all initial states start at zero for both the vehicle and the model simulation. For simplicity, the tuning is performed on tests in which only one gear is active, and no clutch action occurs. It is later possible to adapt the model across different gears.

The model parameters are calibrated through manual tuning and numerical optimization. Specifically, parameter adjustments are performed iteratively to improve the agreement between simulation outputs and measured data, complemented by the use of the MATLAB optimization function `fmincon`, which minimizes the root mean square error (RMSE) between simulated and experimental signals. However, relying solely on numerical optimization may lead to overfitting, whereby the model reproduces measurement noise and delays rather than capturing the underlying system dynamics. Since experimental data inherently contain noise and temporal delays, it is essential to avoid embedding these artifacts into the model structure. Therefore, initial estimates of the variable parameters J_1, J_2, b_1, b_2, i, c, k, and r_{wh} are obtained from specification sheets or vehicle data, serving as physically grounded starting points. Subsequent refinement is then performed through controlled tuning to ensure both fidelity to measured behavior and robustness of the model. The Simulink model used for tuning is represented in Figure 2.2

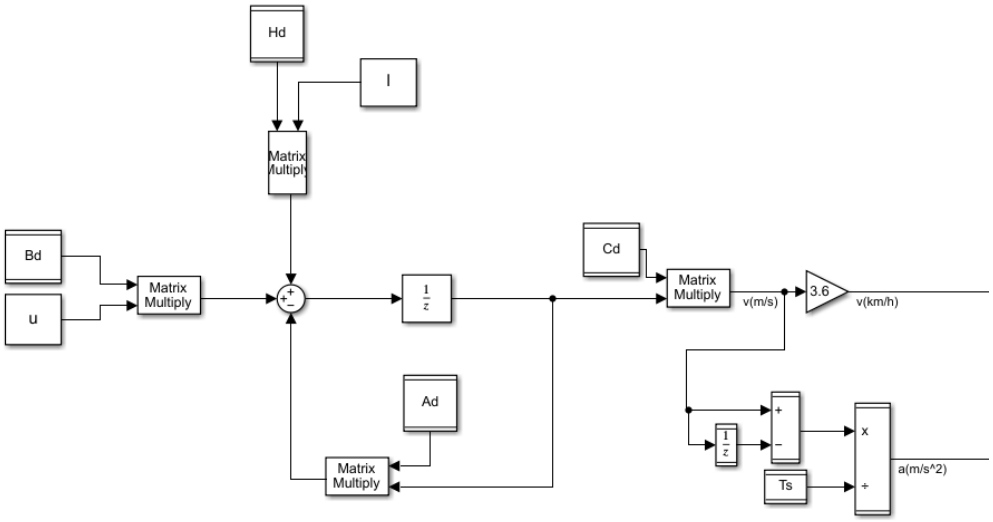
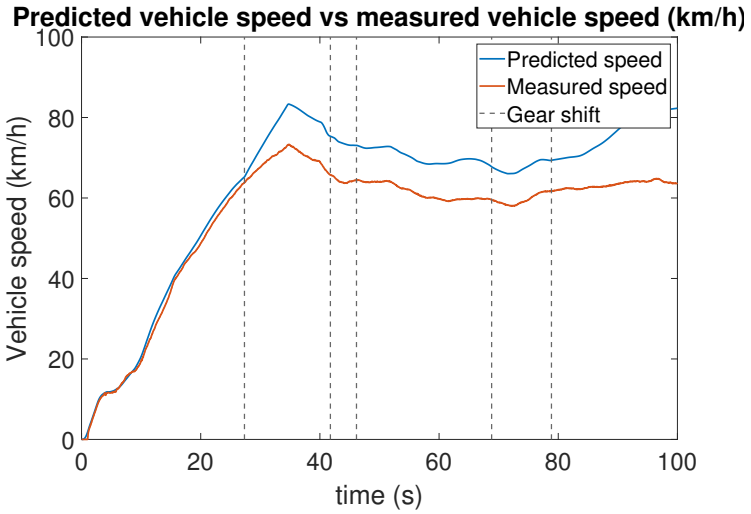


Figure 2.2: Simulink representation of state space model

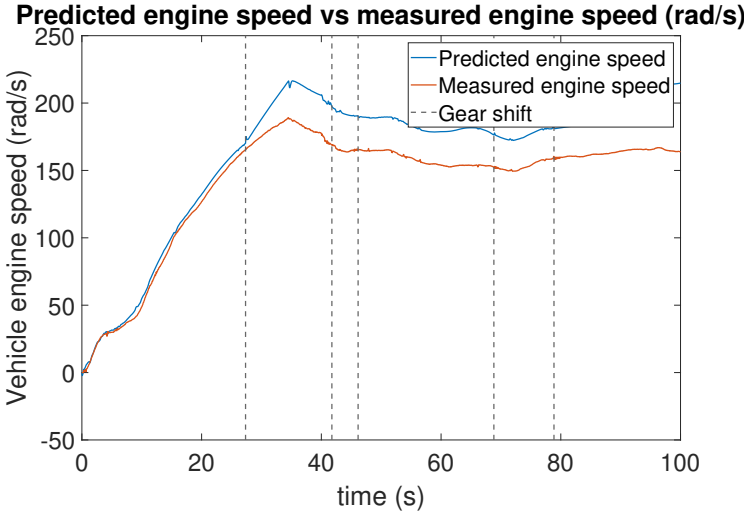
2.3.1 Model performance between different gears

In practice, it is not possible to assume the same gear will always be selected, and gear shifting will occur during driving. Therefore, it is important that the model is valid throughout different gears, and simulations are thus performed on scenarios where gear shifting occurs. An example is shown in Figure 2.3, where gear shifts appear at the dashed lines. As expected, the geargear shifts affect the systems, causing the model to deviate, especially during the first gear shift. To solve this, the gear ratio i in the state space model presented in equation 2.22 is treated as a variable dependent on the current gear. Figure 2.4 shows a simulation in which the gear ratio i varies with the current gear, rather than remaining constant throughout the gear shifts. Here, the immediate deviation at gear shift no longer occurs, and the model performs as expected. However, in the acceleration comparison in Figure

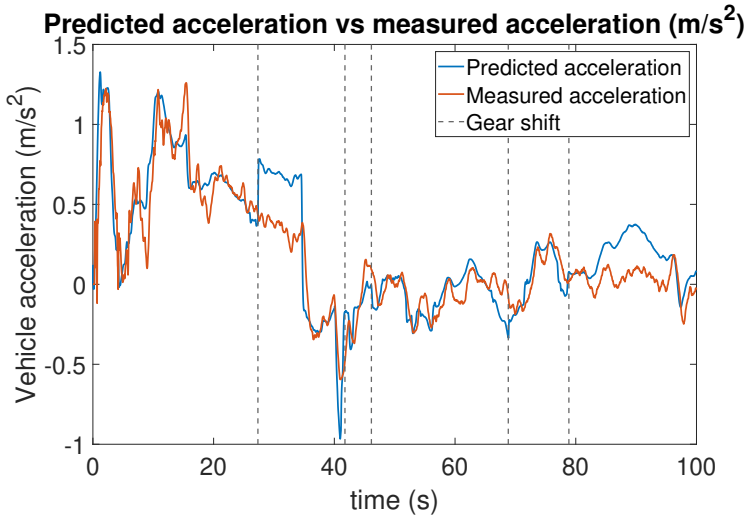
2.4, large jerks occur during gear shifts because the model lacks clutch dynamics. These jerks are highly undesirable because they produce large errors in the estimated acceleration, which can degrade the performance of a model-based controller that relies on the model. To address this, this project proposes a simple solution: if a gear shift occurs, the acceleration remains constant during this period, as shown in Figure 2.5. Gear-switch detection is based on the current gearbox state reported by the vehicle ECU. Note that varying i with the current gear creates different state-space systems for each gear, which can affect model properties such as stability. Therefore, this must be taken into account during further controller design and analysis.



(a) Vehicle speed [km/h]. Comparison of measured vehicle speed and simulated vehicle speed.

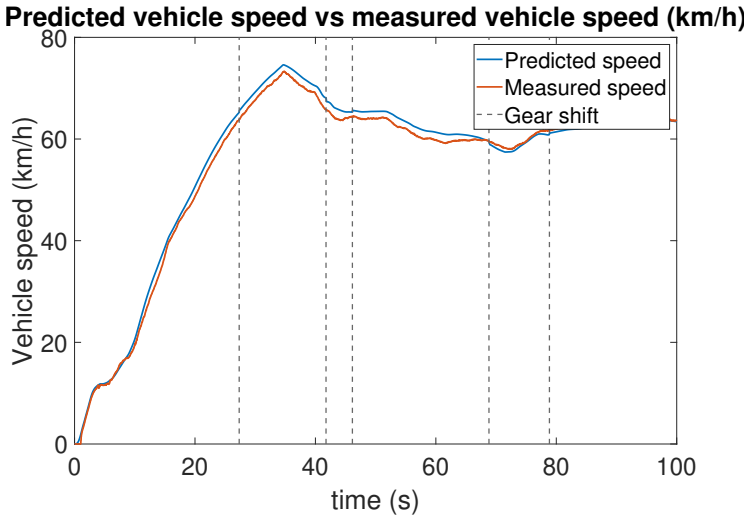


(b) Engine speed [rad/s]. Comparison of measured engine speed and simulated engine speed.

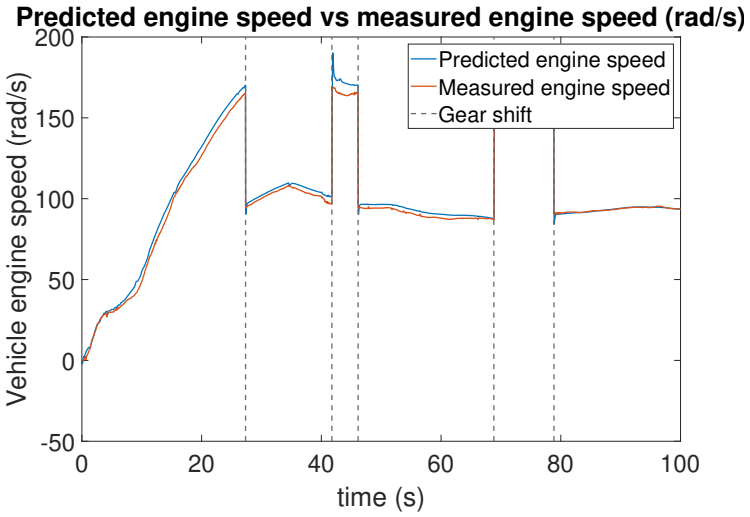


(c) Vehicle acceleration $[m/s^2]$. Comparison of derived vehicle speed to vehicle acceleration and simulated vehicle speed to vehicle acceleration.

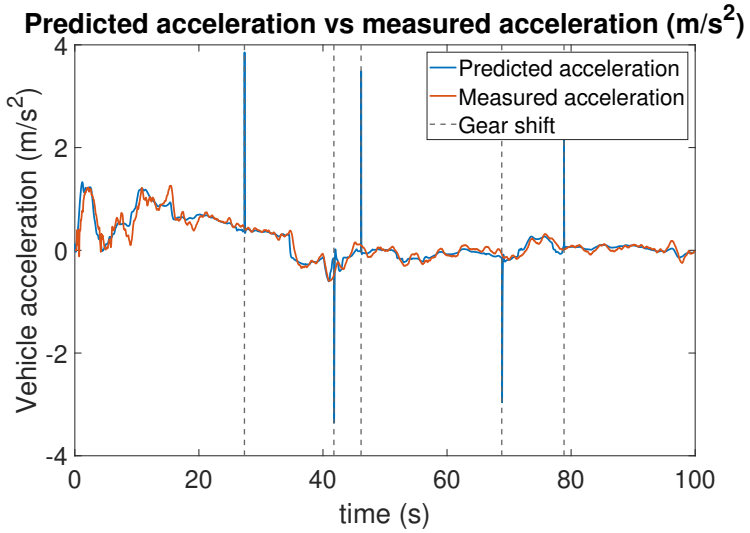
Figure 2.3: Comparison between measured and simulated data. Gear shifts are present, and the model does not adapt to them.



(a) Vehicle speed [km/h]. Comparison of measured vehicle speed and simulated vehicle speed.



(b) Engine speed [rad/s]. Comparison of measured engine speed and simulated engine speed.



(c) Vehicle acceleration $[m/s^2]$. Comparison of derived vehicle speed to vehicle acceleration and simulated vehicle speed to vehicle acceleration.

Figure 2.4: Comparison between measured and simulated data. Gear shifts are present, and the model adapts to them.

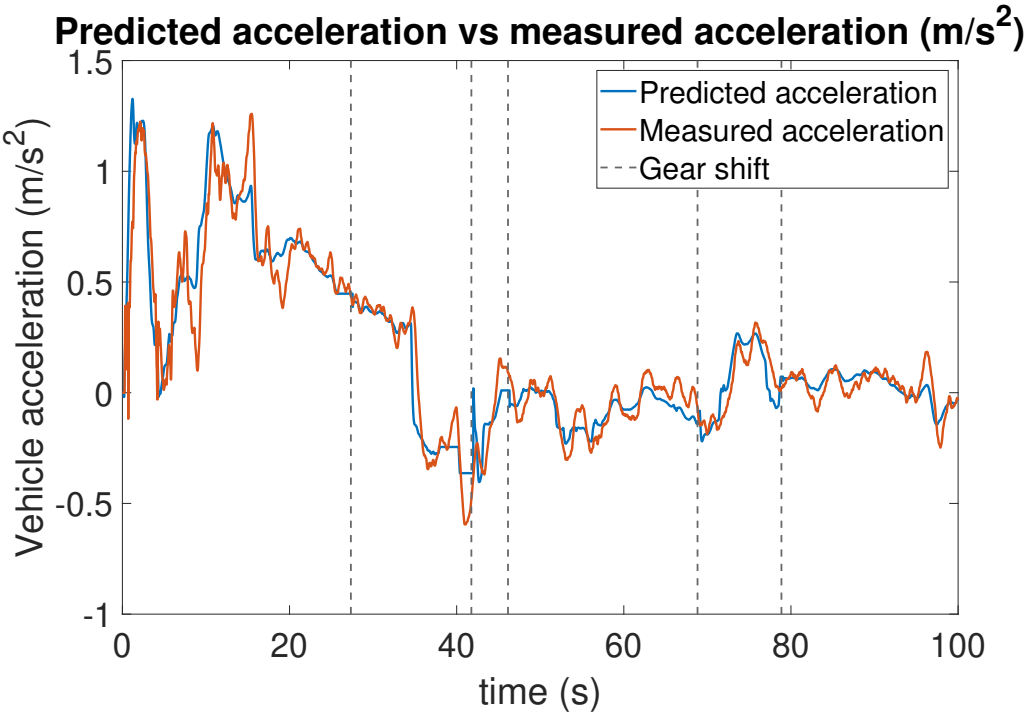


Figure 2.5: Comparison between simulated and measured acceleration. Here, the model adapts to the gear shift, keeps a constant acceleration during a gear, and maintains constant acceleration during the shift.

3
Analysis

To evaluate controller performance, it is necessary to analyze the system thoroughly for stability, robustness, and sensitivity. The drivetrain model developed in the previous section will be used as the plant G throughout this analysis. To conduct the analysis, the driveline state-space model was converted to a transfer function using MATLAB's `ss()` and `tf()` functions. The frequency response of the system will be evaluated using Bode and Nyquist plots, well-established tools in classical control theory for assessing stability and robustness.

3.1 Bode

A Bode plot illustrates a system's frequency response by plotting the magnitude $|G(j\omega)|$ and phase $\arg G(j\omega)$ against frequency, where ω is the angular frequency in radians per second. The magnitude is commonly expressed in decibels (dB) as $20\log_{10}|G(j\omega)|$, as described in [5]. By carefully analyzing the Bode plot, it can be concluded that a minimum-phase system is stable if the gain crossover frequency is lower than the phase crossover frequency. The gain margin indicates how far the magnitude deviates from 0 dB at the crossover frequency, measuring how much the loop gain can increase before the system reaches the boundary of instability. The phase margin, on the other hand, indicates how far the phase deviates from $-180°$ at the gain crossover frequency and measures how much additional phase lag the system can tolerate before becoming unstable. Together, the gain and phase margins provide a comprehensive picture of the system's robustness to parameter variations and modeling uncertainties.

The phase margin can furthermore be used to determine the maximum additional time delay the system can tolerate before becoming unstable. This relationship is given by

$$T_{max} = \frac{PM \cdot \pi/180}{\omega_c} \tag{3.1}$$

where $T_{\max}$ denotes the maximum allowable delay in seconds, PM is the phase margin in degrees, and ω_c is the gain crossover frequency in radians per second. Using (3.1), the phase margin expressed in degrees can be directly converted into the maximum allowable time delay in seconds before the system becomes unstable. This is particularly relevant in practical applications where computational delays, sensor latency, or communication lag may introduce non-negligible time delays into the control loop.

3.2 Nyquist

Bode plots are a widely used, effective method for assessing a system's stability and robustness in control engineering and are often the first tool of choice because of their intuitive graphical representation. However, interpreting stability margins depends on the open-loop pole configuration. Additional care is required for systems with poles in the right half-plane. Therefore, it is beneficial to complement and verify Bode plot results with a Nyquist plot. The Nyquist plot consolidates the complete frequency response into a single diagram in the complex plane, removing the need to cross-reference separate magnitude and phase plots. It also handles systems in which the phase crosses the $-180°$ line multiple times more naturally, making it a more general stability analysis tool. Another significant advantage of the Nyquist plot over the Bode plot is its ability to handle open-loop unstable systems using the Nyquist stability criterion. When assessing a system's stability using the Nyquist criterion, one only needs to know the number of unstable open-loop poles and the number of times the Nyquist curve encircles the critical point -1 in the complex plane. The difference between these two quantities directly determines the number of unstable closed-loop poles, thereby providing a stability assessment.

3.3 Step response

Another important, complementary approach to analyzing and validating a system's stability and performance is using step responses. A step response is a particularly useful tool for gaining a deeper, more intuitive understanding of a system's dynamic behavior, as it provides direct information on key time-domain performance metrics such as rise time, settling time, percentage overshoot, and steady-state value. These metrics relate closely to frequency-domain stability margins and, as such, the step response can effectively cross-validate the stability margins obtained from Nyquist and Bode plots. This is because any violation of these margins should manifest clearly as instability or significantly degraded transient behavior in the step response. By combining insights from frequency-domain analysis and time-domain step response, a more complete and reliable assessment of the closed-loop system performance can be obtained.

3.4 Plant

To analyze the driveline model in the frequency domain, a Bode plot was created for the model's transfer function from engine torque to wheel speeds.

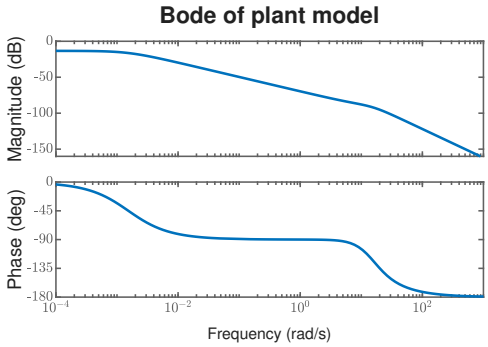


Figure 3.1: Bode plot of plant model

The Bode plot in Figure 3.1 shows no pronounced resonance peak. This suggests that driveline windup may not be the dominant source of the oscillatory behavior.

3.5 Current Controller

To identify potential solutions or improvements, a thorough understanding of the current solution is required. Therefore, a stability analysis of the controller currently in use, a proportional-integral feedforward controller (PI-FF), was carried out. Initially, Bode plots of the open-loop system were generated and analyzed. The plant output is speed, whereas this thesis focuses on acceleration. The plant output must therefore be differentiated to obtain the desired acceleration response, a step performed after the output is passed through a filter.

$$L(s) = Z(s)G(s)F(s)s \tag{3.2}$$

where $Z(s)$ is the controller, $G(s)$ is the plant model, and $F(s)$ is the filter.

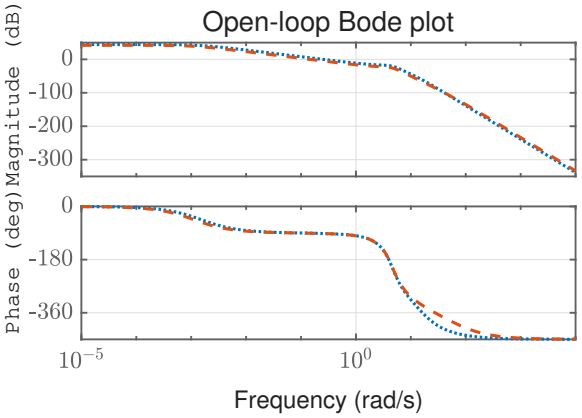
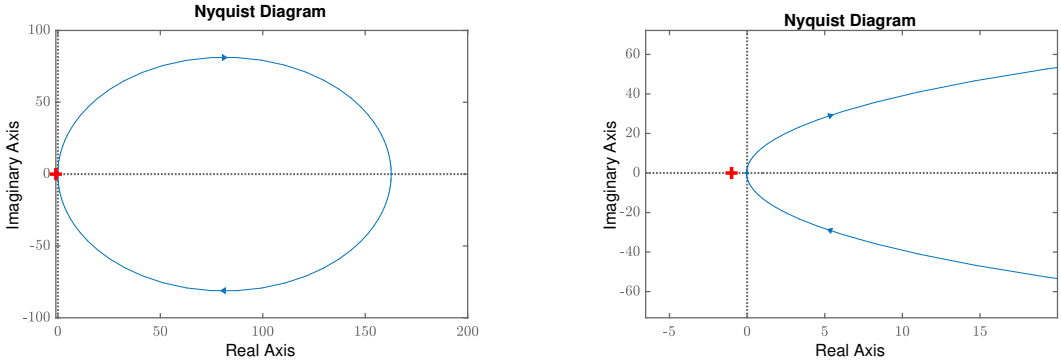


Figure 3.2: Open-loop Bode plot for different gear ratios

	GR = 4.6	GR = 2.6
Gain Margin (dB)	18.7	24.0
GM Frequency (rad/s)	3.97	3.99
Phase Margin (deg)	88.4	89.3
PM Frequency (rad/s)	0.251	0.143

Table 3.1: Stability for different gear ratios

Figure 3.2 shows two Bode plots for the current controller at different gear ratios. The red dashed line corresponds to a gear ratio of 4.6, and the blue dotted line to a gear ratio of 2.6. Both plots show positive gain and phase margins, respectively, indicating that the system is stable. Furthermore, the relatively large margin values in Table 3.1 suggest that the system is also quite robust. From the Bode plot, the closed-loop system is stable, as the magnitude does not exceed 0 dB when the phase is $-180°$, and the phase does not fall below $-180°$ when the magnitude is 0 dB.



(a) Nyquist **(b)** Nyquist plot

Figure 3.3: Nyquist plot

Table 3.2: Open-loop poles

Real Part	Imaginary Part
-12.0550	9.7698i
-12.0550	- 9.7698i
-1.7757	4.2868i
-1.7757	- 4.2868i
-4.2867	1.7758i
-4.2867	- 1.7758i
-0.0015	0.0000i

Table 3.3: Open-loop zeros

Zeros
-10.0000
-2.3333

Tables 3.2 and 3.3 confirm that the system is minimum phase, as all zeros have strictly negative real parts. Combined with the Nyquist plot in Figure 3.3, this confirms that the system is stable, as the Nyquist criterion is satisfied in accordance with [5]. Applying (3.1), the maximum allowable time delay was calculated to be 6.14 seconds.

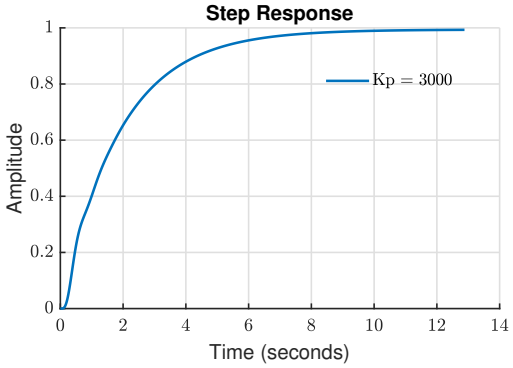


Figure 3.4: Closed-loop step response

To further validate the stability analysis, the step response of the closed-loop system was plotted, as shown in Figure 3.4. The response settles towards a steady-state value of 1, indicating that the closed-loop system is stable.

Analysis of both the plant model and the open-loop system shows no indication of oscillatory behavior. The plant model Bode diagram shows no resonant peak, and the open-loop system has large stability margins. The results suggest that the feedforward path may contribute to the observed oscillatory behavior and therefore warrants further investigation.

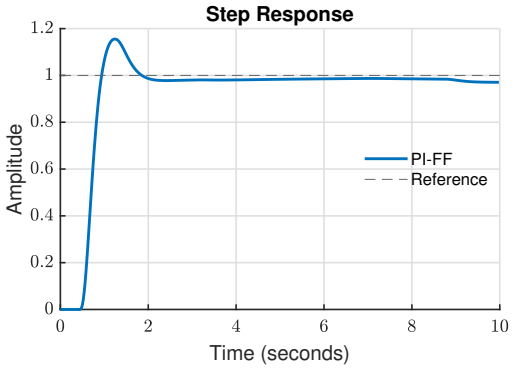


Figure 3.5: Step-response with feedforward

As evident by the step response when including the feedforward controller, significant overshoot occurs in the acceleration output. Since speed is the integral of acceleration, this overshoot directly propagates into an overshoot in vehicle speed. This effect is further amplified during driving conditions where the acceleration reference changes rapidly and continuously, so the system never has enough time to settle between references, allowing overshoots to compound. This is further exaggerated by the system time delay, since the feedback controller receives a delayed error signal, it reacts too late to correct the feedforward-induced overshoot, further amplifying the oscillatory behavior in speed. To address this, a Smith predictor is proposed. By compensating for the delay, the Smith predictor allows the feedback controller to react more quickly, directly reducing the compounding effect. Furthermore, the improved phase margins from delay compensation allow higher controller gains, which in turn may enable removal of the feedforward controller, eliminating the cause of the overshoot while maintaining adequate stability margins.

3.6 Delay estimation

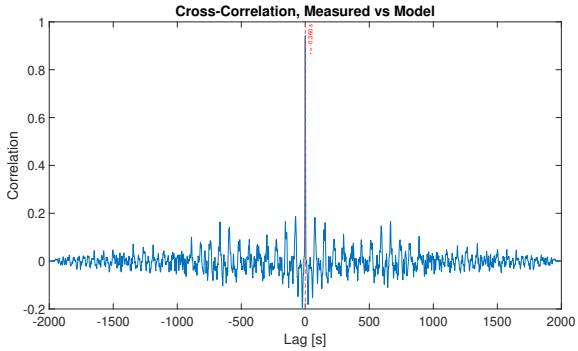


Figure 3.6: Delay estimation. The dotted red line shows $\tau = 0.36$, indicating an estimated time delay of 0.36s.

Since both the model and the plant (real vehicle) are driven by the same input signal, the model and measured acceleration should behave similarly. If the model accurately represents the vehicle dynamics but omits the delay in the real system,

the main difference between the two acceleration signals should be a time shift. This makes it possible to estimate the delay by finding the time shift that best aligns the modeled and measured acceleration signals. To estimate this time shift, the logged modeled and measured vehicle accelerations were compared using cross-correlation. Cross-correlation determines the similarity between two signals as one signal is shifted relative to the other. In this case, the modeled acceleration is shifted to the measured vehicle acceleration, and the similarity between the two signals is calculated for each lag. The lag resulting in the highest cross-correlation corresponds to the time shift at which the model output best matches the measured vehicle acceleration. This lag is converted into a time delay using the sampling time of the logged data. As seen in Figure 3.6, a time delay of 0.36 seconds best matches the measured data from the real vehicle, indicating that this is the delay present in the system.

4

Smith predictor

Systems and controllers with delays are inherently difficult to control, as noted in [5]. Delays introduce a phase shift that reduces the phase margin, making it difficult to maintain a large gain while preserving stability. The Smith predictor addresses this by compensating for dead-time effects. If a sufficiently accurate model is available and the system delay can be estimated reasonably well, a Smith predictor can significantly improve controller performance. The classical Smith predictor assumes stable open-loop plant dynamics and may perform poorly or become unstable when applied to systems with unstable poles, as mentioned in [6]. The analysis performed in the previous section shows that the system has stable open-loop dynamics, meaning the Smith predictor is applicable.

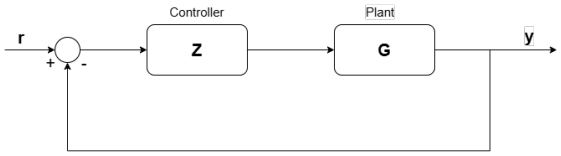


Figure 4.1: System without delay

For a system with plant G and controller Z without any delay, the closed-loop transfer function is given by

$$T_F = \frac{ZG}{1 + ZG}. \tag{4.1}$$

If the plant instead suffers from dead-time delays, the plant transfer function becomes $G = Ge^{-sT}$, where e^{-sT} represents the dead-time delay. The resulting closed-loop transfer function is then

$$T_F(s) = \frac{ZGe^{-sT}}{1 + ZGe^{-sT}}. \tag{4.2}$$

The delay appears in the denominator, directly affecting the characteristic equation and, therefore, the system's stability.

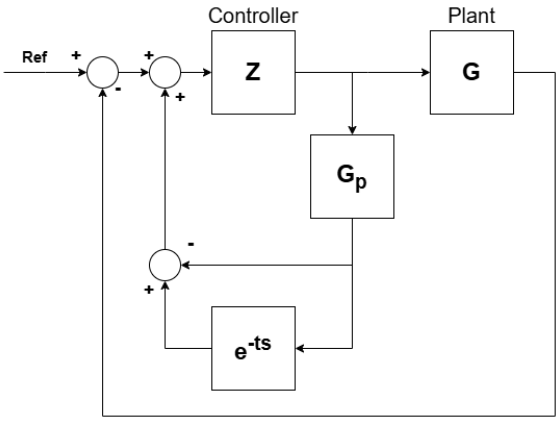


Figure 4.2: Smith predictor block diagram

The Smith predictor compensates for the delay e^{-sT} by introducing an internal plant model, as illustrated in Figure 4.2. In the ideal case, the plant model G_p perfectly matches the actual plant G, i.e., $G_p = G$. Under this assumption, the delayed plant output can be canceled by subtracting the delayed model output $G_p e^{-sT}$ and feeding back the undelayed model output G_p instead. The resulting controller transfer function is given by

$$C_{\text{smith}} = \frac{Z}{1 + (1 - e^{-sT}) Z G_p}. \tag{4.3}$$

Substituting C_{smith} into the closed-loop expression yields

$$T_F = \frac{C_{\text{smith}} G e^{-sT}}{1 + C_{\text{smith}} G e^{-sT}}, \tag{4.4}$$

which, after simplification, reduces to

$$T_F = \frac{Z G e^{-Ts}}{1 + ZG}. \tag{4.5}$$

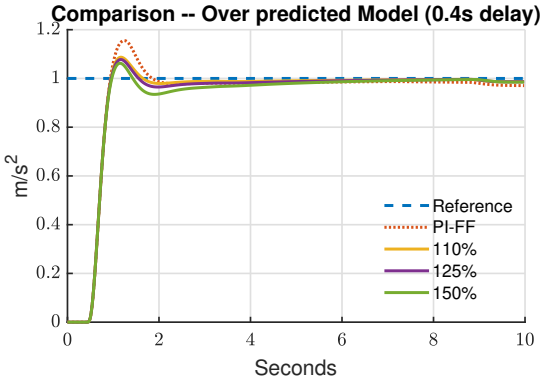
In the ideal case, the delay is removed from the characteristic equation and therefore no longer limits closed-loop stability as in the original delayed system. This effectively increases the stability margins, allowing the controller to be tuned for a faster response. This result holds only in the ideal case where the plant model and delay perfectly match the real system, which is never exactly achievable in practice.

4.1 Robustness to model and delay mismatch

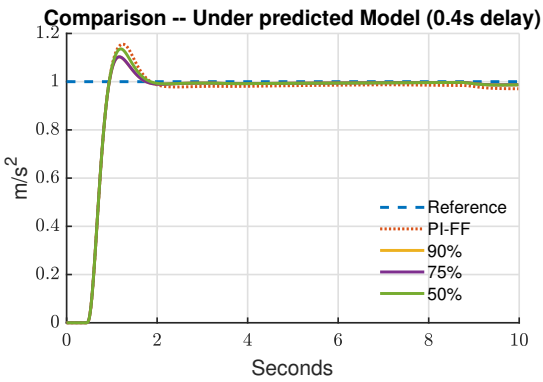
In any practical application of the Smith predictor, the plant model will not perfectly match the plant behavior. It is therefore important to assess the robustness of the Smith predictor to model mismatch. To do so, a series of simulation experiments was conducted. In the first scenario, performance was evaluated under varying levels of model mismatch. In the second scenario, varying levels of delay mismatch were considered. In the third scenario, a combination of both model and delay mismatch was evaluated. All experiments used the current controller as the base, consisting of a PI and a feedforward controller, with no tuning modifications.

4.2 Model mismatch

The Smith predictor is notoriously sensitive to model uncertainties and mismatch, as noted in [6]. It is therefore important to investigate its impact on the system and controller. In this scenario, the delay estimation is assumed to be perfect, meaning there is no difference between the actual and modeled delay. This isolates the effect of model mismatch on the controller's performance and stability. Since previous validation showed that the dominant discrepancy between model and measurements was amplitude-related rather than dynamic, as seen in Figure 2.4, gain mismatch was selected as a representative uncertainty. The percentages used in the plots represent the static gain on the model, for example 150% means the estimated output is the plant output multiplied by 1.5 to introduce the mismatch.



(a) Overpredicted model mismatch



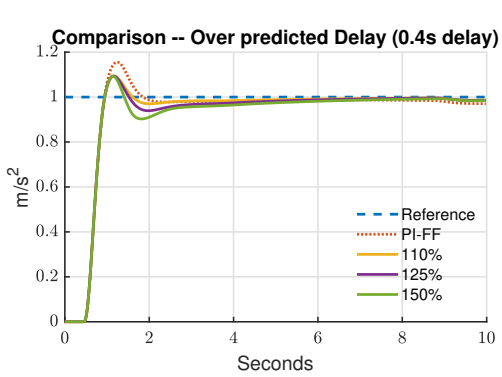
(b) Underpredicted model mismatch

Figure 4.3: Comparison of model mismatch

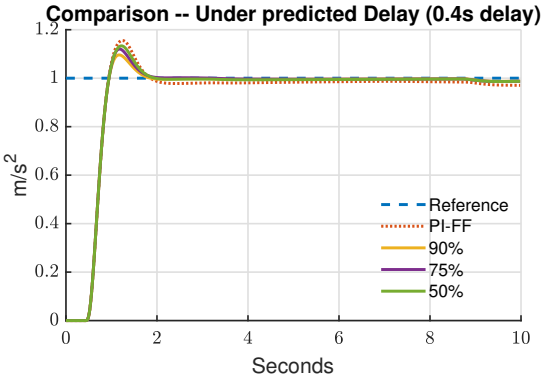
Inspecting Figure 4.3, the Smith predictor responds to the step input faster than the baseline in all cases. Controller performance also degrades as modeling error increases, for both positive and negative mismatch. Importantly, however, the system remains stable across all tested levels of model mismatch.

4.3 Delay mismatch

In this scenario, the plant model is assumed to be perfect, meaning there is no difference between the actual and modeled plant output. This isolates the effect of delay mismatch on the controller's performance and stability.



(a) Over-predicted delay mismatch



(b) Under-predicted delay mismatch

Figure 4.4: Comparison of delay mismatch

As seen in Figure 4.4, the Smith predictor again responds to the input signal faster than the baseline, and performance degrades as the delay mismatch increases. Notably, the system remains stable for delay mismatches of up to ±50%.

4.4 Combined mismatch

In any practical application, it is likely that both model mismatch and delay mismatch occur simultaneously rather than independently. Therefore, several worst-case scenarios were simulated in which both types of mismatch were present simultaneously. These tests were conducted to ensure system stability under more realistic operating conditions and mismatches.

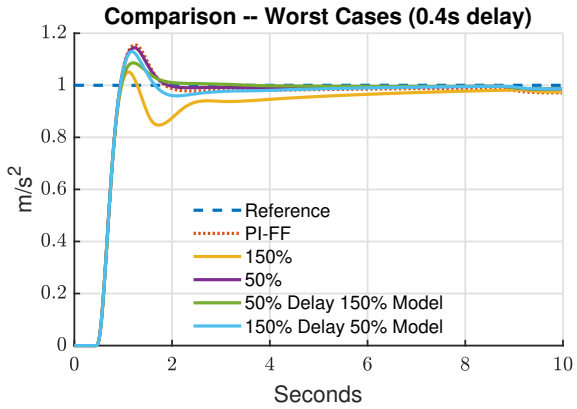


Figure 4.5: Comparison of worst cases with model and delay mismatch

When both model and delay mismatches are introduced simultaneously, performance drops further. As shown in Figure 4.5, the system remains stable, demonstrating that the Smith predictor retains reasonable robustness even under combined mismatch.

4.5 Controller tuning with Smith prediction

In the previous scenarios, the controller was not retuned after introducing the Smith predictor, so additional performance gains are achievable through tuning. As established earlier, the Smith predictor lets the system tolerate larger gains, enabling tuning for a faster response.

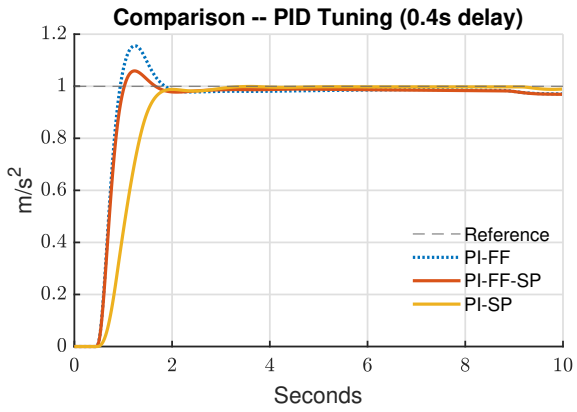


Figure 4.6: Comparison of different PI tunes

Figure 4.6 shows three controller and tuning variations. PI-FF is the state-of-the-art PI and feedforward-based controller used today. PI-FF-SP is the controller with a Smith predictor added. PI-SP uses the PI controller but removes the feedforward and adds the Smith predictor. With the Smith predictor, the PI controller can be tuned more aggressively, and the response speed improves noticeably. It is also worth noting that most results in this section show consistent overshoot. This is attributed to the feedforward controller, which was included to achieve a faster response, as mentioned in [6]. With the introduction of the Smith predictor, additional tests were conducted with the feedforward removed to reduce overshoot while maintaining a fast and accurate response.

4.6 Validation

To validate the effect of the Smith predictor, a delay was introduced into the plant G, and the step response of the system was evaluated with and without the Smith predictor for increasing values of the proportional gain K_p.

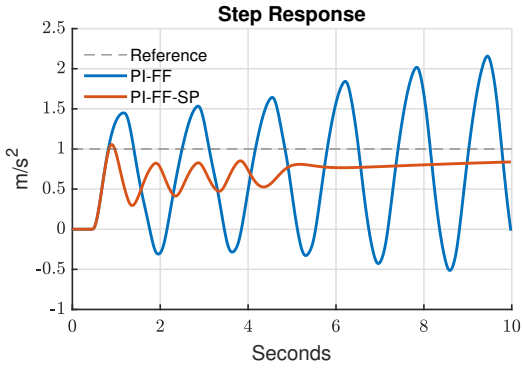


Figure 4.7: Step response of PI-FF-SP during increasing gain

As seen in Figure 4.7 at a K_p gain of 37 000, the system without the Smith predictor goes unstable. The system with the Smith predictor can maintain stability at the same gain level that caused instability without it. This demonstrates that the Smith predictor permits substantially larger proportional gains before instability occurs.

4.7 Measurement-updated Smith predictor

The model presented in the modeling section captures the vehicle dynamics well, however, throughout model validation and testing of the Smith predictor, the model output speed does deviate from the actual vehicle speed, and in some cases, the model output and actual vehicle speed can diverge significantly. Despite the speed error, the Smith predictor can still estimate vehicle acceleration reasonably well in most cases. However, it produces prediction errors that are too frequent and too large, making it unsuitable for use in an automated driving system.

To address this deviation, measurements can be incorporated to prevent the Smith predictor from drifting under disturbances and model mismatch. Although the measurements are delayed, they are assumed to be accurate enough upon arrival, meaning they are sufficient for the PI controller to act appropriately. The core idea is straightforward: at every timestep, the most recent delayed measurement provides a known and reliable snapshot of the vehicle state Δt seconds in the past. The Smith predictor then reinitializes from this snapshot and simulates the vehicle model forward by Δt seconds, producing an estimate of the current vehicle state. This way, rather than accumulating error over a long prediction horizon, the predictor only has to bridge the short, fixed gap of Δt seconds, keeping the prediction accurate and tightly anchored to reality. In Figure 4.8, a simple example demonstrates how the measurement-updated Smith predictor maintains an accurate prediction even when the model suffers from a static gain error or other model mismatch.

To orchestrate this in a discrete Simulink model, a for loop block is used to iterate $\frac{\Delta t}{T_s}$ timesteps ahead from the most recent measurement using the model provided in 2.25. The for loop is necessary because at each timestep the predictor must execute multiple simulation steps within a single Simulink timestep, something a

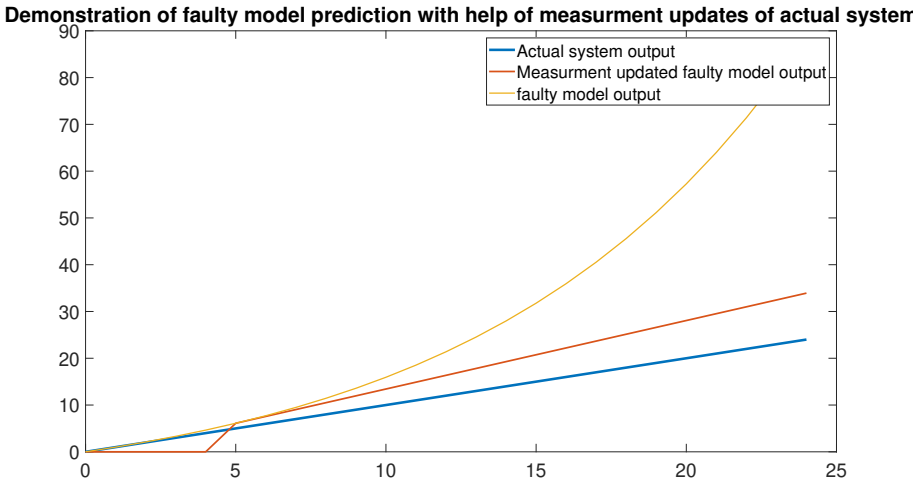


Figure 4.8: Simple example of how the measurement-updated Smith can maintain a more accurate performance compared to the faulty model not updated with measurements. The horizontal axis represents time, while the vertical axis represents the model output.

standard unit delay cannot accommodate. However, since the driveshaft torsion is not directly measurable, and the full state vector $\hat{x}$ is required to reinitialize the predictor at each timestep, a state estimator is needed. This is done using a Kalman filter state estimator, which, according to [1], is the optimal linear filter under the assumption of Gaussian disturbances. Given the system

$$\begin{aligned} x_{k+1} &= A_d x_k + B_d u_k + N v_{1,k} \\ y_k &= C_d x_k + D_d u_k + v_2 \end{aligned} \tag{4.6}$$

with v_1 and v_2 being process and measurement disturbances, respectively. The discrete-time Kalman state estimator is given as

$$\hat{x}_{k+1} = A_d \hat{x}_k + B_d u_k + L(y_k - C_d \hat{x}_k - D_d u_k) \tag{4.7}$$

The Kalman gain L is calculated using

$$L = (A_d P C_d^T + N R_{12})(C_d P C^T + R_2)^{-1} \tag{4.8}$$

where P is the solution to the discrete-time algebraic Riccati equation shown in (4.9).

$$P = A_d P A_d^T + N R_1 N^T - (A_d P C_d^T + N R_{12})(C_d P C_d^T + R_2)^{-1}(A_d P C_d^T + N R_{12})^T \tag{4.9}$$

where R_1 and R_2 are the noise intensities for v_1 and v_2, and R_{12} is the cross-spectrum between the two noise terms. It is also assumed that the system is observable, meaning that the observability matrix $\mathcal{O}(A_d, C_d)$ has full rank, which is the case for this specific system.

The measurements available are vehicle speed and engine speed. However, in the vehicle model presented in 2.25, C_d includes only vehicle speed. Since engine speed

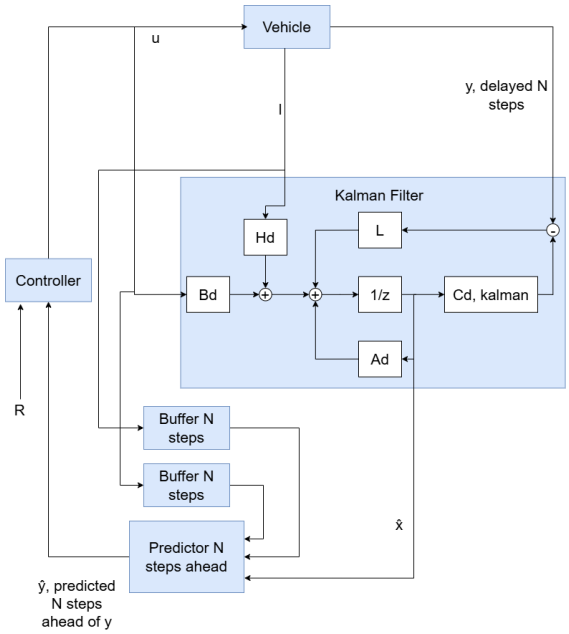


Figure 4.9: Block diagram of measurement-updated Smith predictor structure.

provides direct observability into the drivetrain state, helping to improve the state estimation, a new output matrix incorporating both measurements is proposed for the Kalman filter.

$$C_{d,kalman} = \begin{bmatrix} C_d \\ C_{engspeed} \end{bmatrix}$$
$$C_{d,engspeed} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \tag{4.10}$$

The final structure of the Measurement-updated Smith predictor (MUSP) can be seen in figure 4.9.

5

LQG and integration control

Previous Chapters 3 and 4 focused on analyzing and supporting the existing current controller setup. In contrast, this chapter introduces a new model-based control strategy to investigate whether an alternative design philosophy can mitigate or avoid the identified weaknesses of the state-of-the-art controller.

Using a control strategy separate from the existing PI-FF controller may also yield different closed-loop behavior, potentially reducing overshoot that, as discussed in Chapter 3, can be problematic during rapid reference tracking. The proposed strategy is an LQG controller augmented with integral action, with particular emphasis on robustness and performance under model uncertainty.

5.1 Motivation for LQG Control

In the modeling chapter, a linear state-space model is presented. Simulations show that the model captures vehicle speed and its derivative, acceleration, with high accuracy over short prediction horizons. However, due to disturbances such as input errors and model uncertainties, the predicted speed gradually deviates from the measured speed. In an open-loop configuration, this deviation may become significant over time.

Nevertheless, this does not imply that state-feedback-based control is infeasible using the proposed state space model. As discussed in [7], feedback can reduce the effects of uncertainties, disturbances, and model imperfections. Because the presented model is linear and shows high short-term predictive accuracy, it is well suited for state-feedback control design.

5.1.1 State Estimation and LQG Control

As discussed in Section 4.7, the drive shaft torsion is not directly measurable, so not all states are available for feedback control. Consequently, a state estimator is required to reconstruct the full system state from available measurements. As mentioned, the Kalman filter provides the optimal state estimate for linear systems subject to Gaussian noise, and when combined with an LQR regulator, it forms the LQG controller, which, under the assumptions discussed in [1], yields the optimal linear output-feedback control solution. The LQG controller minimizes a cost

function that balances control error and input effort. A key assumption is that the system matrices are known exactly, which is not strictly true here. However, because the model shows high short-term accuracy and feedback continuously corrects prediction errors using measurements, the controller does not rely on accurate long-term open-loop predictions. This makes LQG a well-suited design choice despite the modeling uncertainties. Additional motivation for adopting LQG appears in [3], where appropriate state weighting reduces driveline oscillations caused by driveline windup, a relevant concern in the present application.

5.1.2 Robustness Considerations

A key consideration when applying LQG control is the closed-loop system's robustness. As noted in [7], the strong robustness properties associated with LQR control when all states are directly measurable do not necessarily carry over to the output-feedback case, which applies to the present application. Consequently, the use of Loop Transfer Recovery (LTR) is proposed to restore the controller's desired robustness properties. Although LTR can recover important robustness characteristics, practical implementations typically cannot recover them exactly due to actuator limitations, model uncertainty, and estimator tuning constraints.

5.1.3 Integral Action

In [8], the authors propose a weighted LQG-LTR approach complemented by an integral action introduced by [9]. The authors claim that this hybrid LQG-LTR with integral action guarantees zero steady-state tracking error for constant references. This weighted LQG-LTR with integral action is particularly relevant to the present project because it combines LQG control characteristics with LTR's improved stability margins and integral action's steady-state tracking performance. This is especially pertinent given the goal of improving handling of small acceleration errors, as identified among the objectives discussed in the introduction.

5.1.4 Practical Considerations

The model presented in section 2 does not fully represent the vehicle under saturated inputs. Because the vehicle is equipped with actuators with actuation limits, we must account for this. Thus, an integration anti-windup scheme is added, following [10], in which the integrator is conditionally updated based on the saturation error.

The LQG solution presented is as mentioned based on the model presented in section 2 where the output is vehicle speed, with that being said, the aim is to control vehicle acceleration. For simplicity, the model is not altered, and the controller is therefore designed to control vehicle speed. Since acceleration is directly related to the derivative of vehicle speed, tracking an integrated acceleration reference allows the problem to be formulated within the existing speed-based state-space model without introducing additional differentiation noise, thus allowing the controller to control the vehicle acceleration.

5.2 Kalman Filter

The same Kalman filter structure as in Section 4.7 is adopted here, utilizing both engine speed and vehicle speed as measurements. Including engine speed improves state reconstruction quality and speeds convergence of the estimation error. According to the separation principle, the controller and observer can be designed independently with respect to nominal closed-loop stability, since the closed-loop eigenvalues consist of the controller and observer eigenvalues. However, the observer dynamics, estimation errors, noise sensitivity, and model uncertainties may still affect the robustness of the overall output-feedback system. Nevertheless, in practice, the improved state estimation from the additional measurement allows the controller to operate on a more accurate estimate of the system states, which can in turn improve the overall system robustness.

The resistance load torque is available as a measurement and is therefore treated as a known input to the model, as formulated in 2.25. Since the load torque is highly nonlinear, it is not included as a state in an augmented system. Treating it as a measured input allows its effect to be accounted for directly, without introducing nonlinear dynamics into the estimator, and, as a result, it does not affect the Kalman gain calculations.

5.3 Gains

To introduce LQG with additional integral control, ILQG, as proposed in [9], the system to be controlled is described by

$$\begin{aligned} \hat{x}_{k+1} &= A_d\hat{x}_k + B_d u_k + v_{1,k} \\ y_k &= C_d\hat{x}_k + w_{2,k} \end{aligned} \tag{5.1}$$

and is controlled by the input u_k defined as

$$u_k = -K_x\hat{x}_k - K_i x_{I,k} \tag{5.2}$$

Here, $x_{I,k}$ denotes the integral state. To apply the standard LQ regulator framework, the state vector must therefore be augmented as

$$x_{aug} = \begin{bmatrix} \hat{x}_k \\ x_{I,k} \end{bmatrix} \tag{5.3}$$

The augmented gain matrix is then given by

$$K_{aug} = \begin{bmatrix} K_x & K_i \end{bmatrix} \tag{5.4}$$

and can be obtained by solving the Riccati equation that minimizes the cost function

$$J = \sum_{n=0}^{\infty} \left(x_{aug,k}^T Q x_{aug,k} + u_k^T R u_k \right) \tag{5.5}$$

where Q and R are the quadratic weighting matrices for the state and input, respectively. To incorporate the integral state into the state-space model, one can discretize it using the forward Euler method. This is done by approximating the time derivative

$$\dot{x}_I(t) = e(t) \tag{5.6}$$

using a forward finite difference approximation

$$\dot{x}_I(t_k) \approx \frac{x_{I,k+1} - x_{I,k}}{T_s} \tag{5.7}$$

where T_s denotes the sampling time. Substituting the continuous-time integrator with the forward finite-difference approximation yields a final discrete integrator update step.

$$\begin{aligned} \frac{x_{I,k+1} - x_{I,k}}{T_s} &= e_k \\ x_{I,k+1} &= x_{I,k} + T_s e_k \end{aligned} \tag{5.8}$$

The integrator state is updated using the error between the Kalman prediction $C_d\hat{x}_k$ and the reference r. This reduces the reliance on an accurate, non-delayed vehicle measurement by using the Kalman state predictor as a filter as well.

$$x_{I,k+1} = x_{I,k} + T_s(r_k - C_d\hat{x}_k) \tag{5.9}$$

The resulting augmented state-space model is represented as:

$$A_{aug} = \begin{bmatrix} A_d & 0 \\ -T_s C_d & 1 \end{bmatrix}, \quad B_{aug} = \begin{bmatrix} B_d \\ 0 \end{bmatrix}, \quad C_{aug} = \begin{bmatrix} C_d & 0 \end{bmatrix} \tag{5.10}$$

With this augmented state-space model, the ILQG regulator gain K_{aug} can be computed, as in [1], by solving the discrete-time algebraic Riccati equation:

$$P_{aug} = A_{aug}^T P_{aug} A_{aug} - A_{aug}^T P_{aug} B_{aug} \left(R + B_{aug}^T P_{aug} B_{aug}\right)^{-1} B_{aug}^T P_{aug} A_{aug} + Q \tag{5.11}$$

where P is the positive semi-definite solution matrix, and the corresponding optimal gain matrix is given by

$$K_{aug} = \left(R + B_{aug}^T P_{aug} B_{aug}\right)^{-1} B_{aug}^T P_{aug} A_{aug} \tag{5.12}$$

When computing K_{aug}, it is assumed that the system is controllable, meaning that $\mathcal{S}(A_{aug}, B_{aug})$ has full rank. Since the load disturbance is measurable but not controllable, it is not included here.

The gains depend only on A_d and B_d. Therefore, including engine speed as a measurement in the state estimator does not affect the LQR or ILQG gains.

5.4 Controller structure

Figure 5.1 shows a block diagram of the controller structure.

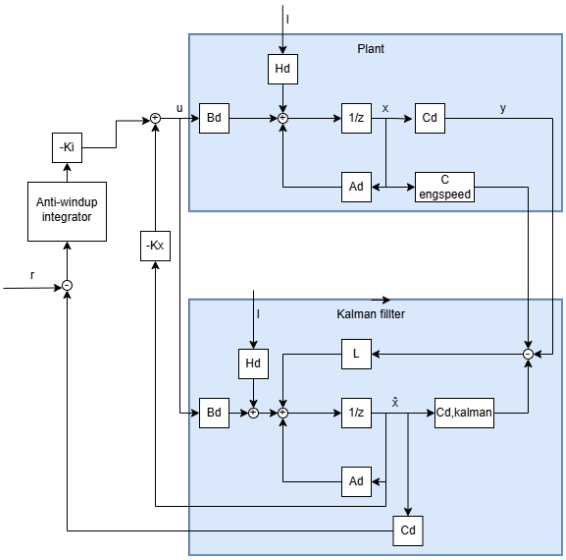


Figure 5.1: Block diagram of ILQG controller structure

5.5 Stability and Robustness Tuning

To recover the robustness margins lost in the output-feedback case, the output LTR proposed by [11] is evaluated. This choice is motivated by the fact that, for SISO systems, the output and input LTRs are practically identical, as noted in [1]. For the output LTR, $Q = C^T C$, and the input penalization R is replaced by $\rho\mathbf{I}$, where driving ρ towards zero should recover the system margins. However, in this application, removing the input penalization leads to overly aggressive input torque, which drastically reduces stability margins. The standard output LTR procedure, therefore, does not work as intended for this system, and an alternative tuning approach is required.

Instead, the method adopted is inspired by the sensitivity of R. The design follows a two-step tuning procedure. In the first step, $Q = \text{diag}([q_i,\ q_1,\ q_2,\ q_3])$ and R are tuned manually to achieve satisfactory closed-loop tracking performance and disturbance rejection. In the second step, Q is replaced by $\bar{Q} = \rho Q$, where driving ρ towards zero softens the state penalty relative to the control penalty, causing the controller to penalize input more heavily and thereby making the controller slower, causing a slower rise time and less overshoot. The value of ρ is selected as the smallest value for which satisfactory stability margins are achieved while tracking performance remains acceptable, reflecting a deliberate trade-off between robustness and performance.

To further improve the stability margins, the Kalman gain is also tuned by reducing

the process-noise covariance matrix, causing the estimator to rely more heavily on the model. However, this must be done carefully because the integral action is based on the error between the estimated output, $(C_d\hat{x})$, and the reference signal, (r). If $C_d\hat{x}$ is determined primarily by the model, significant steady-state errors may occur when the model is inaccurate.

It is important to balance reducing controller performance with relying on Kalman filter predictions to improve system margins while avoiding significant steady-state errors.

5.5.1 Open-Loop Transfer Function and Margin Analysis

To evaluate the stability and robustness of the closed-loop system, the open-loop Bode plot and its associated margins are analyzed using the open-loop transfer function $L = G_{\text{controller}}G_{\text{plant}}$. The controller is treated as a standalone system with the tracking error e as input, and the control signal u as output. The states of the standalone controller are therefore $x_c = \begin{bmatrix} \hat{x} \\ x_I \end{bmatrix}$, resulting in the controller state-space system

$$\begin{aligned} x_{c,k+1} &= A_c x_{c,k} + B_c y_k + B_r r_k \\ u_k &= C_c x_{c,k} \end{aligned} \tag{5.13}$$

To find A_c, B_c, B_r and C_c, $\hat{x}_{k+1}$ and $x_{I,k+1}$ are described using 5.14 and 5.9 respectively.

$$\hat{x}_{k+1} = A_d\hat{x}_k + B_d u_k + L(y_k - C_d\hat{x}_k) \tag{5.14}$$

It is then possible to substitute u_k in the equation above resulting in that $\hat{x}_{k+1}$ can be described as

$$\begin{aligned} \hat{x}_{k+1} &= A_d\hat{x}_k + B_d(-K_x\hat{x}_k - K_i x_{I,k}) + L(y_k - C_d\hat{x}_k) \\ \hat{x}_{k+1} &= (A_d - B_dK_x - LC_d)\hat{x}_k + (-B_dK_i)x_{I,k} + Ly_k \end{aligned} \tag{5.15}$$

Together with 5.9 and 5.2 it is now possible to construct matrices A_c, B_c, B_r and C_c.

$$\begin{aligned} A_c &= \begin{bmatrix} A_d - B_dK_x - LC_d & -B_dK_i \\ -T_sC_d & 1 \end{bmatrix} \\ B_c &= \begin{bmatrix} L \\ 0 \end{bmatrix} \\ B_r &= \begin{bmatrix} 0 \\ T_s \end{bmatrix} \\ C_c &= \begin{bmatrix} -K_x & -K_i \end{bmatrix} \\ D_c &= 0 \end{aligned} \tag{5.16}$$

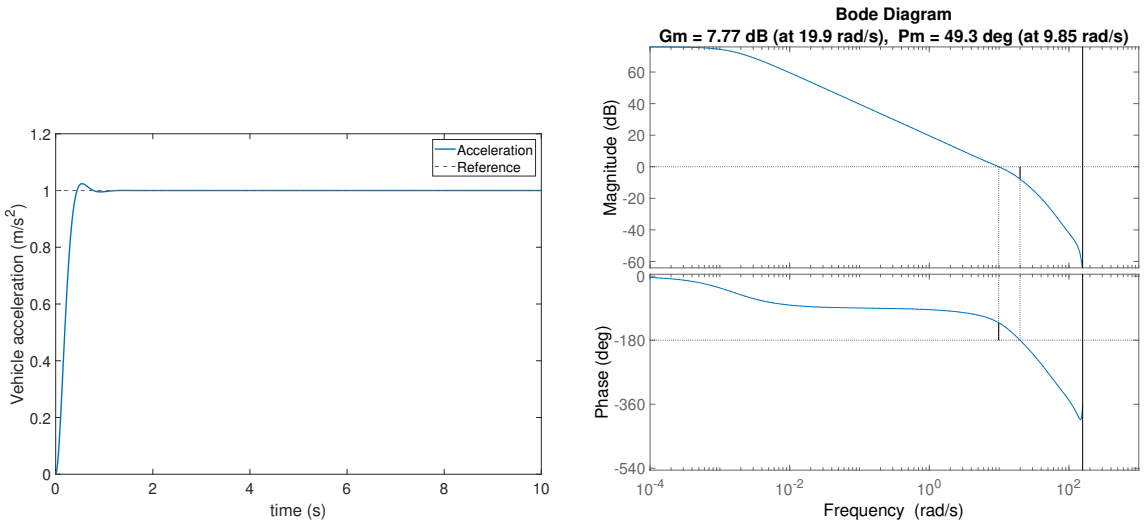
Because the reference signal is not part of the feedback-loop dynamics, it does not affect the loop transfer function. The controller transfer function $G_{\text{controller}}$ is consequently constructed using only A_c, B_c, C_c and D_c. Since the plant is SISO and the use of two measurements in the Kalman filter introduces a MISO element

into the standalone controller, the margin analysis is carried out using only vehicle speed as a measurement. This is because engine speed does not drive the integrator and is therefore not part of the full feedback loop; including it would yield an artificially optimistic picture that does not reflect the actual closed-loop behavior, whereas omitting it gives conservative margins that are expected to be preserved or improved in the full two-measurement configuration.

5.5.2 Tuning for robustness

Figures 5.2 and 5.3 show the controller tuning results with respect to robustness. The former is tuned solely for performance, whereas the latter attempts to recover the stability margins. The performance metrics for both tuning strategies are presented in Table 5.1.

The proposed robust tuning strategy essentially reduces controller performance, making the controller less aggressive. To understand how gain margins affect system robustness, (3.1) can be used to calculate the maximum allowable measurement and actuator delays. The resulting T_{max} values in Figure 5.2 show that the maximum allowed delay can increase drastically while controller performance remains acceptable.



(a) Step response **(b)** Bode diagram

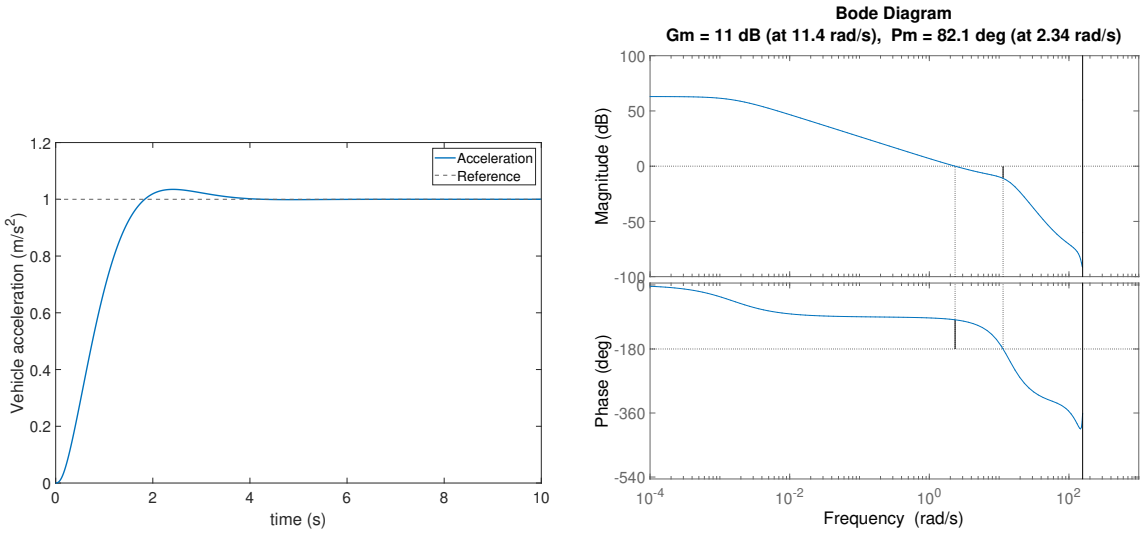
Figure 5.2: LQG controller tuned for accuracy and fast settling time

	Rise time [s]	Overshoot [%]	Settling time [s]	Steady-state error $[m/s^2]$
Performance tuning	0.2	0.2564	0.32	2.309*10-14
Robustness tuning	0.44	2.104	1.08	2.309*10-14

Table 5.1: Performance measurements for different controller tuning

	T_{max}
Performance tuning	0.0874
Robustness tuning	0.6124

Table 5.2: T_{max} for different controller tuning



(a) Step response **(b)** Bode diagram

Figure 5.3: LQG tuning with respect to robustness

6
Results

This chapter presents and evaluates the results from simulation and in-vehicle testing of the model created in Chapter 2 and the controller structures developed in Chapters 4 and 5. The chapter is divided into 2 sections to address the 2 research questions formulated in Chapter 1. The driveline model will be evaluated using RMSE against logged data. The controllers, on the other hand, will be evaluated through step response and metrics such as rise time and settling time, as well as comparisons with logged data and in-vehicle testing results.

6.1 Research Question 1

Research question 1 evaluates the driveline model's performance. This comparison was based on logged vehicle values. The section presents the model's results after tuning.

6.1.1 Model

The resulting tuned model, compared with the measurements, is shown in Figures 6.1, 6.2, and 6.3. As mentioned in the Modeling chapter, the measurements used for comparison are vehicle speed (km/h) and engine speed (rad/s). In addition, both measured and simulated vehicle acceleration are presented, as acceleration is the primary control target. It should be noted that the acceleration is not directly measured or predicted by the model, instead it is obtained by differentiating the measured and simulated vehicle speed signals, respectively. Nonetheless, comparing accelerations still provides valuable insight into the system's dynamic behavior, especially during small speed changes, where differences may be difficult to observe in the speed signals alone.

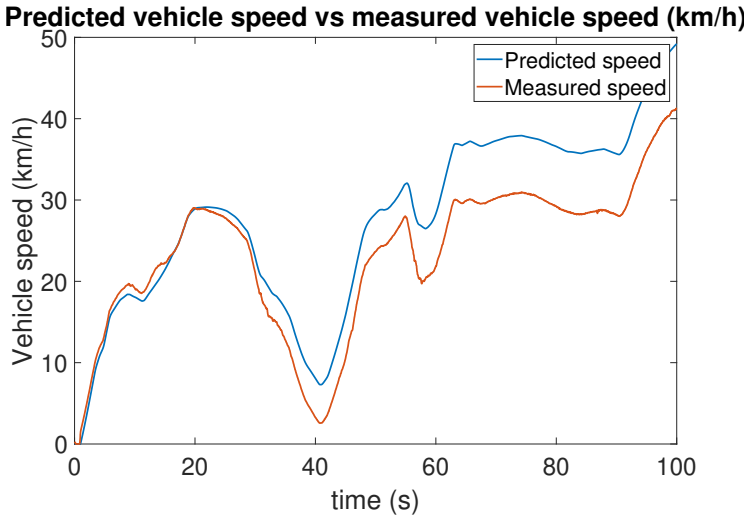
The validation results show that the model captures the overall vehicle dynamics with good accuracy. In the initial stages of the simulations, the measured and simulated vehicle and engine speeds agree closely. As the simulations progress, a gradual deviation between the measured and simulated speeds becomes apparent. Such deviations are expected, as the model represents a simplified description of a complex physical system and therefore cannot capture all aspects of the real vehicle behavior.

Although deviations in vehicle and engine speeds increase over time, the agreement between the measured and simulated acceleration remains relatively good throughout the validation period. This indicates that the model continues to capture the dominant system dynamics despite the accumulation of offset error. Furthermore, the model reproduces the oscillatory behavior observed in engine speed, demonstrating that including drive-axle torsion provides a realistic representation of driveline dynamics.

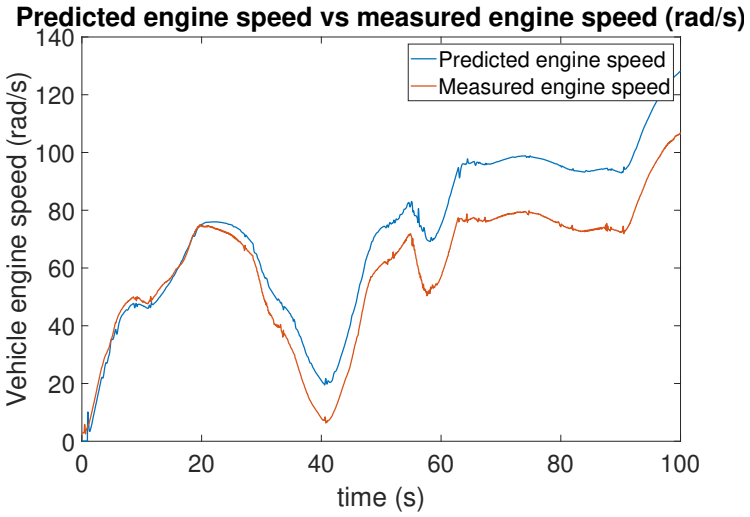
The validation results indicate that a third-order driveline model with engine speed, vehicle speed, and drive-axle torsion as states can capture the dominant vehicle dynamics with sufficient accuracy. The model can reproduce the transient behavior, acceleration response, and driveline oscillations observed in the measurements, making it suitable for further system analysis, simulation studies, and model-based control development.

6.2 Research Question 2

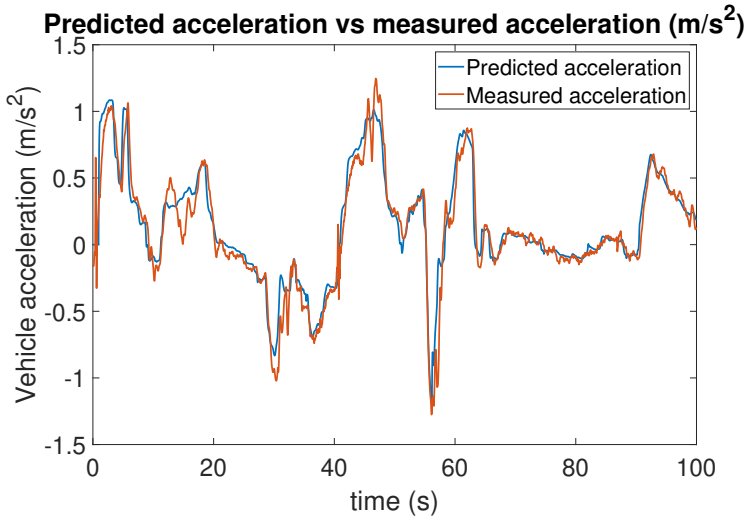
Research question 2 evaluates the impact of model-based control strategies on the acceleration controller's performance. This section presents and discusses the results from the Smith predictor, the measurement-updated Smith predictor, and the LQG controller.



(a) Vehicle speed [km/h]. Comparison of measured vehicle speed and simulated vehicle speed.

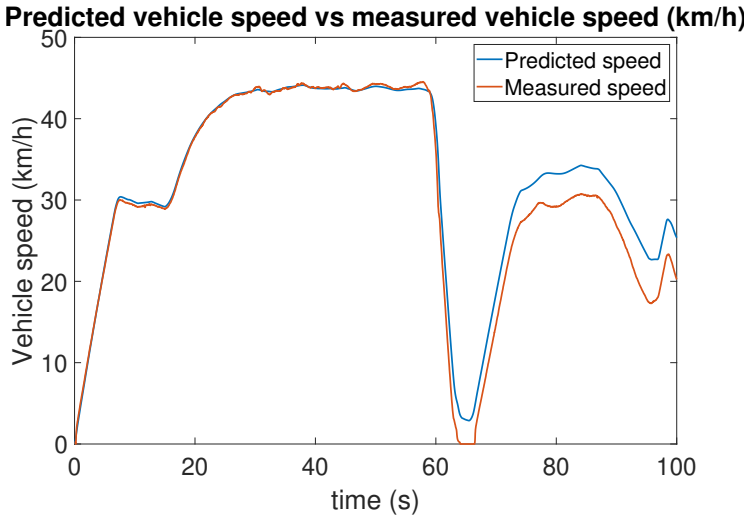


(b) Engine speed [rad/s]. Comparison of measured engine speed and simulated engine speed.

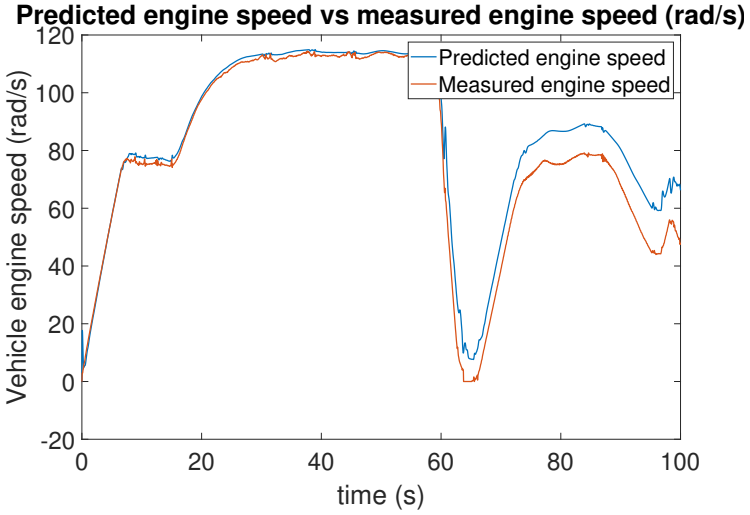


(c) Vehicle acceleration $[m/s^2]$. Comparison of differentiated vehicle speed to obtain vehicle acceleration, and simulated vehicle speed to obtain simulated vehicle acceleration.

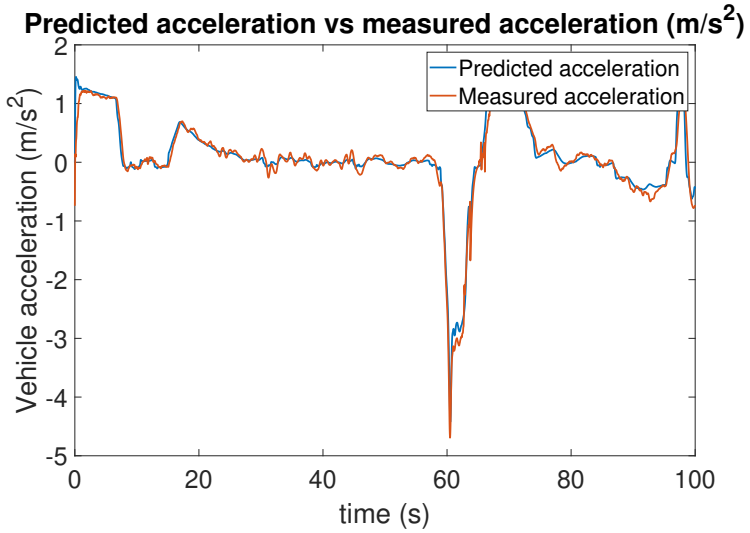
Figure 6.1: Comparison between measured and simulated data during driving opportunity 1.



(a) Vehicle speed [km/h]. Comparison of measured vehicle speed and simulated vehicle speed.

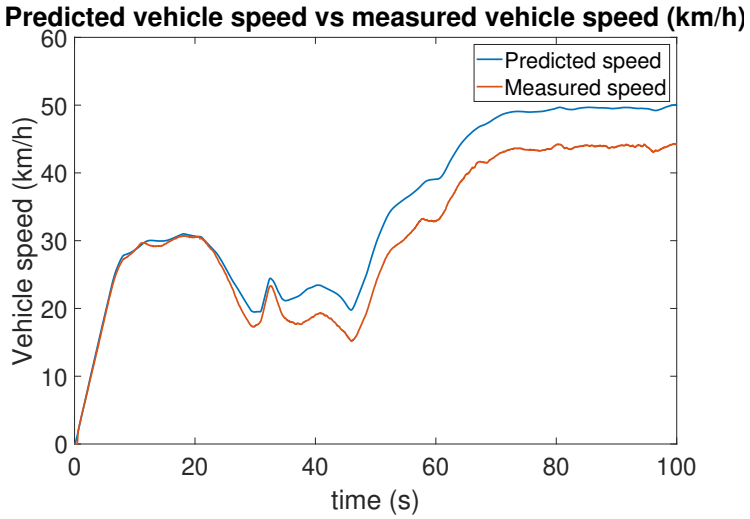


(b) Engine speed [rad/s]. Comparison of measured engine speed and simulated engine speed.

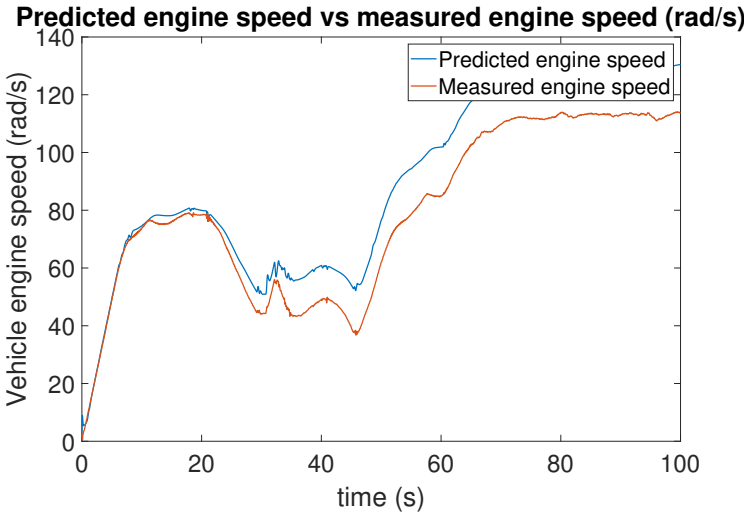


(c) Vehicle acceleration $[m/s^2]$. Comparison of differentiated vehicle speed to obtain vehicle acceleration, and simulated vehicle speed to obtain simulated vehicle acceleration.

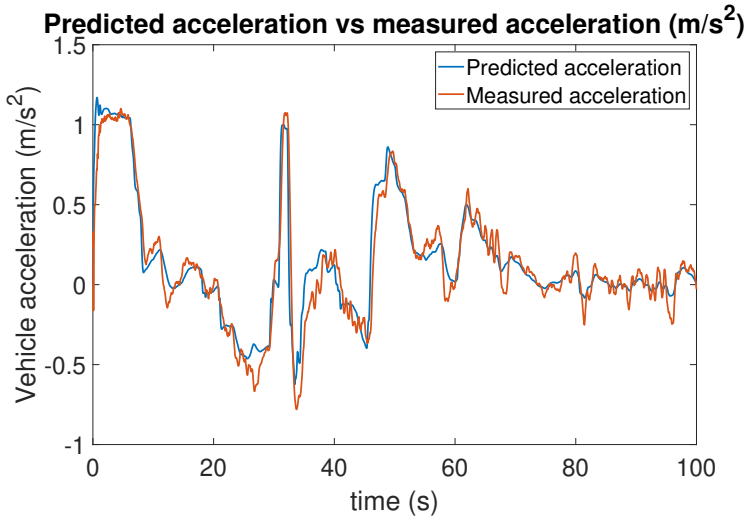
Figure 6.2: Comparison between measured and simulated data during driving opportunity 2



(a) Vehicle speed [km/h]. Comparison of measured vehicle speed and simulated vehicle speed.



(b) Engine speed [rad/s]. Comparison of measured engine speed and simulated engine speed.



(c) Vehicle acceleration $[m/s^2]$. Comparison of differentiated vehicle speed to obtain vehicle acceleration, and simulated vehicle speed to obtain simulated vehicle acceleration.

Figure 6.3: Comparison between measured and simulated data during driving opportunity 3

6.2.1 Smith predictor & measurement-updated Smith predictor

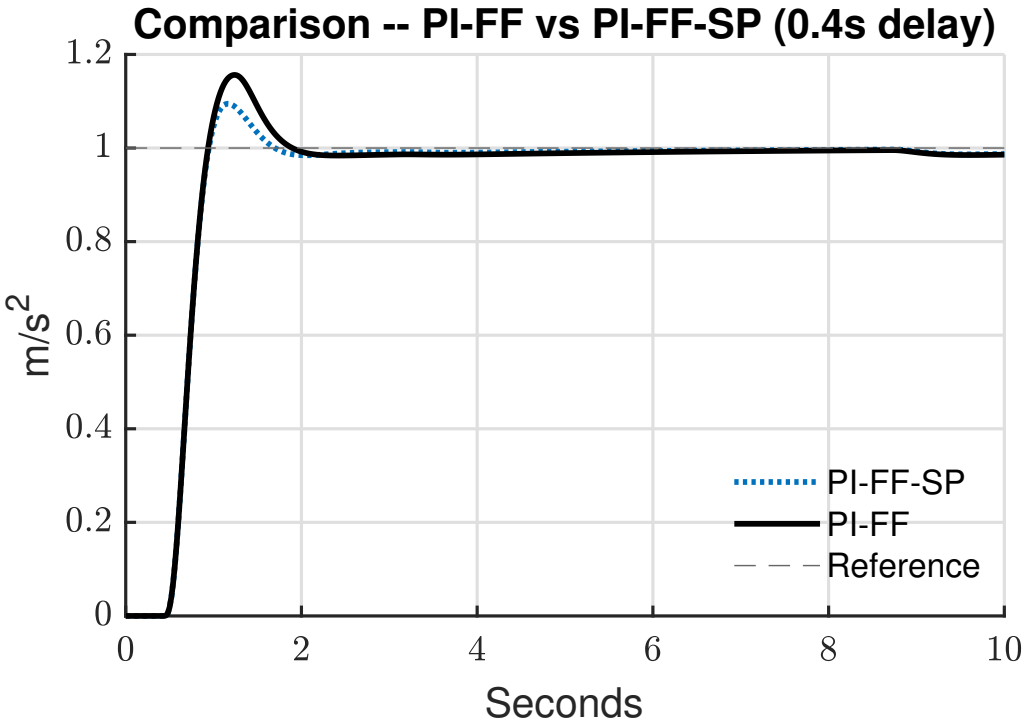
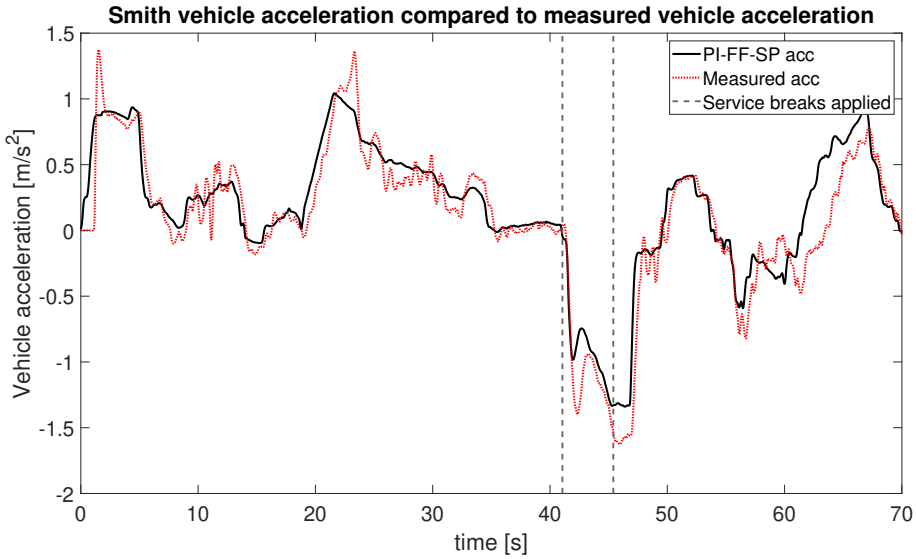


Figure 6.4: Comparison of PI-FF vs PI-FF-SP

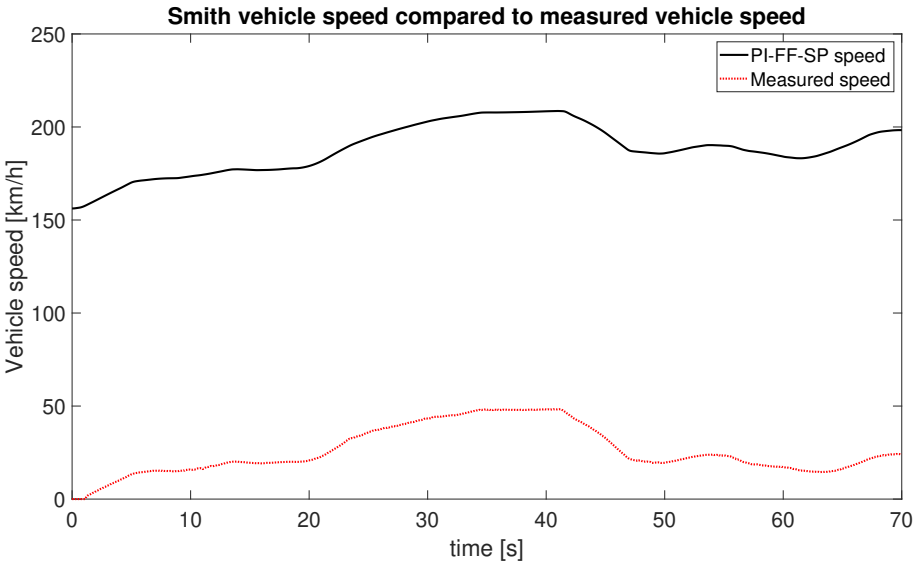
	Rise time [s]	Overshoot [%]	Settling time [s]	Steady-state error $[m/s^2]$
PI-FF	0.32	26.71	5.36	0.0141
PI-FF-SP	0.28	19.00	5.8	0.0117

The step-response of the PI-FF-SP in Figure 6.4 shows that, under ideal conditions, the Smith predictor lowers overshoot by 29 % compared to the PI-FF. Other metrics remain similar under ideal simulation conditions.

To further test and validate the Smith predictor, it was also evaluated in-vehicle. To enable validation, data from the current solution and from the implemented Smith predictor were logged.



(a) Acceleration PI-FF (measured) vs PI-FF-SP (Smith predictor)

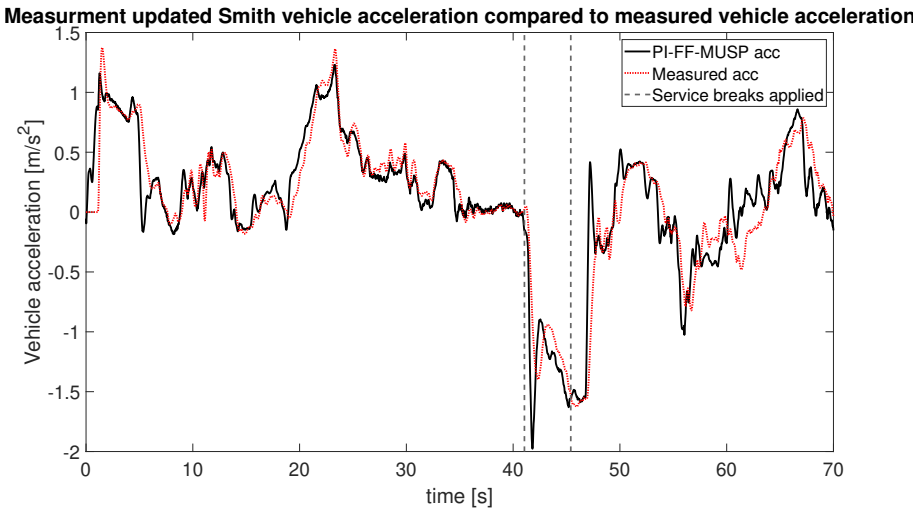


(b) Speed PI-FF (measured) vs PI-FF-SP (Smith predictor)

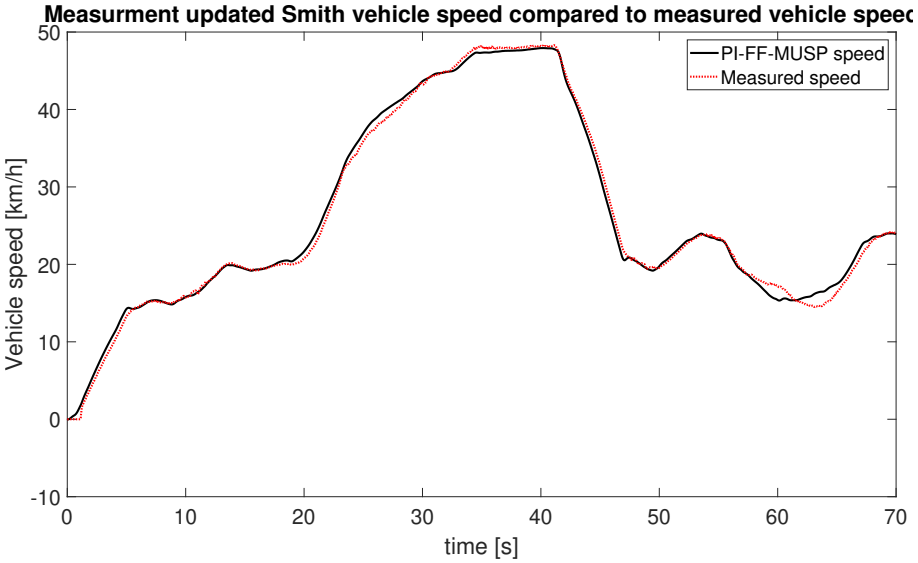
Figure 6.5: In-vehicle testing: PI-FF vs PI-FF-SP Smith predictor

	Cross corr estimated delay [s]	RMSE $[m/s^2]$	R^2
Measurement updated Smith	0.3	0.1797	0.8627
Regular Smith	0.26	0.8627	0.4482

Table 6.1: Cross-correlation-based comparison of the standard and measurement-updated Smith predictors.



(a) Acceleration PI-FF (measured) vs PI-FF-MUSP (Measurement-updated Smith predictor)



(b) Speed PI-FF (measured) vs PI-FF-MUSP (Measurement-updated Smith predictor)

Figure 6.6: In-vehicle testing: PI-FF vs PI-FF-MUSP Measurement-updated Smith predictor

When comparing the PI-FF-SP to PI-FF, it can be seen that it performs adequately in capturing overall acceleration dynamics, see Figure 6.5a. Certain smaller jerks

are not captured, but the main dynamics of the acceleration are captured. Although acceleration is the primary metric considered in this thesis, a comparison of speed is also made, as can be seen in Figure 6.5b. Speed remains an important measure, as the model outputs speed, which is then differentiated to obtain acceleration. The speed comparison shows a noticeable discrepancy due to significant drift. However, the system dynamics are still captured, so the acceleration is represented relatively well.

The measurement-updated Smith predictor significantly improves the prediction accuracy compared with the standard Smith predictor. The RMSE is reduced from 0.8627 to 0.1797 m/s^2, corresponding to a reduction of approximately 79.2%, whilst R^2 increases from 0.4482 to 0.8627. These results indicate that the measurement-updated Smith predictor captures the measured acceleration dynamics more accurately. This improvement is evident in the acceleration by capturing smaller jerks, as shown in Figure 6.6. In terms of speed, the improvement is large, with the drift barely noticeable.

To obtain quantifiable results, cross-correlation was performed between the predicted and measured accelerations. The predicted acceleration signal was shifted by the lag identified in the analysis chapter via cross-correlation, after which RMSE and R^2 were computed to evaluate prediction accuracy. Furthermore, another cross-correlation was performed to assess the actual lead that the measurement-updated Smith predictor achieved. The results can be seen in Table 6.1, Looking at the table the measurement updated Smith predictor shows lower RMSE and higher R^2 values as well as a higher lead. Meaning it outperforms the Standard Smith predictor in every measure. It can also be said that implementing a Smith predictor, especially a measurement-updated Smith predictor, can improve control performance by reducing the effects of delays.

6.2.2 LQG

The ILQG controller is evaluated through simulation and compared with the existing PI-FF controller. No real vehicle testing or hardware simulations were performed for the ILQG controller due to the complexity and development effort required to safely integrate the control structure into a vehicle. The simulations therefore use the vehicle model developed in the modeling section as the plant.

As an initial comparison, a step-response simulation is performed. This provides a baseline test from which key performance metrics such as rise time, overshoot, settling time, and steady-state error can be evaluated. The resulting performance measures are presented in Figure 6.7 and summarized in Table 6.2. The step-response simulation shows that the ILQG controller achieves lower overshoot, faster settling time, and a smaller steady-state error than the PI-FF controller. These improvements are obtained at the cost of a longer rise time.

To evaluate the controllers in a realistic driving scenario, a reference signal recorded

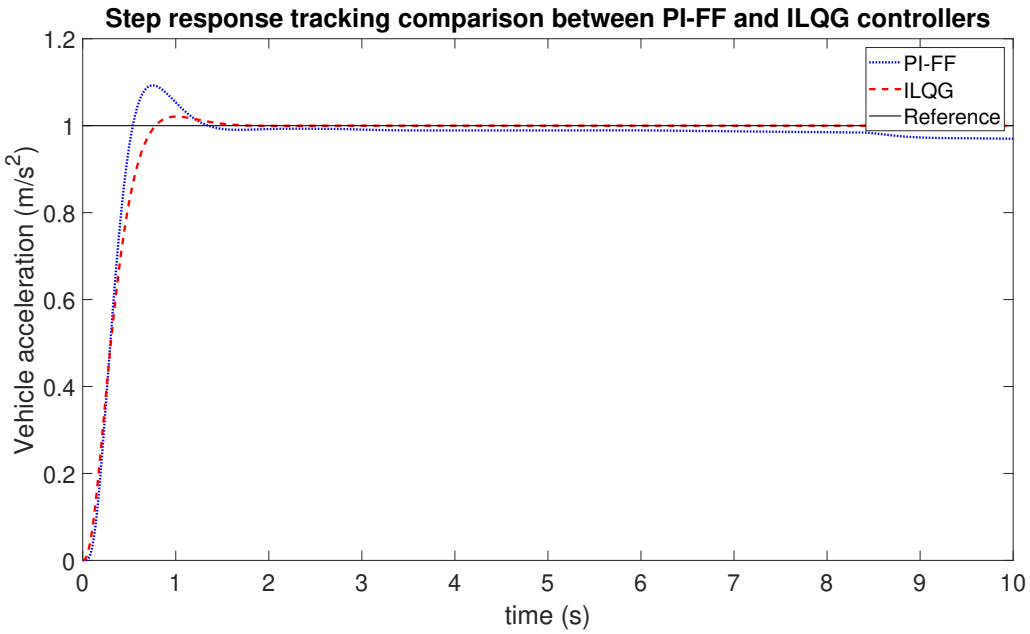


Figure 6.7: Step response tracking comparison between PI-FF and ILQG controllers

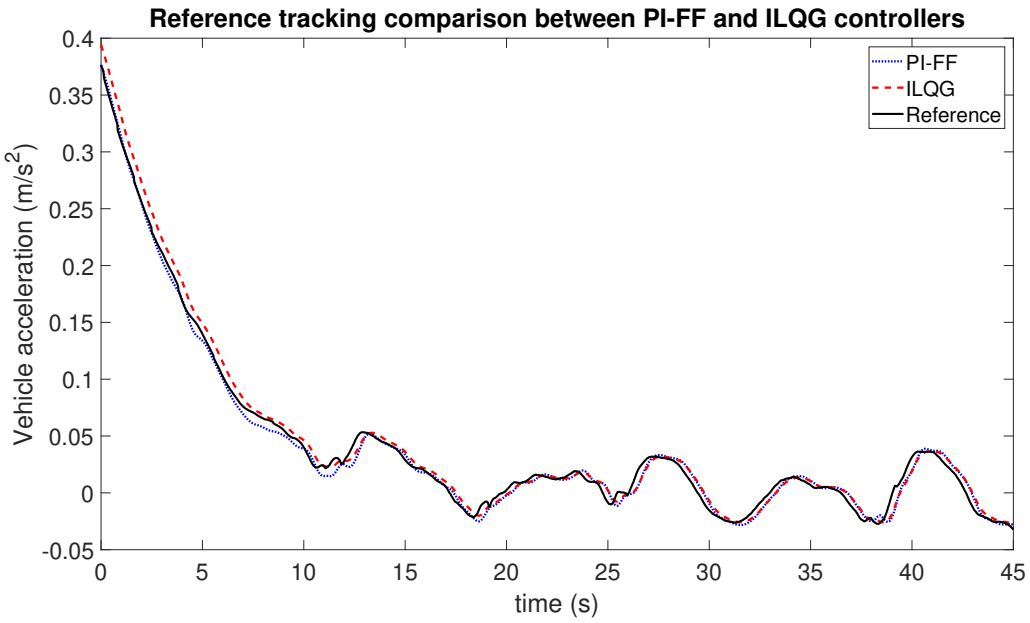


Figure 6.8: Reference tracking comparison between PI-FF and ILQG controllers

	Rise time [s]	Overshoot [%]	Settling time [s]	Steady-state error $[m/s^2]$
PI-FF	0.32	9.1982	1.18	0.0165
ILQG	0.44	2.1036	1.08	$2.3 \cdot 10^{-14}$

Table 6.2: Step response performance measurements for PI-FF and ILQG

	RMSE (m/s^2)
PI-FF	0.0059
ILQG	0.0079

Table 6.3: Root mean square error for PI-FF and LGQ+I

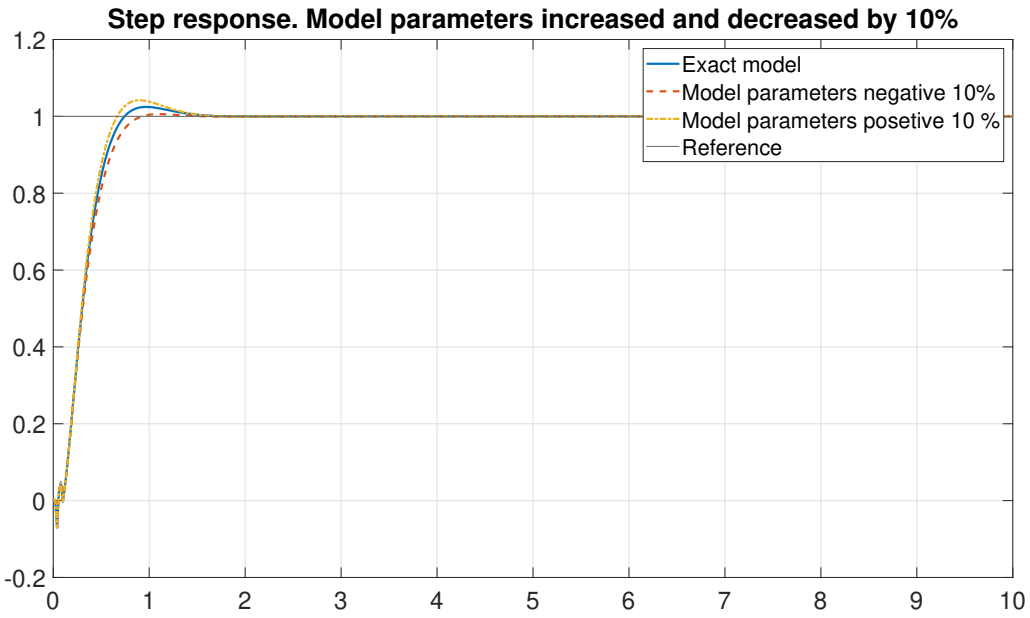


Figure 6.9: Step response with altered model parameters to demonstrate model mismatch. The parameters are faulty by a 10 percent increase and decrease.

while the vehicle is driving is used as the acceleration reference for the simulations. Figure 6.8 shows the resulting responses. Visually, the two controllers' responses are very similar and difficult to distinguish. Therefore, the root mean square error (RMSE) relative to the reference signal is computed and presented in Table 6.3. The RMSE values show that both controllers achieve similar reference-tracking performance. The PI-FF controller achieves a slightly lower RMSE of 0.0059 m/s^2 compared to 0.0079 m/s^2 for the ILQG controller. Despite this difference, both controllers track the reference closely throughout the simulation.

6.2.3 Model mismatch

To evaluate the robustness of the ILQG controller under model mismatch, a separate test was conducted. Here, the model parameters were altered uniformly by $\pm 10\%$, and the results are shown in Figure 6.9. Since the gear ratio estimate is known to be accurate, it is not altered by this test.

7
Discussion

This chapter discusses the results obtained and the methods used throughout the thesis. It presents an analysis and discussion of the modeling results and approach, an evaluation and discussion of the Smith predictors' performance, as well as an assessment of the LQG controllers' design tuning. Analyzes and discusses the modeling results and approach, evaluates the Smith predictors' performance, and assesses the LQG controllers' design, tuning, and results. The thesis also presents recommendations for future work based on its findings.

7.1 Model

The model presented in the Results chapter captures the dominant vehicle dynamics and is therefore sufficiently accurate for both system analysis and model-based control design. Nevertheless, some limitations do remain and should be acknowledged. The most significant limitation is the gradual deviation between the simulated and measured vehicle states as the simulation progresses. This deviation is particularly noticeable during deceleration, where the model tends to predict a lower acceleration magnitude than the measurements show. This behavior suggests that either the model does not fully represent certain deceleration dynamics, or the model inputs do not accurately reflect the vehicle's true inputs.

One possible source of error is uncertainty in the model inputs. Even though the engine torque, brake torque, and resistance torque are treated as measured signals, they are, in practice, estimates of the true physical quantities. The engine and brake torques originate from their respective control systems and are estimated as applied torques rather than measured directly. The same applies to the resistance torque measurement, which is estimated using models such as equation 2.16 and is therefore also subject to modeling inaccuracies. Small errors in these estimated inputs may accumulate over time, resulting in the speed deviations observed during validation.

Another possible explanation is that the model structure does not capture some vehicle dynamics. The validation vehicle is a BEV that utilizes regenerative braking during deceleration. During regenerative braking, the electric machine acts as a generator, producing braking torque while recovering energy. This operating mode differs fundamentally from propulsion and may introduce dynamics and losses that are not captured by the current model. Because the model structure follows a conventional ICE driveline representation, assumptions about damping, friction,

and torque transmission may be less accurate under regenerative braking. This hypothesis is supported by the observation that the model captures vehicle dynamics with high accuracy during acceleration and periods of near-constant speed, whereas the largest inconsistencies occur during deceleration.

Despite these deviations, the validation results demonstrate that the model successfully captures the dominant driveline dynamics. Although vehicle and engine speed predictions gradually drift from the measurements, the acceleration response remains in close agreement throughout most simulations. Because the model is intended for system analysis, simulation, and controller development rather than long-term open-loop prediction, the observed deviations are acceptable. The results therefore indicate that a third-order model with engine speed, vehicle speed, and drive axle torsion as states provides a suitable balance between model simplicity and dynamic accuracy.

Possible future areas of work have been identified. The model has only been validated for a specific vehicle configuration. To enable application across a wider range of vehicles with different masses, driveline layouts, and gear ratios, additional validation is required. Investigating automated parameter identification methods could improve the model's adaptability across different vehicle setups.

Another area for future work is validation at very low vehicle speeds (below approximately 5 km/h). Due to limited measurement accuracy and increased nonlinear effects in this operating region, the model performance at low speeds has not been properly evaluated.

Finally, the deviations observed during deceleration suggest that a more detailed representation of regenerative braking dynamics could further improve model accuracy.

7.2 Smith predictor

In the state-of-the-art controller investigation, it was suggested that the oscillations observed in the system could also be caused by delays in the acceleration controller, rather than originating solely from tire dynamics and driveline windup. Hence, a delay estimation was performed, and a Smith predictor was introduced. As mentioned in Chapter 4, the Smith predictor is highly effective when handling pure delays in a system.

Initially, step-response tests were conducted with the Smith predictor, some of which are shown in Chapter 4, Figures 4.4-4.6, to validate the concept. The challenge with delayed measurements is that effective control becomes harder, often resulting in over- and undershoots, or oscillations. The Smith predictor appeared very promising during the initial validation phase, demonstrating much faster response than the system without it whilst still maintaining good tracking performance.

Based on these results, it was decided to implement the Smith predictor in the current in-vehicle code to assess its performance in a real-world scenario. As shown in the Results chapter, the Smith predictor did not perform as expected, exhibiting relatively large errors compared to the baseline measured acceleration. One challenge during model and controller investigation and validation was the lack of ground-truth data. Since no ground truth exists, it was reasoned that the current acceleration controller captures the acceleration dynamics sufficiently and was therefore selected as the baseline.

Another limitation of the Smith predictor is that many of the smaller jerks are not captured. This is most likely due to the significant drift occurring in the speed estimate, which makes it less sensitive to these smaller changes. As a result, the estimated acceleration does not capture these smaller changes. Capturing these smaller changes is important for obtaining accurate acceleration responses.

The measurement-updated Smith predictor captures the small jerks, as shown in Figure 6.6a. This is because speed tracking has improved greatly, which in turn provides much more accurate inertia estimates and captures acceleration dynamics much better.

One issue with the measurement-updated Smith predictor, as shown in Figure 6.6a, is significant overshoot during large jerks. One potential explanation is that delayed measurements cause the prediction to overreact during these large jerks, resulting in increased overshoot. One more potential cause is the use of service brakes.

7.3 LQG

The integral-supported LQG controller showed promising performance in simulation when compared to the existing PI+FF controller. As shown in Figure 6.8, both controllers achieve comparable reference-tracking performance during a realistic driving scenario. Although the PI+FF controller achieves a slightly lower RMSE than the ILQG controller, the difference is small, indicating that the proposed controller can maintain a similar level of tracking performance.

The most significant improvements appear in the step-response simulations in Figure 6.7. Compared to the PI+FF controller, the ILQG controller reduces overshoot from 9.2% to 2.1%, achieves a faster settling time, and eliminates steady-state error by including integral action. These improvements come at the cost of a somewhat longer rise time. The longer rise time likely results from the more conservative control action produced by the optimal feedback design, which prioritizes damping and smooth system behavior over aggressive reference tracking.

The reduced overshoot and improved damping characteristics suggest that the ILQG controller may also reduce oscillations within vehicle speed and acceleration during operation. Because the acceleration reference varies significantly in real driving, a controller with lower overshoot is expected to reduce oscillations. Reduced oscilla-

tions in vehicle speed and acceleration are generally desirable, as they can improve vehicle comfort and drivability. However, no direct measurement or quantification of a controller with lower overshoot has been performed, therefore, this potential benefit cannot be conclusively verified from the presented results.

The robustness of the ILQG controller was evaluated in Chapter 5 and further assessed through simulations with model mismatch, as shown in Figure 6.9. The results indicate that the controller can incorporate robustness while still achieving satisfactory performance. However, the trade-off between robustness and performance depends heavily on the specific application. Establishing a robust controller that performs well across a range of heavy-duty vehicle configurations and operating environments would require more extensive testing under varying vehicle setups and environmental conditions.

All evaluations of the ILQG controller are based solely on simulation using the developed vehicle model. Consequently, interpret the results as an indication of potential performance rather than proof of real-world improvements. While simulations provide a controlled environment for comparing control strategies under identical conditions, the conclusions' accuracy depends entirely on whether the model accurately represents reality under all circumstances.

Model accuracy matters most for the ILQG controller because it relies on both optimal state feedback and state estimation through a Kalman filter. While the Kalman filter improves robustness to measurement noise and sensor imperfections, model inaccuracies can introduce steady-state errors and degrade state estimates, which in turn affect the resulting control actions. Furthermore, variations in vehicle mass, driveline configuration, gear selection, and other conditions can alter system dynamics and may require retuning the LQR and Kalman filter gains to maintain the desired balance between robustness and performance.

Testing shows that the existing PI+FF controller exhibits more tracking error and oscillatory behavior than predicted by the simulation. This indicates that the model does not capture all real-world dynamics, so the results may underestimate the actual benefit of improved damping. Nevertheless, without implementing and testing the ILQG controller on a real vehicle, it cannot be determined whether the simulation improvements translate directly to vehicle operation.

Overall, the results indicate that the ILQG controller offers potential for improved transient response while maintaining accurate tracking performance. Further work involving implementation and validation on a real vehicle is required to determine the practical benefits of the proposed control structure.

7.4 Comparison of Control Strategies

This project proposes three control strategies: PI-FF-SP, PI-FF-MUSP, and ILQG. Since the ILQG controller has not been implemented or tuned for the real vehicle,

a direct quantitative comparison with the other strategies is not entirely fair, as PI-FF-SP and PI-FF-MUSP were tuned based on real-world vehicle testing. However, it is still possible to compare the potential benefits and drawbacks of the different control strategies based on their structures, theoretical properties, and simulation results.

7.4.1 PI-FF-SP

The PI-FF-SP strategy offers a relatively simple control structure, making it easy to implement, tune, and maintain. From an industrial perspective, this simplicity is desirable because it reduces implementation complexity and supports future development and maintenance.

However, the performance of the PI-FF-SP strategy depends on accurate knowledge of the system delay and a model that accurately captures the real vehicle's behavior. Deviations between the assumed and actual delay, or model inaccuracies, may therefore negatively affect prediction and overall control performance.

7.4.2 PI-FF-MUSP

The PI-FF-MUSP strategy addresses some weaknesses of the PI-FF-SP strategy by improving prediction and increasing robustness to drift and offset errors. By adding state estimation, the controller can account for disturbances and model deviations that might otherwise reduce PI-FF-SP performance.

However, adding state estimation also increases the control system's complexity. The state estimator's performance depends on the accuracy of the system model and the available measurements. Furthermore, a trade-off exists between estimation convergence and sensitivity to measurement noise, where faster observer dynamics may improve the reaction time but increase sensitivity to noise.

7.4.3 ILQG

Theoretically, the ILQG strategy offers the most promising control, as the controller design allows selecting a control input based on a trade-off between tracking performance and control effort. The simulation results also show reduced overshoot compared with the other control strategies, which is desirable for this project's system.

However, unlike the PI-FF-SP and PI-FF-MUSP strategies, the ILQG controller has not been implemented or verified in a real vehicle, and its real-world performance remains uncertain. Consequently, we can draw only limited conclusions from the ILQG performance. Furthermore, the ILQG strategy, like the PI-FF-MUSP, introduces state estimation, which introduces the same drawbacks that the PI-FF-MUSP suffers from.

7.4.4 Recommended Control Strategy

Based on the results obtained throughout this project, the PI-FF-MUSP strategy is considered the most suitable control strategy for practical implementation. Compared with the PI-FF-SP strategy, it provides improved prediction and increased robustness against drift and offset errors. At the same time, it maintains a relatively simple control structure compared with the ILQG strategy.

Although the ILQG strategy may demonstrate promising performance in simulation, including reduced overshoot, it has not been implemented or verified in a real vehicle. Its practical performance therefore remains uncertain, and further tuning and validation would be required before its potential benefits can be fully evaluated.

The PI-FF-MUSP strategy, by contrast, has been evaluated in a real vehicle and has shown improved performance while maintaining a reasonable level of complexity. Introducing state estimation increases the complexity of the control structure and makes it sensitive to model inaccuracies and measurement noise. However, considering the observed improvements in performance and robustness, these drawbacks are considered acceptable. Therefore, among the control strategies investigated in this project, the PI-FF-MUSP strategy offers the best balance of performance, robustness, complexity, and practical applicability.

8

Conclusion

The research questions aimed to investigate how model-based control can improve acceleration-tracking performance and to determine which driveline relationships are required to obtain a sufficiently accurate vehicle model. The modeling results show that, with appropriate parameters, a third-order state-space representation can capture the dominant driveline and vehicle dynamics. Furthermore, by incorporating measurement updates, the model can compensate for small deviations and provide accurate estimates of both driveline states and vehicle behavior.

The control results are more difficult to interpret conclusively. Two control structures were investigated, a Smith predictor intended to support and improve the existing acceleration controller, and an integral-supported LQG controller designed as a model-based alternative. The Smith predictor showed promising results in simulation, where compensation for measurement delays reduced the negative effects associated with delayed feedback. However, in the real-vehicle implementation, no actual ground truth could be provided, and it is therefore not possible to evaluate if the Smith predictor with the estimated delay could provide an improvement in performance.

The ILQG controller demonstrated the most promising results from a control perspective. Simulation studies showed improved transient behavior and reduced overshoot, and maintained reference-tracking performance compared to the current PI+FF controller. Nevertheless, these results depend entirely on the accuracy of the developed vehicle model and have not yet been experimentally validated. Consequently, real-vehicle testing is needed to verify the controller's practical benefits.

Overall, the results indicate that model-based methods have significant potential for improving acceleration control performance. The developed model provides a suitable foundation for analysis and controller design, while the ILQG and proportional-integral feedforward measurement-updated Smith predictor (PI-FF-MUSP) controllers demonstrate the potential advantages of model-based control strategies in driveline applications. Future work should improve model accuracy during deceleration, extend validation across different vehicle configurations, and implement the proposed controllers in a real-vehicle environment to assess practical performance and robustness.

Bibliography

[1] T. Glad and L. Ljung, *Control Theory: Multivariable and Nonlinear Methods*. Taylor & Francis Ltd, 2000.

[2] M. Pettersson, *Driveline Modeling and Control*. Dissertation no. 484, linköping studies in science and technology, Linköping University, Linköping, Sweden, 1997.

[3] J. Fredriksson, H. Weiefors, and B. Egardt, "Powertrain control for active damping of driveline oscillations," *Vehicle System Dynamics*, vol. 37, no. 5, pp. 359–376, 2002.

[4] G. Pechlivanidou and N. Karampetakis, "Zero-order hold discretization of general state space systems with input delay," *IMA Journal of Mathematical Control and Information*, vol. 39, pp. 708–730, June 2022.

[5] T. Glad, L. Ljung, and M. Enqvist, *Reglerteknik: grundläggande teori*. Lund: Studentlitteratur, 5 ed., 2024.

[6] K. J. Åström and T. Hägglund, *Advanced PID Control*. International Society of Automation, 2006.

[7] K. J. Åström, "Model uncertainty and robust control," *Department of Automatic Control, Lund University*, 2000.

[8] S. Das and K. Halder, "Missile attitude control via a hybrid lqg-ltr-lqi control scheme with optimum weight selection," in *2014 First International Conference on Automation, Control, Energy and Systems (ACES)*, pp. 1–6, 2014.

[9] F. Huerta, D. Pizarro, S. Cobreces, F. J. Rodriguez, C. Giron, and A. Rodriguez, "Lqg servo controller for the current control of lcl grid-connected voltage-source converters," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4272–4284, 2012.

[10] K. J. Åström and L. Rundqwist, "Integrator windup and how to avoid it," in *Proceedings of the American Control Conference (ACC)*, (Pittsburgh, PA, USA), pp. 1693–1698, June 1989.

[11] F. S. Barbosa, G. P. das Neves, and B. A. Angélico, "Discrete lqg/ltr control augmented by integrators applied to a 2-dof helicopter," in *2016 IEEE Conference on Control Applications (CCA)*, pp. 1238–1243, 2016.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY