Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2017

Enabling Testing of Lateral Active Safety Functions in a Multi-Rate Hardware in the Loop Environment

Fredrik Björklund and Elin Karlström

LINKÖPING
UNIVERSITY

Master of Science Thesis in Electrical Engineering

Enabling Testing of Lateral Active Safety Functions in a Multi-Rate Hardware in the Loop Environment

Fredrik Björklund and Elin Karlström

LiTH-ISY-EX--17/5080--SE

Supervisor: **Mahdi Morsali**
isy, Linköping University
Nadeem Afzal
Volvo Cars

Examiner: **Erik Frisk**
isy, Linköping University

Division of Vehicular Systems
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

As the development of vehicles moves towards shorter development time, new ways of verifying the vehicle performance is needed in order to begin the verification process at an earlier stage. A great extent of this development regards active safety, which is a collection name for systems that help both avoid accidents and minimize the effects of a collision, e.g brake assist and steering control systems. Development of these active safety functions requires extensive testing and verification in order to guarantee the performance of the functions in different situations. One way of testing these functions is to include them in a Hardware in the Loop simulation, where the involved hardware from the real vehicle are included in the simulation loop.

This master thesis investigates the possibility to test lateral active safety functions in a hardware in the loop simulation environment consisting of multiple subsystems working on different frequencies. The subsystems are all dependent of the output from other subsystems, forming an algebraic loop between them. Simulation using multiple hardware and subsystems working on different frequencies introduces latency in the simulation. The effect of the latency is investigated and proposed solutions are presented. In order to enable testing of lateral active safety functions, a steering model which enables the servo motor to steer the vehicle is integrated in the simulation environment and validated.

Acknowledgments

First of all, we would like to thank Volvo Cars AB for giving us the opportunity to perform this master thesis. A special thanks to our supervisor Nadeem Afzal, our co-supervisor Siddhant Gupta and the HIL group. Furthermore, we also want to thank Weitao Chen for sharing your knowledge within coupling simulations.

We would also express our gratitude towards our examiner Erik Frisk and supervisor Mahdi Morsali at Linköping University. They have always been available for both encouragement and good discussions.

Finally, we would like to thank our friends and family for their love and support.

Göteborg, June 2017
Fredrik Björklund and Elin Karlström

Contents

Notation **ix**

1 Introduction **1**
1.1 Background . . . 1
1.2 Problem Formulation . . . 1
1.3 Purpose and Goal . . . 2
1.4 Hardware In the Loop . . . 2
1.5 Related Research . . . 3
1.5.1 Coupled Systems . . . 3
1.6 Contributions . . . 3
1.7 Thesis Outline . . . 4

2 Simulation Environment **5**
2.1 Model In the Loop . . . 5
2.2 Software In the Loop . . . 5
2.3 Real-time simulation . . . 6
2.4 Coupling Simulation . . . 6
2.5 Multi-rate Simulations . . . 7
2.5.1 Multi-rate Methods . . . 7

3 Steering Theory **9**
3.1 Steering System . . . 9
3.1.1 Movement of the Steering Rack . . . 10
3.1.2 Steering Controller . . . 10
3.2 Bicycle Model . . . 11
3.3 Lateral Active Safety . . . 13

4 Integration of a Steering Model in the HIL-Environment **15**
4.1 Simulation Environment . . . 15
4.1.1 Communication . . . 17
4.2 Steering Model . . . 18
4.2.1 Manual Gear . . . 18
4.2.2 Servo Gear . . . 18

4.2.3 Steering Rack 19
4.2.4 Power Steering Control Module 19
4.2.5 Servo Motor Model 20

5 Evaluation of Steering Behaviour 21
5.1 Steering Model in open-loop 21
5.1.1 Manual Gear & Rack without feedback 21
5.1.2 Manual Gear & Rack 23
5.1.3 Servo Gear & Rack without feedback 25
5.1.4 Servo Gear & Rack 26
5.2 Steering model in closed-loop 27
5.2.1 Sinusoidal input signals without driver 28
5.2.2 Real measurement data without driver 29
5.2.3 Simulation with driver 31
5.3 Observations 32

6 Improvements 33
6.1 Latency in Simulation 33
6.1.1 Rod Force 33
6.1.2 Prediction Trajectory 34
6.2 Multi-Rate Simulation 35
6.2.1 Discrete Averaging Filter 35
6.2.2 Low Pass and Lead Compensation Filter 36

7 Result and Discussion 41
7.1 Filter 41
7.2 Rod Force 43
7.3 Complete HIL Simulation with Driver 44
7.4 Stability Analysis of Feedback System with Delay 48
7.5 Discussion 51

8 Conclusions and Future Work 53

Bibliography 55

Index 57

Notation

Notations Bicycle Model

Notation	Meaning
Ω_z	Yaw angle [deg]
$\dot{\Omega}_z$	Yaw rate [deg/s]
δ_f	Steering angle [deg]
m	Vehicle mass [kg]
V_x	Velocity in local x-direction [m/s]
V_y	Velocity in local y-direction [m/s]
F_{yr}	Lateral rear force [N]
F_{yf}	Lateral front force [N]
F_{xf}	Longitudinal front force [N]
I_z	Inertia [kg $\cdot m^2$]
α_f	Slip angle front [deg]
α_r	Slip angle rear [deg]
L_1	Distance between front axle and center of gravity of the vehicle [m]
L_2	Distance between rear axle and center of gravity of the vehicle [m]
$C_{\alpha f}$	Cornering stiffness of front tire [N/deg]
$C_{\alpha r}$	Cornering stiffness of rear tire [N/deg]

Notations Steering System

Notation	Meaning
Θ_{sw}	Steering wheel angle [deg]
Θ_p	Pinion angle [deg]
τ_{TB}	Torsion bar angle [deg]
F_{mech}	Mechanical force [N]
F_{servo}	Servo force [N]
F_{rod}	Rod force [N]
x_r	Position rack [m]
m_r	Steering inertial mass [kg]

Notations Filter

Notation	Meaning
w_c	Cut-off frequency [Hz]

Abbreviations

Abbreviation	Meaning
ARMA	Auto-regressive moving average
PID	Proportional, integral, differential (regulator)
HIL	Hardware In the Loop
MIL	Model In the Loop
LKA	Lane Keeping Assist
ECU	Electric Control Unit
STM	Steering Torque Manager
SIL	Software in the Loop
ZOH	Zero-Order Hold
FOH	First-Order Hold
ASDM	Active Safety Domain Module
EPAS	Electric Powered Assisted Steering
PSCM	Power Steering Control Module

1

Introduction

1.1 Background

The automotive industry moves towards shorter development times. In some ways this is caused by the speed-up development of autonomous driving and electrification of vehicles due to competition between major car producers. A big part of this regards the development of active safety functions, which aims to increase the safety of the vehicle in various driving scenarios. These complex functions and new techniques needs extensive testing to insure relevant legislation. A great part of the testing is currently done in real test vehicles in order to investigate the functions in a real environment. However, this contributes to longer and more inflexible development processes. New simulation environments is one solution to shorten the time span and enable production of advanced and safe vehicles. This master thesis aims to evaluate and improve the lateral steering behaviour of a Hardware in the Loop (HIL) - environment within Active Safety at Volvo Cars[1].

1.2 Problem Formulation

In today's development process some faults and bugs in the steering controller are found once it is implemented in a real vehicle, thus a large number of test vehicles are used to improve and verify the performance of the various controllers. This is a time consuming process which is both costly and can include driver biasness. In order to reduce the development time span a simulation environment is needed, which includes the hardware to be tested, a model for the vehicle as well as the road forces affecting the vehicle.

[1] www.volvocars.com

In this thesis, a more advanced steering model including models of the steering controller and servo motor are implemented in an already built simulation setup (HIL-environment). Including the controller in the steering system is a requirement in order to enable testing and evaluation of assist torque from a servo motor and lateral control. The HIL- environment will be analyzed, improved and evaluated. This will be done by investigating the vehicle dynamics, the setup of the environment and the simulations. The target HIL- environment will consist of hardware components (active safety domain module and front facing camera) as well as models (vehicle model, steering controller, servo motor, driver model and mechanical parts of the steering). This thesis aims to evaluate the HIL- environment in order to answer the following questions:

- With the developed and improved HIL-environment, can the HIL-environment be used for testing of the steering controller's performance? What further developments can be made?

- What are the challenges regarding time integration and simulation when developing a HIL-environment? What problems should be prioritized and what are the possible solutions?

1.3 Purpose and Goal

The purpose of this thesis is to evaluate and improve the steering behaviour to make the simulation in the HIL-environment more realistic. The goal is to include a servo model in order to analyze the torque provided to assist the driver during steering or when any active safety function demands a torque request. This in order to enable testing of the steering controller performance in an earlier stage of the development process. Further, this thesis will include an investigation on how the performance of a validated steering model in a simple test environment without hardware, differs from a more complex engineering system with more components and communication.

1.4 Hardware In the Loop

Hardware in the Loop (HIL) simulation is a type of simulation with at least one hardware from the real system included within the simulation loop. It is used in automotive Electric Control Unit (ECU) development due to the increased need to test functions earlier in the development process and to enable simulations of situations that in real cases could hurt equipment/people. The HIL environment can help create a more realistic simulation and therefore find errors earlier in the development process [1].

1.5 Related Research

HIL-environments have been used for verification and testing for a long time but most solutions are done in an ad hoc fashion. Due to the specific arrangement of the environment and simulations in this thesis, few research studies in this area have been carried out.

1.5.1 Coupled Systems

The research that has been done on modelling and simulation of complex engineering systems includes vehicles where components are modelled and connected to form a simulation environment which can emulate a real car. Most research discusses the integration of the different subsystems. In many cases the different subsystems are individually modelled for tailored requirements and when the global system is simulated as a whole, various problems emerge. One of the main problems with a simulation consisting of multiple subsystems working in different systems is the communication between them.

An example is that the time scales and frequencies usually are different for the physical components of the system. Hence, in the time integration of multiscale problems, the stepsize for the overall system is restricted by the time scale of the fastest subsystem. Commonly proposed solutions are various multi-rate methods, which for example uses signal-based extrapolation techniques [2, 3].

Another proposed solution is to use model-based coupling schemes instead of using the signal-based extrapolation approach to resolve the bidirectional dependencies between the involved subsystems. Model-based coupling schemes identifies models in order to predict the future behavior of the involved subsystems, in contrast to the signal-based approach that only uses known coupling signals. The model-based approach applies an adaptive adjustment of the extrapolated data as well as it compensate for the sending and receiving dead-times [4].

1.6 Contributions

The contributions developed in this thesis are described in this section.

- A basic platform for developing and testing of steering as well as lateral active safety functions.
- An extensive mapping and investigation of the HIL-environment with the implemented steering model. This included building an understanding of the problems Volvo Cars are facing regarding test environments with various subsystems.
- An investigation which shows what problems to prioritize, what kind of improvements that can be made and further how the solutions can be developed in the future.

1.7 Thesis Outline

The outline of the report is organized in several chapters. Chapter 2 explains the simulation environments used and analyzed in this thesis. This section also describes the positive and negative aspects of coupled subsystems used in complex engineering systems. The theory around the implemented steering model is described in chapter 3. Chapter 4 explains the implementations of the models and the various hardware in the HIL-environment. Chapter 5 contains evaluation of the implemented steering model in the HIL environment. The steering model is evaluated in open-loop and closed-loop simulations. The improvements made to the HIL-environment are described in chapter 6 and the results are presented in chapter 7. Finally, the conclusions and suggestions for future work is presented in chapter 8.

2

Simulation Environment

Simulation environments are used to move the testing of various functions and systems from the real environment into a computer based simulation in order to more effectively develop new products and techniques. The development of vehicles is a complex process where different environments and models are used to develop specific parts and functions. This chapter will first build an understanding about the other different simulation environments - Model in the Loop (MIL) and Software in the Loop (SIL). The main difference between these simulation environments and the HIL-environment can be seen in 2.1a, 2.1b and 2.1c. In regards to simulation with complex systems, this chapter describes the phenomenon of using coupling simulation and multi-rate simulations.

2.1 Model In the Loop

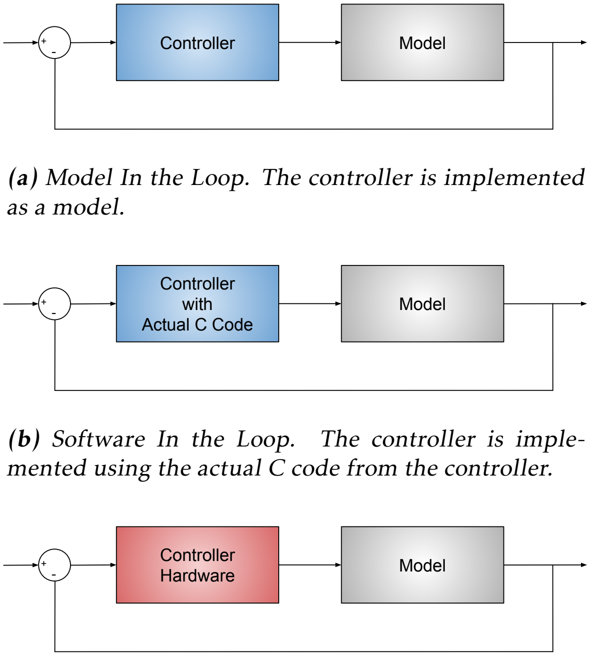
The MIL technique is used to test functions and software model architecture in a closed loop environment without hardware. The environment is built up by various mathematical models of the entire vehicle, for example the detailed physical models of the engine, the external environment and the operator of the vehicle [1]. Running simulations using only models, makes it possible to calculate the different states with the same frequency.

2.2 Software In the Loop

The goal of SIL is to enable simulation of the whole Electric Control Unit (ECU) and create a full simulation of an automotive electronics system. This setup is dependent on the actual ECU code and basic software (OS, models of the communication drivers, I/O etc.) [1].

2.3 Real-time simulation

Real-time simulation is used when simulating HIL since it enables running models and hardware at required speeds and with precise timing requirements between subsystems. Hence, the inputs and outputs in the virtual world is running at the same time as in the real world. In this way it is possible to test and investigate scenarios which can be complex and dangerous to perform with a real vehicle.



(a) *Model In the Loop. The controller is implemented as a model.*

(b) *Software In the Loop. The controller is implemented using the actual C code from the controller.*

(c) *Hardware In the Loop. The controller is implemented as the actual hardware.*

Figure 2.1: *A general overview of the difference between the simulation environments explained in the thesis - MIL, SIL and HIL.*

2.4 Coupling Simulation

Complex engineering systems like automobiles requires modelling of components from various engineering fields, e.g., mechanics and control. Due to this the global system is generally divided into several subsystems, which contributes with advantages like re-use of existing validated models, independent modelling within each subsystem and use of different software for each module.

However, simulation of the global system with this kind of coupled subsystems can cause unstable behaviour due to algebraic loops between the subsystems. This is created if any interconnections in the global system form a closed loop of subsystems, all of which have outputs which are explicitly dependent on the inputs. There are various coupling methodologies in co-simulation applications with different solutions.

When running a simulation with fixed step time for the solver, Simulink solves the algebraic loop with an iterative method automatically. In a real-time simulation a non-iterative method should be applied. This means that it concerns a non-iterative co-simulation, which for example can be solved with stepwise extrapolation of the coupling signals [5, 6].

2.5 Multi-rate Simulations

For a complex simulation environment, subsystems can be separated and solved using different solvers and step-sizes. The communication between the different subsystems usually occur with a specific time interval, also referred to as *macro steps*. This means the communication between the subsystems is restricted to discrete synchronization points.

During the time between one synchronization point and the next, the subsystems solves the internal dynamics in time steps referred to as *micro steps*. During the micro steps, no data exchange between the subsystems occur, which means the subsystems have no information about the states in the other subsystems. This means the signals entering a subsystem from another subsystem will be constant during one macro step. If the state signals in a subsystem is not available during a synchronization point, the signal will be kept the same as during the last synchronization point. [2].

2.5.1 Multi-rate Methods

In order to solve the issue with missing data between the synchronization points, or during one macro step, there are some suggested solutions. One of these methods is to use extrapolation techniques in order to predict the missing points [6]. These methods includes for example the well known Zero-Order Hold (ZOH) and First Order Hold (FOH). ZOH simply holds the last known value until a new value arrives, which means the signal is constant during one macro step. FOH takes the values from two macro steps and predict missing data using this information.

3

Steering Theory

More vehicles utilizes electric powered assist steering systems (EPAS) for steering assist. The EPAS provides extra force to the steering rack, which contributes to a decreased force required from the driver. The EPAS contains a steering controller which controls the desired steering characteristics. In this chapter the overall steering theory implemented in order to test the steering controller is described. To further understand the vehicle dynamics, a simple bicycle model is presented, which later on will be used for stability analysis. The active safety function Lane Keeping Aid (LKA) is described in order to explain how this function interacts with the steering of the vehicle through the servo motor.

3.1 Steering System

In Figure 3.1 a schematic picture of the electro-mechanical rack and pinion system is shown. The steering column transfers the torque from the steering wheel to the rack. The torque is calculated from a torsion bar angle measured with a sensor, which is connected between the pinion and the steering column. This torsion bar angle is a result of both the rack forces and the driver manoeuvres. In order to turn the wheels, the pinion transforms the rotational motion into a translation of the gear rack [7]. The translation is further reconverted into the rotation of the wheels with help from the steering arms, or tie rods. When the steering torque is transferred to the wheel, a steering angle is generated. The servo motor is connected to the steering rack through a belt to a ball nut-gear that translates the servo motor torque into a translational force acting on the rack. The torque from the servo motor is provided to assist the driver during steering or when any active safety function requests a torque.

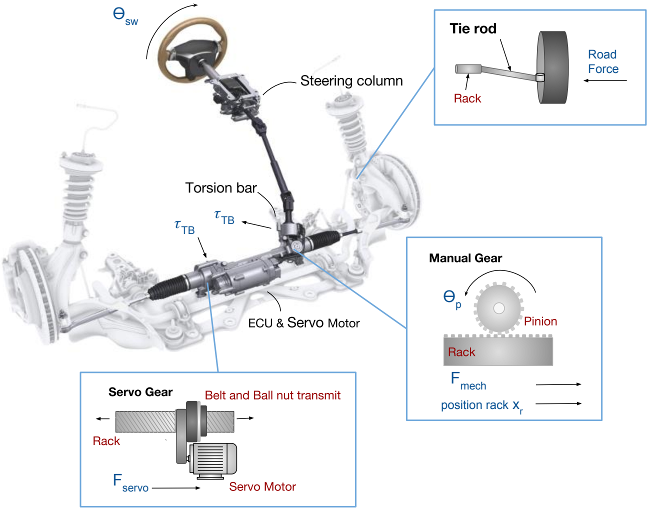


Figure 3.1: *A schematic picture of the electro-mechanical rack and pinion system [8].*

3.1.1 Movement of the Steering Rack

The movement of the steering rack is calculated using Newtons second law. The equation is expressed as

$$m_r \ddot{x}_r = F_{servo} + F_{mech} - F_{road}. \tag{3.1}$$

Where m_r and x_r is the steering inertial mass and the position of the rack, F_{servo} the force on the rack from the servo motor, F_{mech} the force from the driver and F_{road} is the road forces at the tires translated to a corresponding force at the rack through the tie rods.

3.1.2 Steering Controller

An overview of the steering controller including the input and output signals is shown in Figure 3.2. The steering controller calculates a requested servo motor torque with regards to both the requested pinion angle and how much servo motor torque is needed to assist the driver. The inputs are the torsion bar angle from the driver action, the steering angle from the vehicle state and the pinion angle request from the lateral active safety controller.

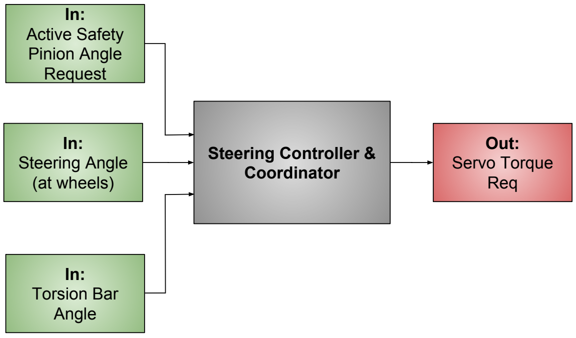


Figure 3.2: *A schematic picture of the steering controller and coordinator together with the input and output signals.*

3.2 Bicycle Model

As the complexity of the simulation environment is far to great to be able to do any analytically analysis, a simplified vehicle model is used for this purpose. In Figure 3.3, a simplified vehicle model called the bicycle model is presented.

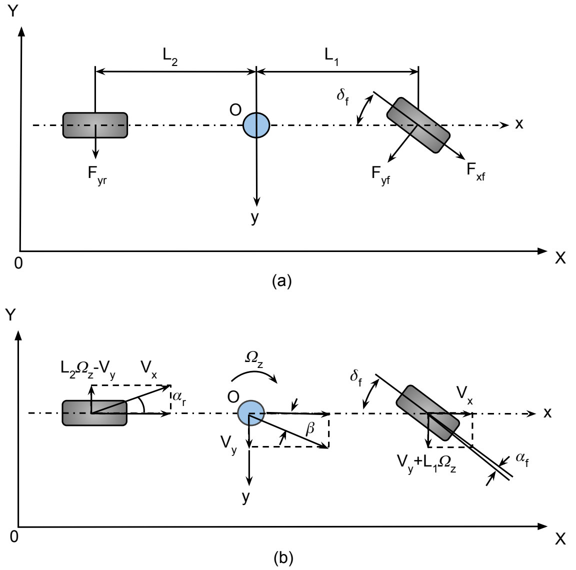


Figure 3.3: *A bicycle model for analysis of transient motion.*

The bicycle model is a dynamic vehicle model with two degrees of freedom that models the vehicle with two wheels instead of four. The model provides a mathematical description of the lateral vehicle motion and is for example used for stability analysis and validation of filter performance. The two degrees of freedom are the lateral position y and the yaw angle Ω_z.

In this thesis a bicycle model in transient motion is analyzed, thus the state after the steering input and before the vehicle reaches a steady state motion. During a turning manoeuvre the vehicle is in both translation as well as rotation. Therefore, to analyze this state the inertial properties need to be taken into account. A simplified way to describe this motion is to use a set of axes fixed to and moving with the vehicle body. In this way the moments of inertia of the vehicle is constant [9].

The simplified model of the vehicle is assumed not to be accelerating or decelerating along the local x axis. Referring to Figure 3.3(a), the equations used with the small angle assumptions are given by [9]

$$m(\dot{V}_y + V_x\Omega_z) = F_{yr} + F_{yf}\cos(\delta_f) + F_{xf}\sin(\delta_f) \tag{3.2}$$

$$I_z\Omega_z = L_1Fyf\cos(\delta_f) - L_2F_{yr} + L_1F_{xf}\sin(\delta_f) \tag{3.3}$$

where m is the mass of the vehicle, δ_f is the steering angle, V_x and V_y are the local directional velocities, Ω_z is the yaw angle, F_{yf}, F_{yr}, F_{xf} and F_{xr} are the lateral and longitudinal forces at the rear/front wheels. L_1 and L_2 are the distances between front/rear axle and center of gravity of the vehicle and I_z is the inertia. Further, the slip angles α_f and α_r are defined according to Figure 3.3(b), once again using the small angle approximation.

$$\alpha_f = \delta_f - \frac{L_1\Omega_z + V_y}{V_x} \tag{3.4}$$

$$\alpha_r = \frac{L_2\Omega_z - V_y}{V_x} \tag{3.5}$$

Slip angles are mainly due to the lateral elasticity of the tire and arises when a side force is applied on a pneumatic tire. This force develops a lateral force on the contact area, which make the wheel move along this slip angle with the wheel plane. The slip angles are here used to calculate the lateral forces acting on the front and rear tires, which are expressed by

$$F_{yf} = 2C_{\alpha f}\alpha_f \tag{3.6}$$

$$F_{yr} = 2C_{\alpha r}\alpha_r \tag{3.7}$$

where $C_{\alpha f}$ and $C_{\alpha r}$ are the cornering stiffness of the front and rear tire. By combining (3.2), (3.6) and (3.7), the equations for lateral and yaw motions are given by

$$m\dot{V}_y + \left(\frac{2C_{\alpha f} + 2C_{\alpha r}}{V_x}\right)V_y + \left(mV_x + \frac{2L_1C_{\alpha f} - 2L_2C_{\alpha r}}{V_x}\right)\Omega_z = 2C_{\alpha f}\delta_f(t) \tag{3.8}$$

$$I_z\dot{\Omega}_z + \left(\frac{2L_1C_{\alpha f} - 2L_2C_{\alpha r}}{V_x}\right)V_y + \left(\frac{2L_1^2C_{\alpha f} + 2L_2^2C_{\alpha r}}{V_x}\right)\Omega_z = 2L_1C_{\alpha f}\delta_f(t) \tag{3.9}$$

Here, the lateral and yaw motions of the vehicle are calculated with the steering angle as the only input.

3.3 Lateral Active Safety

Today various active safety functions are developed in order to provide autonomous steering, which further provides lateral control of the vehicle. One of these functions is called Lane Keeping Aid (LKA) and its purpose is to warn and assist the driver when the vehicle is heading out of the lane.

In order to detect lane markers and other surroundings, the LKA system uses a module which consists of a camera and radars situated behind the rear-view mirror. The information gathered from this system is processed by a lateral controller. The controller calculates a desired pinion angle based on a heading offset, which is the lateral position error from the desired path. The pinion angle is then sent as an input to the steering controller, which calculates the torque required to steer the vehicle towards the desired direction. This torque request is based on both the input from the lateral controller and the servo torque needed to assist the driver. Figure 3.4 illustrates an example of this LKA manoeuvre.

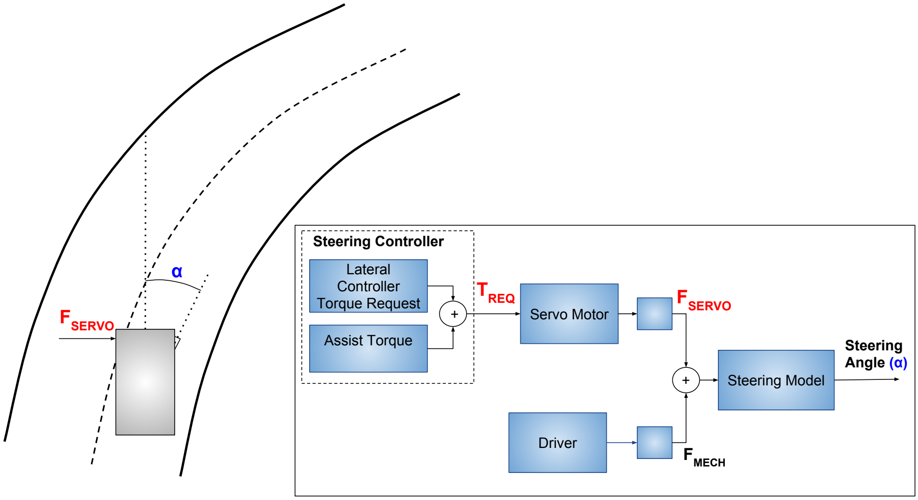


Figure 3.4: *An illustration of a LKA manoeuvre. The pinion angle request from the lateral controller has been translated into the torque needed in order to change the direction according to the requested pinion angle. The steering controller calculates the required servo torque based on the lateral torque request and the torque needed to assist the driver. In the model used in this project, the torque is translated into forces and then used to calculate the rack movement which is further translated into the vehicle's steering angle.*

The LKA helps the vehicle stay within the lanes of the road. This function can be implemented in three different ways: Passive, active or semi-active. Passive means that the function only sends a warning to the driver when the vehicle departure from the lane, Active means that the vehicle will actively help the driver stay in the lane by using the electrical steering system and semi-active means that the driver will get a warning by a vibration in the steering wheel about the lane departure. All of the above described implementations aims to improve the safety and reduce the number of accidents on the roads.

4 Integration of a Steering Model in the HIL-Environment

In order to analyze the steering in the HIL-environment and the effect of the servo gear, an implementation of a more complex steering model was required. The current simplified steering model translates the steering wheel angle and steering wheel speed to rack displacement and rack speed with a calculated gain factor. By integrating the steering model, the complexity in the collection of different subsystems, software and hardware needed to be taken into consideration. This as well as a more thorough explanation about the implemented steering model is further described in this chapter.

4.1 Simulation Environment

The steering model was implemented in the simulation environment as shown in Figure 4.1. The simulation environment consists of a real time simulator connected to the included hardware in the loop. The real time simulator is also connected to two different computers, one which includes the driver and is used for creating scenarios and rendering of the graphics and another computer for controlling the simulation and saving the simulation results. An overview of the HIL setup with the various parts of the test rig used in this project is shown in Figure 4.2.

The different test cases for testing the steering behaviour of the virtual vehicle in the simulation environment are developed in the scenario computer and visualized graphically on the computer screen. As in a real vehicle, a camera and radar is used to detect the road, lanes and objects. This information is used in the active safety domain module (ASDM) by for example a lateral controller to help steer the vehicle and keep it within the lanes.

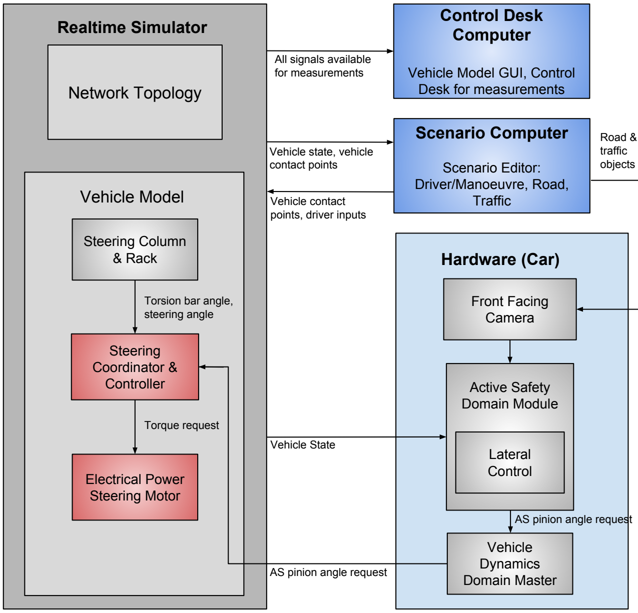

Realtime Simulator
Network Topology
Vehicle Model
Steering Column & Rack
Torsion bar angle, steering angle
Steering Coordinator & Controller
Torque request
Electrical Power Steering Motor
All signals available for measurements
Control Desk Computer
Vehicle Model GUI, Control Desk for measurements
Vehicle state, vehicle contact points
Vehicle contact points, driver inputs
Scenario Computer
Scenario Editor: Driver/Manoeuvre, Road, Traffic
Road & traffic objects
Computer Screen
Hardware (Car)
Front Facing Camera
Active Safety Domain Module
Lateral Control
Vehicle State
AS pinion angle request
Vehicle Dynamics Domain Master
AS pinion angle request

__Figure 4.1:__ A schematic picture of the simulation environment with complete steering implemented as models. The real-time simulator is connected to both hardware from the vehicle and two different computers. One which creates scenarios and one which controls the simulation and saves the results. The hardware used in this setup is a camera and an active safety domain module. The camera is used to detect the road and objects in the visualized scenario while the ASDM is used to request a pinion angle according to the position of the vehicle in the lane.

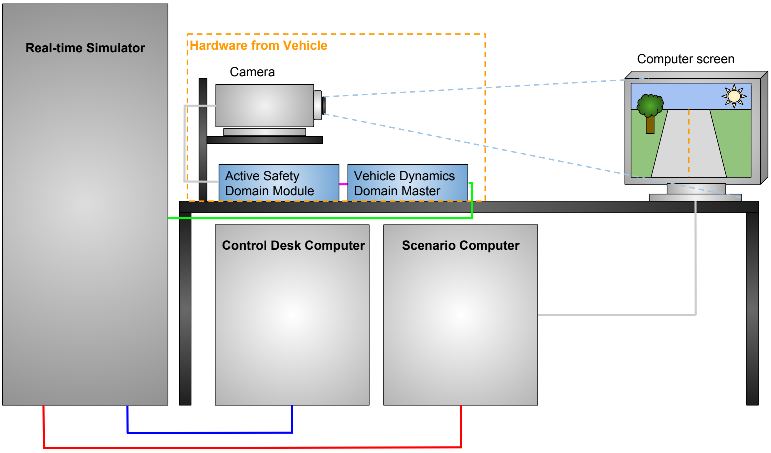


__Figure 4.2:__ An overview of the HIL environment and setup.

4.1.1 Communication

The realtime simulator communicates with different computers and included ECUs from the vehicle with different communication protocols. In Figure 4.3 an overview of the communication within the HIL environment is shown.

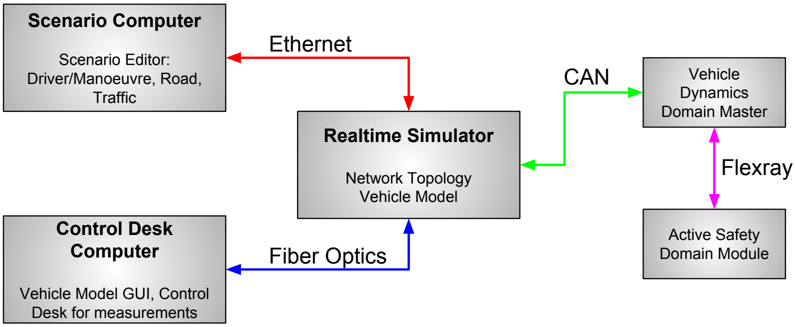

Scenario Computer
Scenario Editor: Driver/Manoeuvre, Road, Traffic
Ethernet
Realtime Simulator
Network Topology
Vehicle Model
CAN
Vehicle Dynamics Domain Master
Flexray
Active Safety Domain Module
Control Desk Computer
Vehicle Model GUI, Control Desk for measurements
Fiber Optics

Figure 4.3: *An overview of the communication within the HIL-environment.*

It is important to note that the communication between the involved computers and ECU will introduce some latency in the simulation. In this project, one of the most important signal to consider is the steering angle target from the driver on the scenario computer to the vehicle model on the real-time simulator. The signals sent from the scenario computer to the real-time computer are sampled with a sampling rate of 100 Hz, whilst the vehicle model itself runs at a frequency of 1 kHz.

Simplified Steering System

The previously used setup in the HIL-environment includes a simplified model for the steering. The steering model calculates the steering rack position and speed from the steering wheel angle and angular speed by using simple gain factors and a filter, see Figure 4.4.

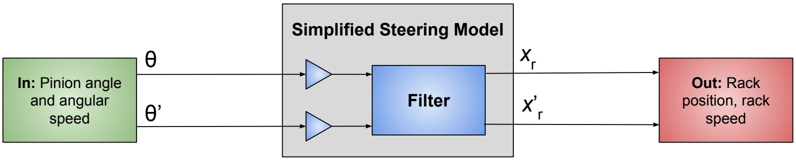

Simplified Steering Model
In: Pinion angle and angular speed
θ
θ'
Filter
x_r
x'_r
Out: Rack position, rack speed

Figure 4.4: *A schematic picture of the simplified steering system.*

As shown in Figure 4.4, the simplified model for the steering does not enable steering support from the servo motor. This means no active safety functions that includes any assisting lateral movement of the vehicle can be tested using this setup. This is the reason this model needs to be replaced.

4.2 Steering Model

In order to enable testing of lateral active safety functions, a more advance model of the steering was implemented. An overview of the model for the steering is shown in Figure 4.5.

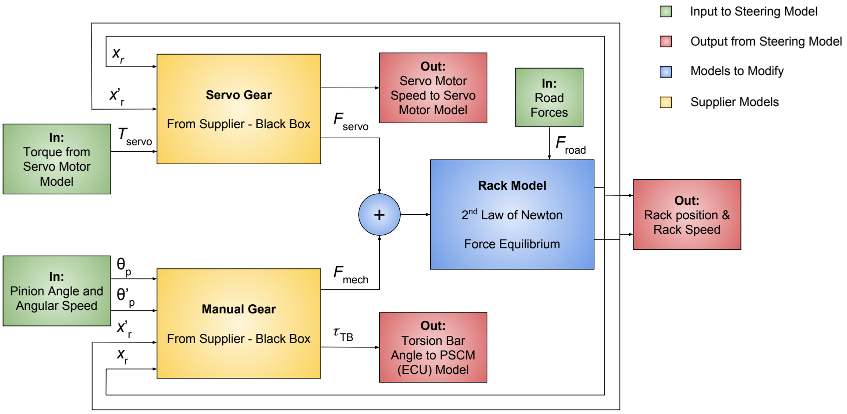

Input to Steering Model
Output from Steering Model
Models to Modify
Supplier Models
x_r
x'_r
Servo Gear
From Supplier - Black Box
Out: Servo Motor Speed to Servo Motor Model
In: Road Forces
In: Torque from Servo Motor Model
T_servo
F_servo
F_road
Rack Model
2nd Law of Newton
Force Equilibrium
+
Out: Rack position & Rack Speed
In: Pinion Angle and Angular Speed
θ_p
θ'_p
x'_r
x_r
F_mech
Manual Gear
From Supplier - Black Box
τ_TB
Out: Torsion Bar Angle to PSCM (ECU) Model

Figure 4.5: *Overview of the MIL-steering model.*

Further, the model for the steering controller, steering coordinator and the servo motor, is extended in order to enable integration of the new steering model. This steering model contains several black-boxes which are created by a supplier, thus only the inputs and outputs are known.

4.2.1 Manual Gear

The block denoted *Manual Gear* corresponds to the estimation of the force from the driver and the torsion bar angle at the steering column, see Figure 3.1. This block takes the pinion angle, pinion angular speed, rack position and rack movement speed as inputs and calculates the force at the rack and the torsion bar angle. The torsion bar angle is the angle on the torsion bar between the pinion and the rack. The implementation of the block will not be further explained as it is supplied by a third party company and not developed during this project.

4.2.2 Servo Gear

The *Servo Gear* block transforms the torque from the servo motor into a force acting on the steering rack. The inputs to this block is the rack position, rack movement speed and the torque from the servo motor. This subsystem was supplied from an third party company an not developed during the project. The block also calculates the rotational speed of the servo motor, which is used by the servo motor model.

4.2.3 Steering Rack

The model for the steering rack was implemented based on the second law of Newton, according to (3.1). An overview of the model is shown in Figure 4.6.

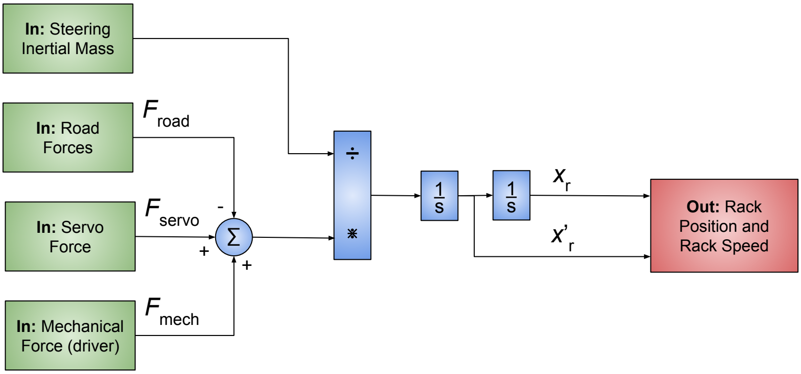


Figure 4.6: A schematic picture of the steering rack model.

4.2.4 Power Steering Control Module

The Power Steering Control Module (PSCM) includes the ECU and the model for the servo motor, an overview of the PSCM can be seen in Figure 4.7. The inputs to the PSCM from the steering are the torsion bar angle from the manual gear and the servo motor speed from the servo gear.

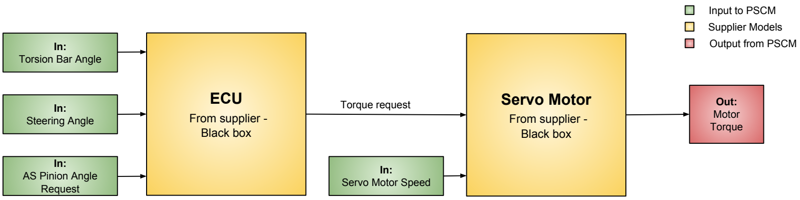


Figure 4.7: Overview of the Power Steering Control Module.

The steering controller is included within the block called ECU. This block calculates the requested torque to the servo motor based on the driver inputs as well as inputs from any lateral active safety function. The torque request includes the servo torque needed to assist the driver and the so called overlay torque based on the pinion angle request from ASDM.

4.2.5 Servo Motor Model

The model for the Servo Motor, see block *Servo Motor* in Figure 4.7, takes the requested torque and calculates the output torque based on the rotational speed of the motor. This block is also supplied from a third party company and will not be explained further.

5

Evaluation of Steering Behaviour

In this chapter, the implemented steering system is evaluated. At first the steering model is evaluated in an open loop system with sinusoidal inputs and then the complete MIL steering system is evaluated in real-time with and without the driver on the scenario computer. By open loop in this case means there is no driver in the loop with feedback from the vehicle states.

5.1 Steering Model in open-loop

The steering model is first evaluated by itself without including the rest of the vehicle model. Further, no real time simulator is used but instead the simulation was performed on a Windows computer with a fixed step size of 1 ms for the solver. When including the rack model in the evaluation, the road force is estimated using the position of the rack and a gain factor.

5.1.1 Manual Gear & Rack without feedback

For validation of the steering model, a number of test cases are defined. The first test case evaluates the manual gear and the signals for the pinion angle and angular speed is fed with sinusoidal input signals. Rack position and rack movement speed are also fed with sinusoidal signals, where the rack position is set to be in phase with the pinion angle and the rack movement speed in phase with the angular speed of the pinion. The reason to feed the manual gear with known signals for the rack state is to isolate it from the affects from the rack model. The setup for test case 1 is shown in Figure 5.1. Here, the servo force is disconnected and does not affect the rack movement.

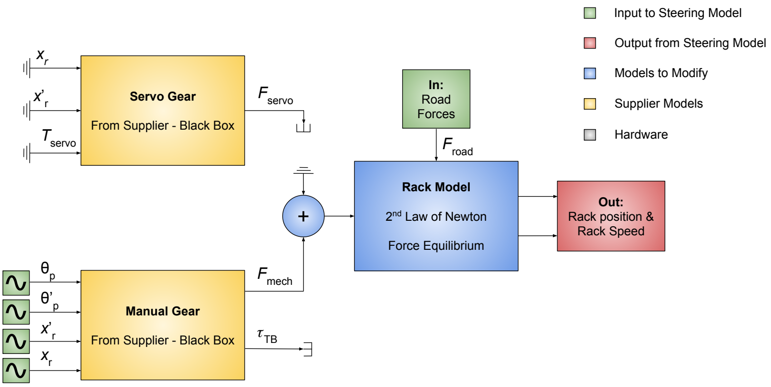


Figure 5.1: *Test setup for evaluation of the manual gear & rack without feedback.*

The results from the simulation of the manual gear fed with continuous sinusoidal inputs for the pinion angle and the rack states can be seen in Figure 5.2

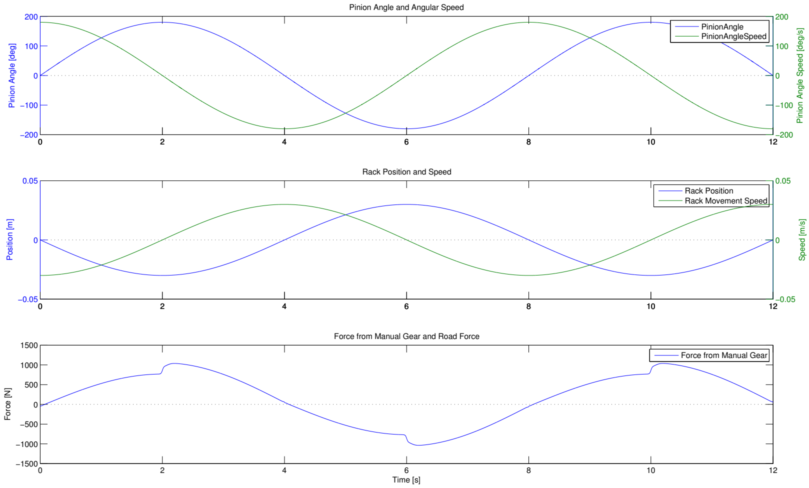


Figure 5.2: *Simulation results from simulation of the manual gear with continuous sinusoidal signals for pinion and rack.*

In overall, the manual gear behaves as expected, the force from the manual gear follows the pinion angle speed in order to move the rack in the desired direction. Note the force from the manual gear shows some strange behavior whenever the pinion angle speed or the rack movement speed are zero. When pinion angle speed and the rack movement speed are zero, the absolute amplitude of the force increases rapidly. The sinusoidal input is now fed as a sampled signal at 50 Hz, to correspond to a typical signal received from the scenario computer. The rack states are still kept as continuous sinusoidal signals as this would be the case with the rack model in the loop. The results are shown in Figure 5.3.

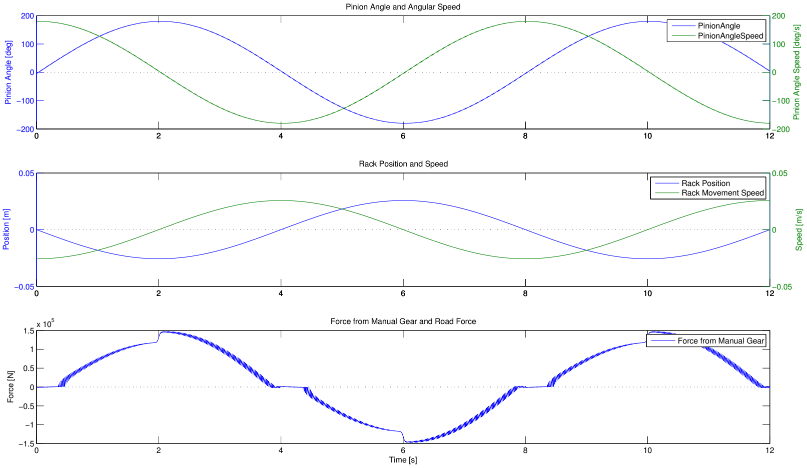


Figure 5.3: *Simulation results from simulation of the manual gear with sampled sinusoidal signals for pinion and continuous sinusoidal signals for the rack.*

Feeding the manual gear with sampled signals results in oscillations on the force. The reason for this is that the sampled signals for the pinion angle and angular speed are constant while the rack changes during the same period. Also, it is not reasonable to have a constant non-zero pinion angle speed at the same time as the pinion angle is constant during more than one simulation time step, or micro step. This contributes to some problems within the manual gear as these signals are compared to the position and the movement speed of the rack. As noted before, the rack position and the pinion angle are compared in order to calculate the force as well as the torsion bar angle from the manual gear.

5.1.2 Manual Gear & Rack

The second test case also evaluates the manual gear, but with the feedback of the rack position and movement speed from the rack model. In this case the pinion angle and angular speed are again fed with sinusoidal signals. In order to include the rack model, an estimation of the road forces (rod force) is done by feeding back the rack position multiplied with a tuning factor. The servo force is disconnected from the rack model. The test setup is shown in Figure 5.4.

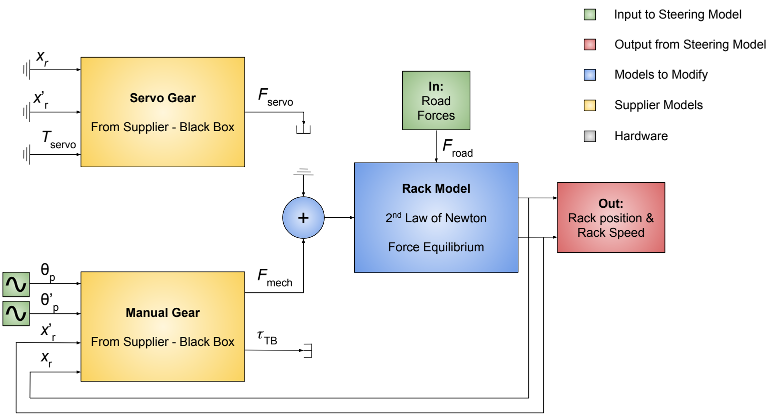


Figure 5.4: *Test setup for evaluation of the manual gear & rack with feedback.*

The pinion angle and the angular speed was fed with continuous sinusoidal signals and the rack position and rack speed was fed back from the rack model. The results from the simulation are shown in Figure 5.5.

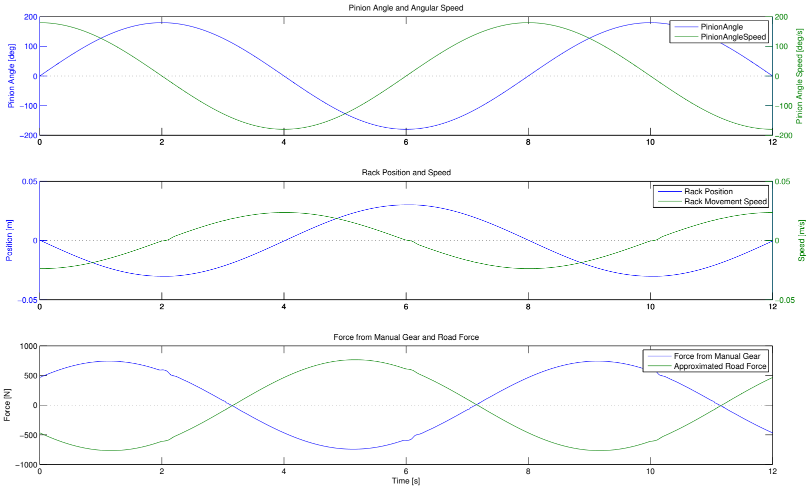


Figure 5.5: *Simulation results from simulation of the manual gear with continuous sinusoidal signals for pinion and the rack states fed back from the rack model.*

Comparing the result from Figure 5.5 with the results in Figure 5.2, shows that feeding back the rack states instead of feeding them as sinusoidal signal eliminates the strange behavior when the pinion angular speed and the rack movement speed are zero. This might be due to the fact that the affects from the road force was not included when feeding the servo gear with sinusoidal signals for the rack states. In another experiment, the sinusoidal signals for the pinion angle and angular speed is fed with sampled signals with a sampling frequency of 50 Hz. The results from the simulation are shown in Figure 5.6.

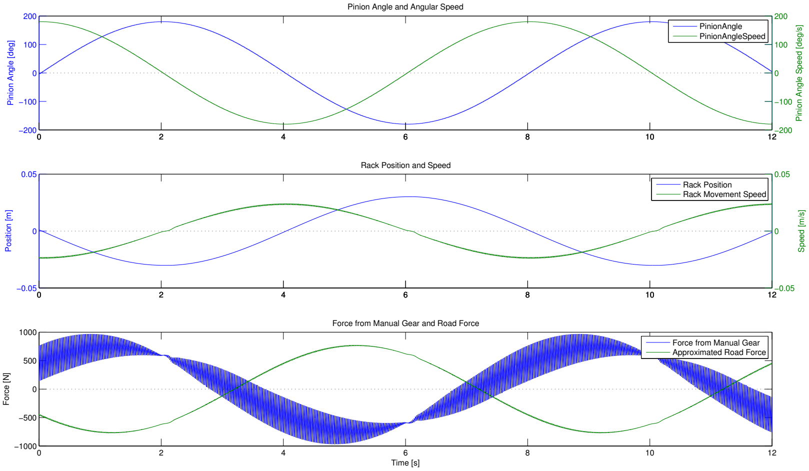


Figure 5.6: *Simulation results from simulation of the manual gear with sampled sinusoidal signals for pinion and rack states fed back from the rack model.*

Feeding the manual gear with sampled signals results in unwanted dynamics in the force from the manual gear. The pinion angle and the angular speed should never be constant non-zero at the same time during more than one micro step.

5.1.3 Servo Gear & Rack without feedback

This test case evaluates the servo gear and the test setup is shown in Figure 5.7. When evaluating the servo gear, the manual gear is disconnected from the rack model. The servo torque as well as the rack position and speed into the servo gear are fed with sinusoidal signals. The rack speed is set to be in phase with the servo torque and the position corresponding to the speed of the rack. This test case does not involve any sampled signals as all the input signals using the complete simulation setup will come from the vehicle model, meaning the input signals will be updated just as fast as the model runs. The results from the simulation of the servo gear are shown in Figure 5.8.

As is shown in Figure 5.8, the force from the servo gear follows the torque into the servo gear as expected.

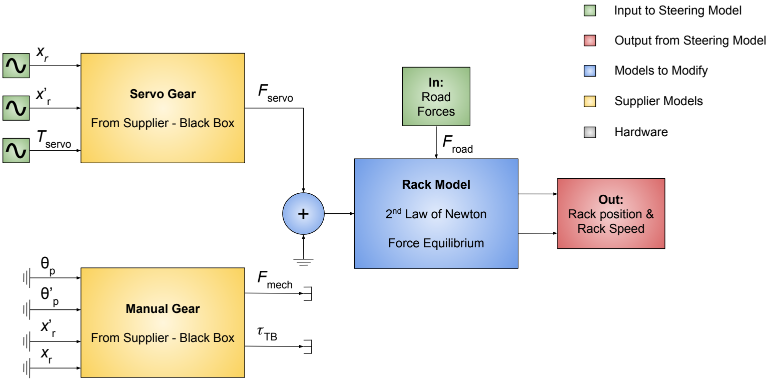


__Figure 5.7:__ Test setup for evaluation of the servo gear without feedback from the rack model.

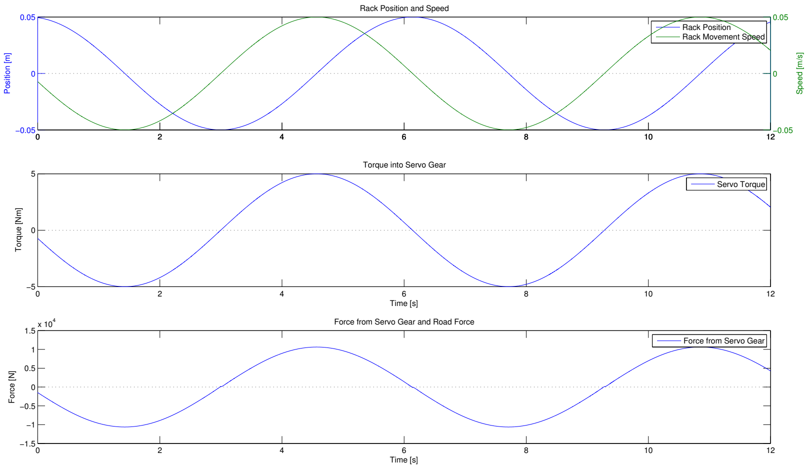


__Figure 5.8:__ Simulation results from simulation of the servo gear with sinusoidal signals for servo motor torque and rack states.

5.1.4 Servo Gear & Rack

The setup with the servo gear and rack is shown in Figure 5.9. The test case evaluates the servo gear with the rack position and movement speed fed back from the rack model. The servo motor torque is once again fed with a continuous sinusoidal signal. The results from the simulation of the servo gear fed back with the rack position and speed from the rack model are shown in Figure 5.10.

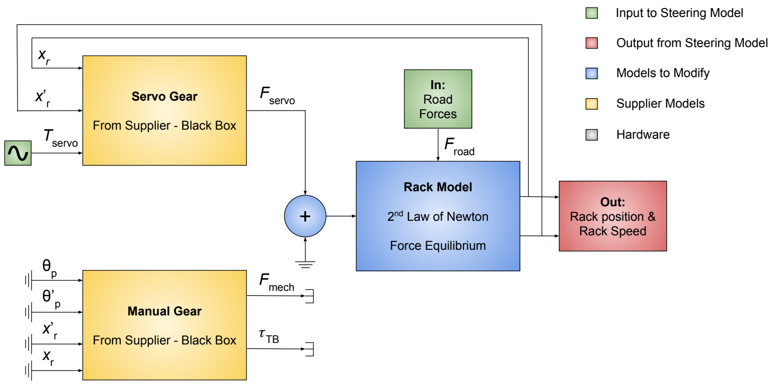


Figure 5.9: *Test setup for evaluation of the servo gear with feedback from the rack model.*

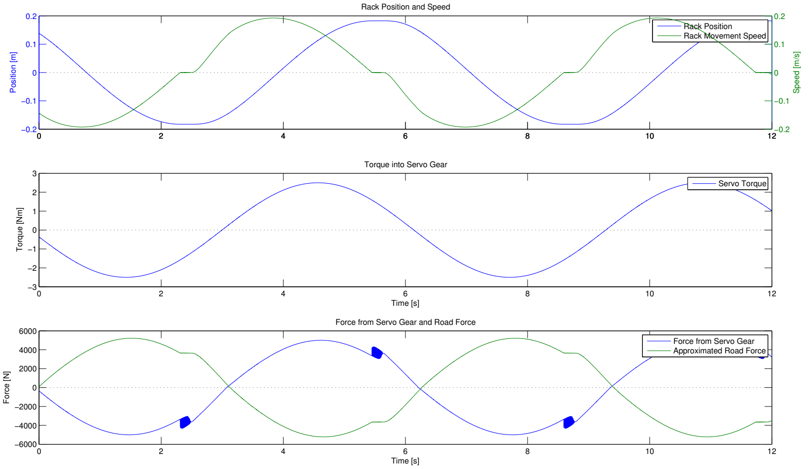


Figure 5.10: *Simulation results from simulation of the servo gear with sinusoidal signal for the servo motor torque and rack states fed back from the rack model.*

As shown in Figure 5.10, the force from the servo gear follows the torque, but there are some oscillations whenever the rack movement speed is zero.

5.2 Steering model in closed-loop

The steering model is further evaluated in closed-loop, real-time simulations with and without the driver from the scenario computer. At first, the steering model is fed with continuous sinusoidal signals for the pinion angle and pinion angular speed and is compared with sampled sinusoidal inputs with a sampling rate of 50 Hz. The steering model is then simulated with real measurement inputs in order to enable validation of the simulation results against an already validated MIL environment. Validation against measured signals from the actual tests is not possible due to confidentiality reasons. At last the steering model is connected to the driver from the scenario computer.

5.2.1 Sinusoidal input signals without driver

This test case evaluates the complete model in the loop steering system in a real-time simulation, but without using the steering inputs from the scenario computer. The pinion angle and angular speed are instead implemented as sinusoidal waves with different sampling frequencies. At first the signals for the pinion angle and pinion angular speed are implemented as sinusoidal waves with an amplitude of 30 and with a frequency of 1 rad/s. The phase offset for the angular speed is set to $\pi/2$, which is shown in Figure 5.11 together with the simulation results. In Figure 5.12, the sampling frequency of the sinusoidal wave is set to be 50 Hz in order to emulate the frequency of the received signal from the scenario computer, see subsection 4.1.1.

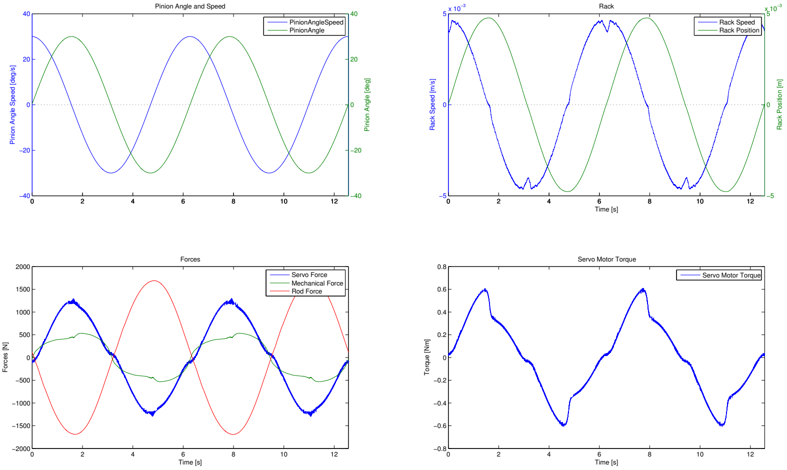


Figure 5.11: *Simulation results from real-time simulation of the complete model in loop steering system with continuous sinusoidal input signals for the pinion angle and angular speed.*

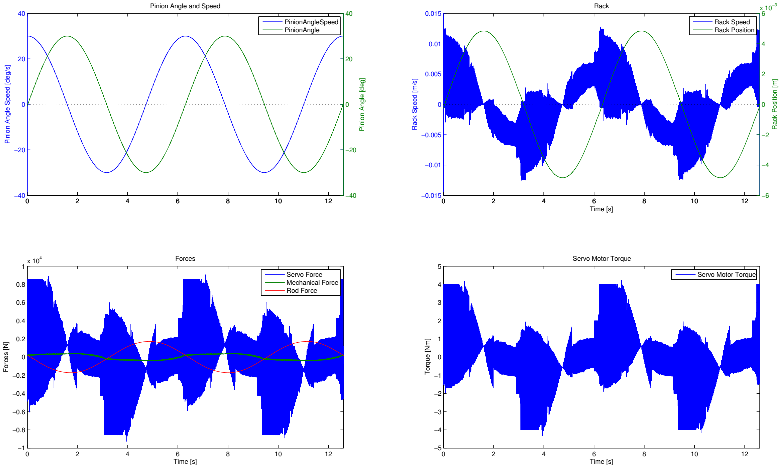


Figure 5.12: *Simulation results from real-time simulation of the complete model in loop steering system with sinusoidal input signals for the pinion angle and angular speed which has a sampling frequency of* 50 Hz.

As is shown in the simulation results, the difference in sampling rate results in some noisy signals and an oscillatory behaviour. The signals which are fed back from rack are updated with a frequency of 1 kHz while the pinion angle and angular speed into the manual gear are updated with a frequency of 50 Hz, which causes the signals to be out of phase. However, the results from Figure 5.11 shows some spikes in the manual force as well as the rack movement speed.

5.2.2 Real measurement data without driver

In order to validate the results of the steering behaviour, real measurements of a sinus with dwell manoeuvre are used to compare and validate the results with a validated MIL-environment. The results of the simulation in the MIL- and HIL-environment is shown in Figure 5.13 and Figure 5.14. Note that the input pinion angle and angular speed signals are sampled at the same rate as the vehicle model runs. This scenario is used to see how the response of the vehicle resembles with the already validated results from the MIL-environment in terms of steering behavior.

By comparing the two results, observations can be made about the forces and the torque from the results in the HIL. The servo force and the torque does not show the constant behaviour in the dwell section. Also note the amplitude of the rod force is smaller in the results from the simulation in the HIL environment compared to MIL-setup. The rod force also seems to have a delay in HIL compared to the simulation in MIL.

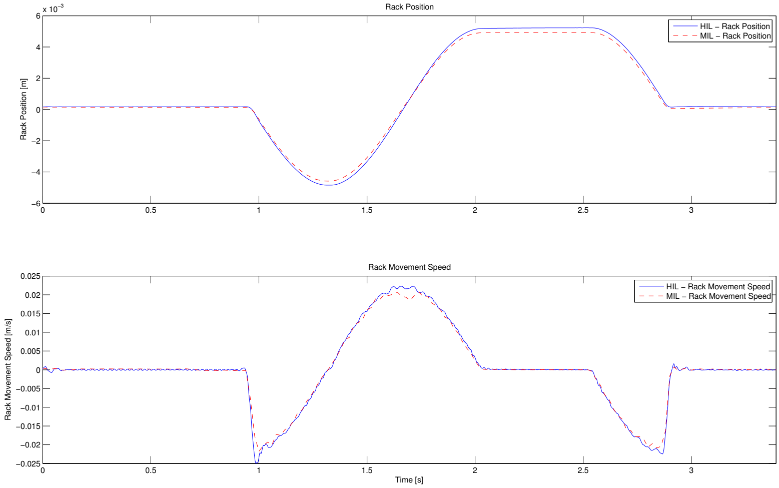


Figure 5.13: *Simulation results for rack position and rack movement speed from real-time simulation in* HIL *together with validation data from* MIL*.*

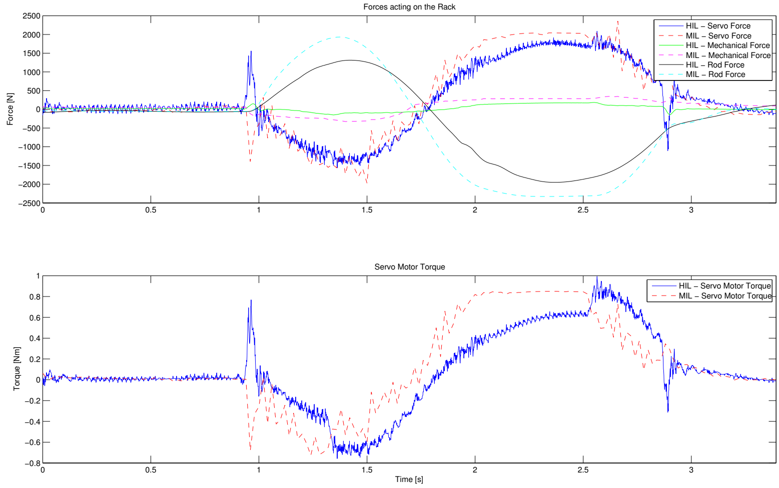


Figure 5.14: *Simulation results for forces acting on the rack and servo motor torque from real-time simulation in* HIL *together with validation data from* MIL*.*

5.2.3 Simulation with driver

The setup for the simulation with driver is shown in Figure 5.15. This test case evaluates the complete model in the loop steering system. Steering inputs comes from the driver on the scenario computer and the torque to the servo gear from the PSCM.

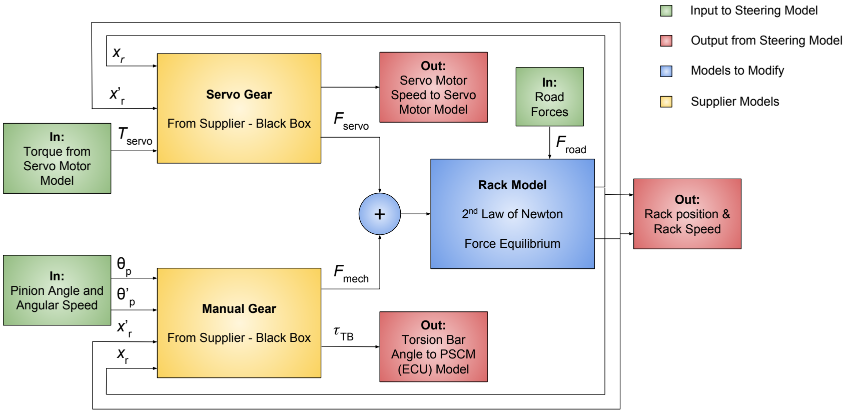


Figure 5.15: *Test setup for evaluation of the complete steering model.*

In Figure 5.16, the driving scenario used in the simulation with the scenario computer and driver is shown. Further, Figure 5.17 shows the simulation results. In this scenario the vehicle starts by driving straight, then it changes lane and continues to drive straight in the new lane. It is used since it effectively emulates the steering behaviour.

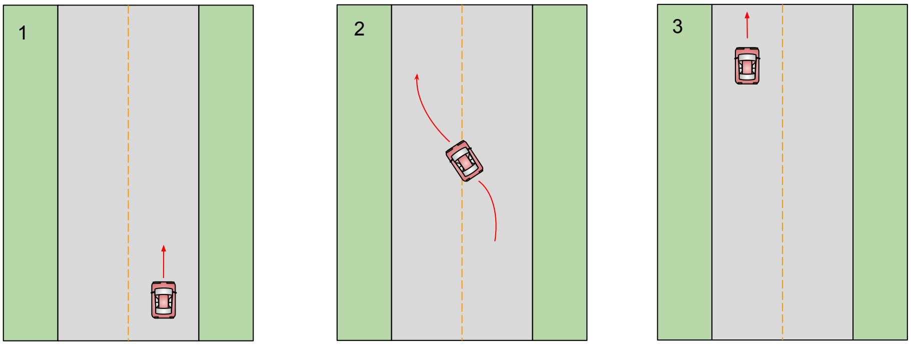


Figure 5.16: *The driving scenario used in the closed-loop simulation with the scenario driver.*

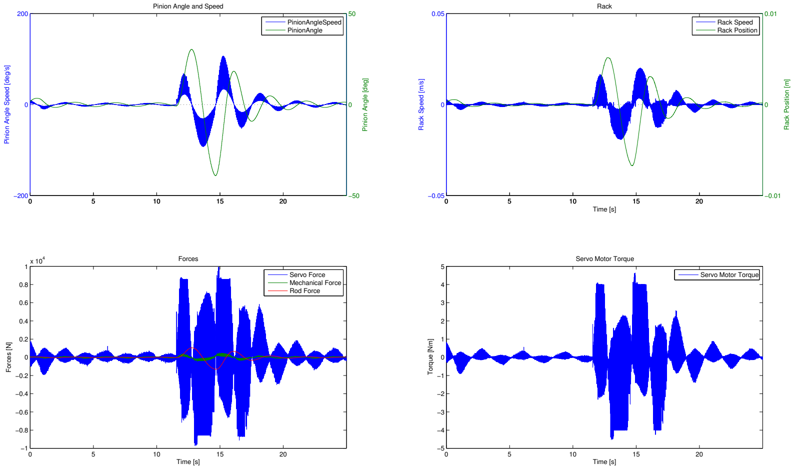


Figure 5.17: *The results from the closed-loop simulation with the scenario driver.*

Due to the use of the driver from the scenario computer in this simulation, the input pinion angle signal is updated with a frequency of approximately 100 Hz. The signal is held constant in a variation from 10 ms up to 30 ms due to the fact that the scenario computer is not ready to send new data at the synchronization times. Moreover, as described in subsection 4.1.1 there is a latency of approximately 150 ms in the feedback to the driver. This results in oscillatory and unstable behaviour. Note that the pinion angle speed is calculated as the derivative of the pinion angle.

5.3 Observations

The most important observation from the results from the individual simulations of the manual gear and the servo gear shows that the manual gear should not be fed with sampled signals. As this will occur when involving the complete simulation environment, this needed to be improved in order to achieve reasonable simulation results. Another observation is the fact that the manual gear and the servo gear are both very stiff models, which means that the models have high and fast dynamics due to the lack of damping. This might cause problems when involving the rest of the simulation environment.

The validation using the measurement data from the validated MIL-environment shows that the rod force in the HIL-environment needs to be investigated.

6

Improvements

As seen in chapter 5, in order to enable testing of the steering behaviour and lateral active safety functions, the HIL- environment needed some improvements in order to achieve better results.

6.1 Latency in Simulation

As mentioned in subsection 5.2.3, the pinion angular speed was very noisy. The reason for this is that it was calculated as the continuous derivative of the pinion angle signal. In order to eliminate the noise in the signal, the discrete derivative was calculated. However, this creates a drawback of increased latency in the simulation.

To decrease the latency added by calculating the discrete derivative inside the vehicle model, the calculation of the derivative was instead moved to the scenario computer. The latency of approximately 150 *ms* within the simulation environment creates issues and instability. To solve this, two different approaches was investigated and are further described in this section.

6.1.1 Rod Force

As noted in the evaluation of the steering model, the force from the road into the rack model (rod force) differed from the force seen in the validation data. The amplitude of the rod force seemed to be smaller compared to the rack movement. Moreover, there seems to be a small delay in the rod force from the vehicle model. One of the main reasons for this is because of the complexity of the simulation environment compared to the environment used for the validation data. The subsystems denoted manual gear and servo gear are both tuned according to some expected road force. Also, the rack model itself is a physical model and the forces should change continuously without any delay.

In order to evaluate the steering model, this rod force needed some improvement. The way this was done was by simply use the rack position and multiply it with a tuning factor to achieve similar characteristics for the rod force as in the validation data. This signal, rack position multiplied with a tuning factor, was then fed back to the rack model directly. In this way, the road force was updated according to the position of the rack. As the road force was not considered as part of the steering system, it was simplified in order to perform a fair validation of the steering system.

6.1.2 Prediction Trajectory

The computational time and the latency in communication between the driver on the scenario computer and the vehicle model create steering problems. In the scenario computer, the driver steers and follows a path by sending the wheel angle to the steering model on the real time simulator, which calculates and feeds back the vehicle states to the driver. Due to the latency, the driver does not receive the states in time which causes an oscillatory behaviour of the vehicle. The vehicle states includes the position of the vehicle in global x and y coordinates as well as the heading Ω_z angle, these are shown in Figure 6.1.

To solve the oscillatory behaviour a prediction method was implemented where the predicted position in regards to the latency was calculated. Since the speed of the vehicle is known in the various directions, it was multiplied with a look ahead time in order to estimate the predicted movement of the vehicle in the global coordinate system. This was then added to the current position of the vehicle in order to calculate the predicted position to reduce the effect of the latency.

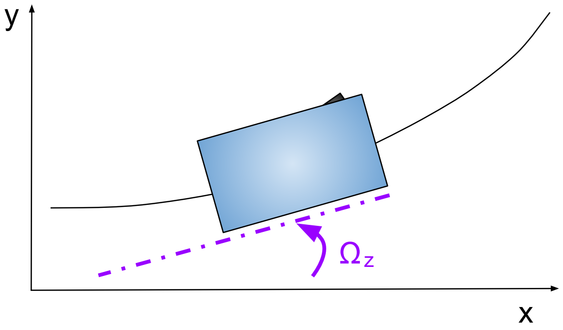


Figure 6.1: The figure shows the global coordinate system of x and y and the heading Ω_z.

6.2 Multi-Rate Simulation

As discussed earlier, the simulation setup includes subsystems working on different frequencies. The data from the scenario computer should be sent to the vehicle model every 10 ms, but as the scenario computer is not always ready during the synchronization points, the signals are kept constant during 20-40 ms. As noted in the evaluation of the steering model in chapter 5, sending sampled signals into the model caused some unwanted behaviour of the model. Therefore, the signals received from the scenario computer needed to be processed before entering the steering model. The signals from the scenario computer are sampled at approximately 100 Hz and the steering model runs at 1 kHz. See Figure 6.2 for a typical signal received from the scenario computer into the vehicle model, with the use of zero-order hold extrapolation.

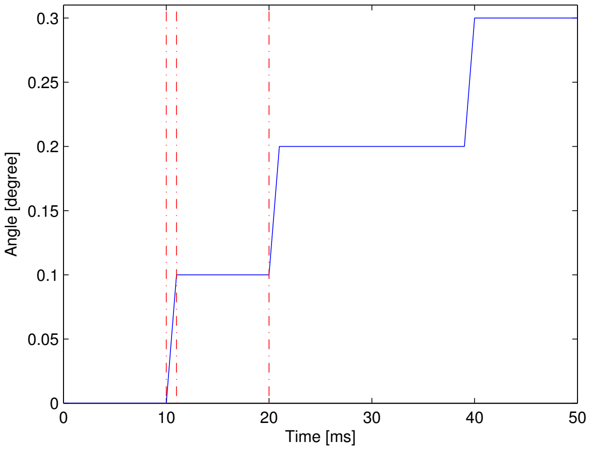


Figure 6.2: *Staircase signal received from the scenario computer into vehicle model.*

To extrapolate data using a higher order polynomial during the macro steps, as discussed earlier, did not give satisfying results as this implementation would require the sampling frequency to be known. Another option would be to implement a model-based extrapolation scheme [4]. The problem with implementing this is the fact that the sending and receiving dead-times between the involved subsystems is not known and also time-varying. Therefore the signals was processed with two different filters instead in order to get rid of the staircase signals.

6.2.1 Discrete Averaging Filter

The obvious solution for smoothing the staircase signals would be to implement a discrete filter. The filter was implemented as the mean value for the sum of N discrete time steps as

$$y(n) = \frac{1}{N+1}\sum_{i=0}^{N} x(n-i) \tag{6.1}$$

However, implementing a discrete filter adds a delay dependent of the number of time steps N chosen. This means adding a discrete filter would increase the latency between the driver on the scenario computer and the feedback from the vehicle model. Two typical signals for the pinion angle and the angular speed received from the scenario computer with and without the discrete filter can be seen in Figure 6.3 and Figure 6.4. Note that the signals are processed after the simulation.

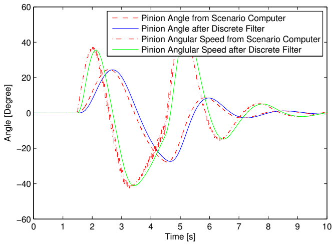


Figure 6.3: *Pinion angle and angular speed received from the scenario computer before and after the discrete filter.*

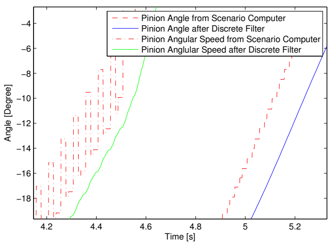


Figure 6.4: *Zoomed in - Pinion angle and angular speed received from the scenario computer before and after the discrete filter.*

6.2.2 Low Pass and Lead Compensation Filter

Another option was to implement a continuous first order low pass filter. The cut-off frequency for the filter was chosen to eliminate the steps in the signal and was chosen experimentally to 5 Hz. Adding a low pass filter with such a low cut-off frequency would also add some unwanted phase lag. In order to solve this, a phase advance filter, or a lead compensation was added after the low pass filter [10], see Figure 6.5. The first order low pass filter was implemented as

$$G_{lowpass} = \frac{1}{\frac{s}{\omega_c}+1} \tag{6.2}$$

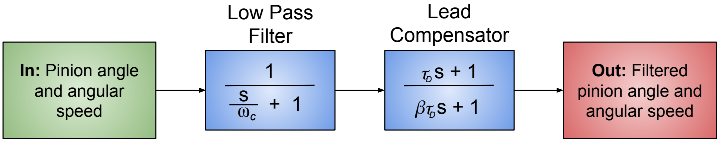

Low Pass Filter
Lead Compensator
In: Pinion angle and angular speed
$\frac{1}{\frac{s}{\omega_c} + 1}$
$\frac{\tau_D s + 1}{\beta \tau_D s + 1}$
Out: Filtered pinion angle and angular speed

Figure 6.5: *An illustration of the implemented low pass filter and the lead compensator.*

Where the cutoff frequency ω_c was set to be $2\pi \cdot 5\ rad/s$. A bode plot for the selected filter can be seen in Figure 6.6.

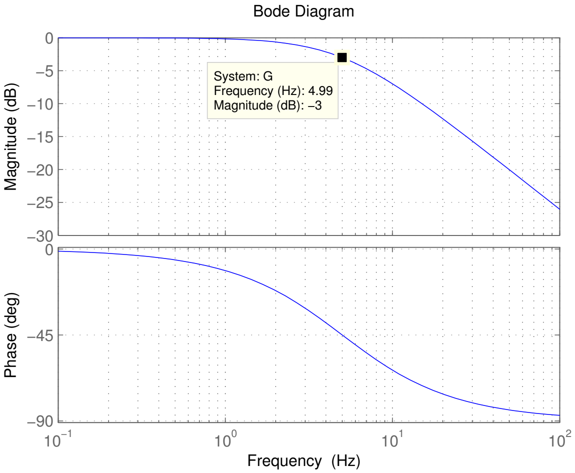


Figure 6.6: *Bode plot for the implemented first order low pass filter.*

As can be seen in Figure 6.6, the low pass filter adds an unwanted phase lag. This phase lag was manipulated by using lead compensation as

$$F_{lead} = \frac{\tau_D s + 1}{\beta \tau_D s + 1} \tag{6.3}$$

Where β is a tuning parameter to adjust the phase advancement of the lead compensator. In order to achieve a phase advancement, the absolute magnitude of β needs to be smaller than one. Decreasing the value of β moves the pole of the lead compensator away from the imaginary axis and hence increases the phase advancement. The drawback with a too small β is that the lead compensator will increase the amplitude of the high frequent signals. As the value of the parameter β could not be calculated analytically, it was tuned in order to achieve satisfying results for the phase lag of the product of the low pass filter and the lead compensator. Once the value for β was chosen, the value for τ_D was calculated so that the maximum phase advance would occur in the desired cut-off frequency according to

$$\tau_D = \frac{1}{\omega_c \sqrt{\beta}} \tag{6.4}$$

Applying the lead compensator after the low pass filter gives the following expression for the transfer function

$$G_{lowpass} F_{lead} = \frac{1}{\frac{s}{\omega_c} + 1} \frac{\tau_D s + 1}{\beta \tau_D s + 1} \tag{6.5}$$

In Figure 6.7 the resulting bode plot for the filter followed by the lead compensator is shown.

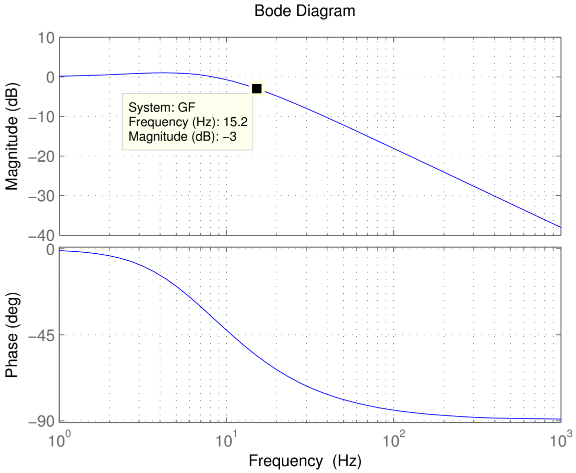


Figure 6.7: *Bode plot for the implemented first order low pass filter followed by a lead compensator.*

Two typical signals for the pinion angle and angular speed received from the scenario computer into the vehicle model together with results after the low pass filter with and without the lead compensator can be seen in Figure 6.8 and Figure 6.9.

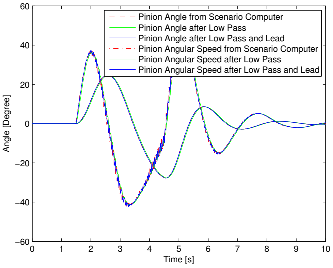


Figure 6.8: *Pinion angle and angular speed received from the scenario computer before and after the low pass filter with and without the lead compensation.*

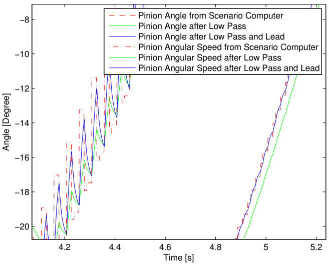


Figure 6.9: *Zoomed in - Pinion angle and angular speed received from the scenario computer before and after the low pass filter with and without the lead compensation.*

7 Result and Discussion

In this chapter the final results with the improvements presented in chapter 6 implemented in the HIL-environment are presented. The scenarios will be the same as the ones presented in chapter 5. During all the simulations presented in the results, the real time simulator was used but with and without the driver on the scenario computer.

7.1 Filter

As discussed earlier, the signals from the scenario computer into the vehicle model needed processing before entering the steering model. To verify the results from the filters alone, sampled sinusoidal signals for the pinion angle and angular speed was used instead of using the driver on the scenario computer. The signals was sampled at 50 Hz and the results from using the discrete filter and the low pass with lead compensation can be seen in Figure 7.1 and Figure 7.2.

From the results using the discrete filter and the low pass with lead compensation, there is obvious the discrete filter is the better choice to achieve results with less noisy signals.

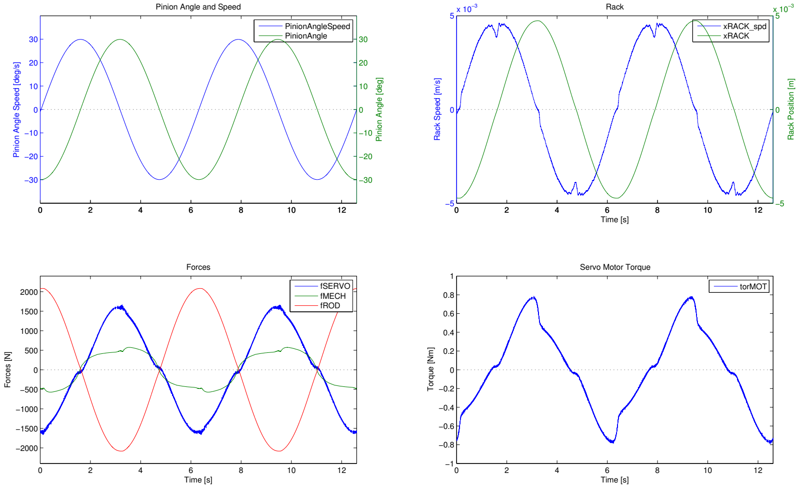


Figure 7.1: *The results from the simulation using sampled sinusoidal inputs for pinion angle and angular speed with discrete filter.*

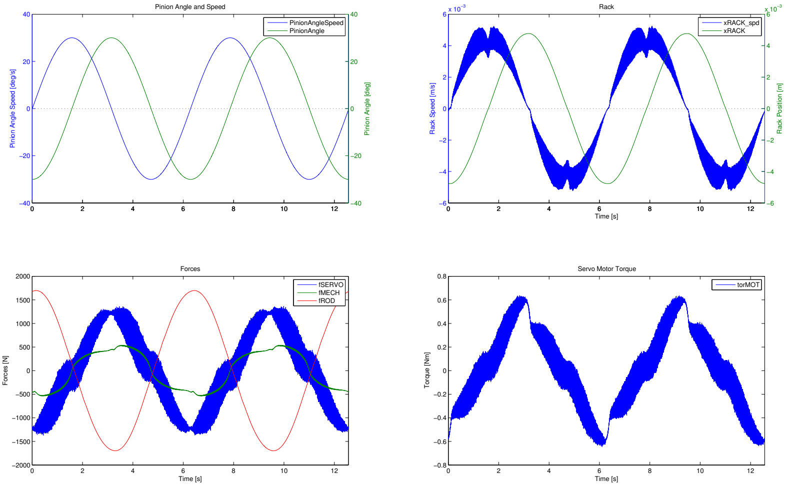


Figure 7.2: *The results from the simulation using sampled sinusoidal inputs for the pinion angle and angular speed with low pass filter and lead compensation.*

7.2 Rod Force

As previously mentioned in subsection 5.2.2, simulations was compared with results from simulations in a validated MIL-environment. To see the difference in using the updated rod force and in order to validate the steering model, the HIL-environment was in the same way as in subsection 5.2.2 simulated without the scenario driver and with the validated data as input signals for the pinion angle and angular speed. The results from this simulation can be seen in Figure 7.3 and Figure 7.4. Note that each signal is compared with the same signal simulated in the validated MIL-environment. Also note that no filters was used when validating the changes for the rod force as the inputs from the validation data was already sampled at the same rate as the vehicle model runs.

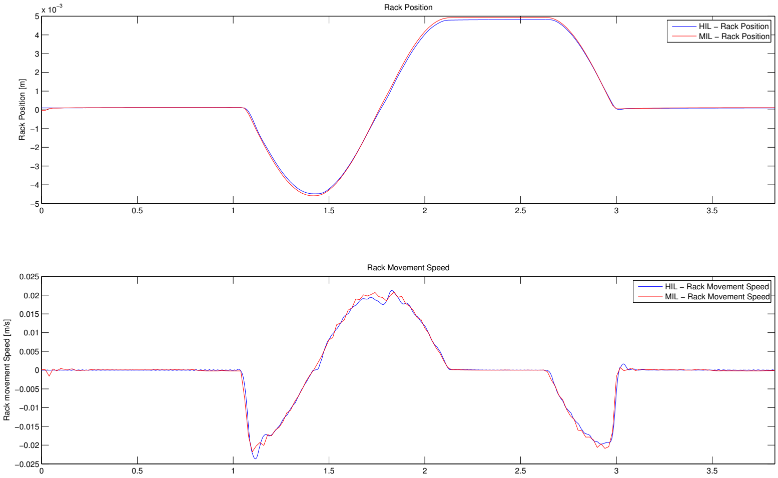


Figure 7.3: *The simulation results from improved steering model in comparison to the already validated measurements. The result shows the signals for the rack position and the rack movement speed.*

As can be seen, the results from the HIL-environment has similar behaviour as the results from the validated MIL-environment. Further, the signals are smooth and even less oscillatory than the validation data. Note the amplitude of the force from the manual gear is slightly bigger in the HIL than in MIL. As this force is bigger, the force required from the servo gets smaller. Further, the difference in the rod force between the two simulation environments is due to the fact that the rack position with a tuning factor is used to achieve the same characteristics in the HIL-environment as in the validated MIL-environment. Thus, this force does not consider slip angles at the tires. As the rod force is different in HIL, the resulting forces from the manual gear and servo gear will not be the same as in MIL. In terms of steering of the vehicle, the most important signals to study are the rack position and rack movement speed, seen in Figure 7.3. The rack position and movement speed from the HIL follows the validation data from MIL almost perfectly.

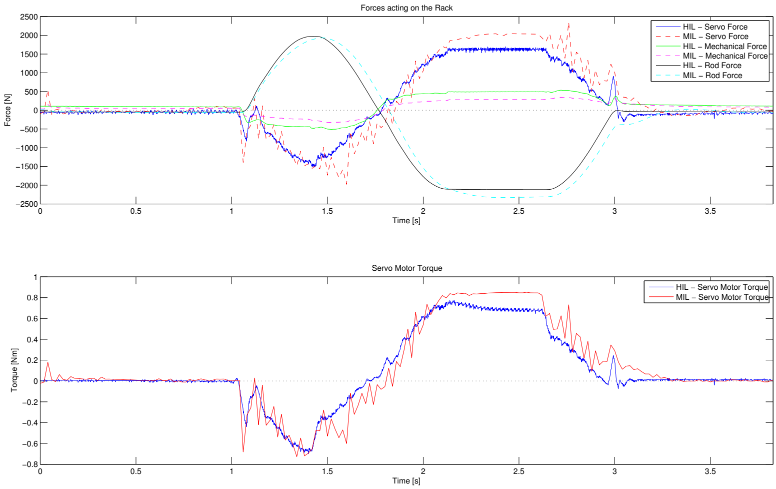


Figure 7.4: *The simulation results from improved steering model in comparison to the already validated simulation results. The figure shows the signals for the servo force, mechanical force and rod force as well as the servo motor torque.*

7.3 Complete HIL Simulation with Driver

When simulating the complete HIL-environment, all improvements presented in chapter 6 was used, with varying look ahead time for the position prediction. At first the discrete filter was used and then the low pass with lead compensation. The results from simulation with the implementation of the discrete filter is shown in Figure 7.5. Note that no position prediction was used in this case.

As can be seen, the results are oscillatory. This is due to the latency between the driver and the vehicle model. In order to reduce these oscillations, the position prediction presented in subsection 6.1.2 was included. At first a prediction of 50 ms was simulated and can be seen in Figure 7.6.

The results from simulations with the prediction shows the prediction indeed decreases the oscillations. However, as the discrete filter adds further delay inside the vehicle model, the prediction time was extended to 300 ms. The results can be seen in Figure 7.7.

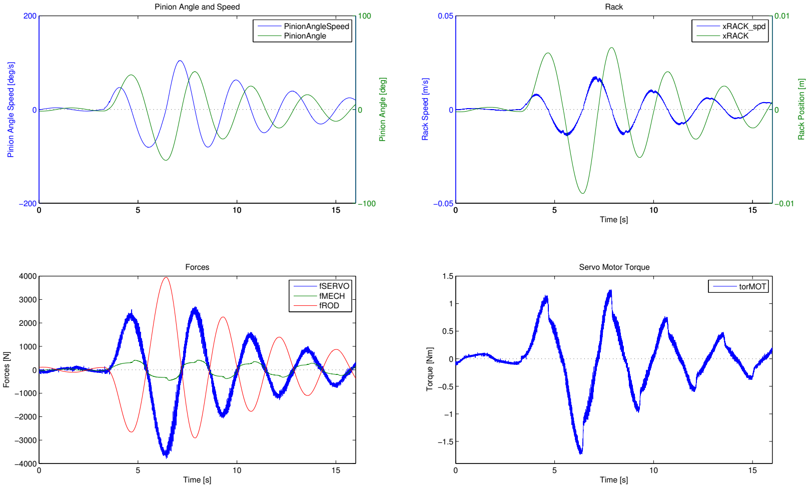


Figure 7.5: *Simulation results from simulation using the discrete filter for the steering signals from the driver on the scenario computer and no position prediction.*

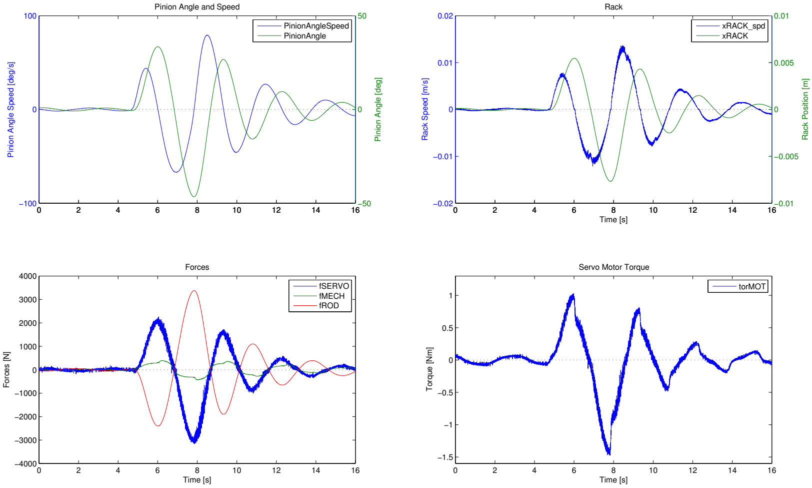


Figure 7.6: *Simulation results from simulation using the discrete filter for the steering signals from the driver on the scenario computer and position prediction with 50 ms look ahead time.*

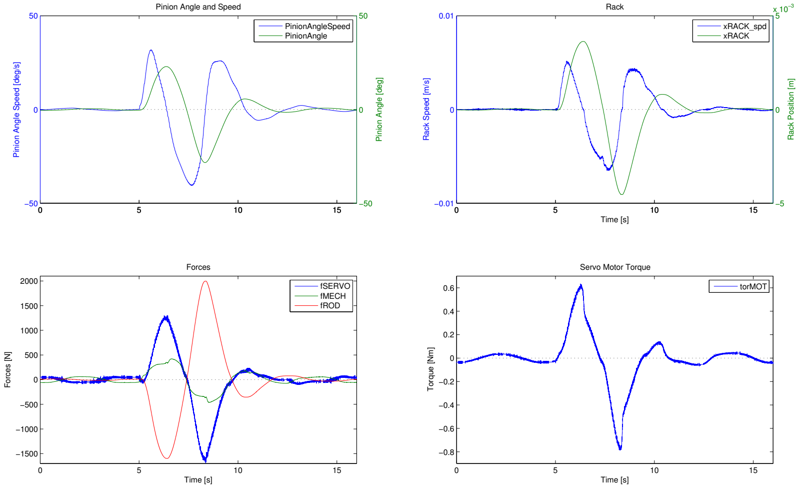


Figure 7.7: *Simulation results from simulation using the discrete filter for the steering signals from the driver on the scenario computer and position prediction with 300 ms lock ahead time.*

Using 300 ms prediction time reduces the oscillations dramatically. The problem with the oscillations can then be isolated to the fact it is due to the latency. The suggested low pass filter with lead compensation was then tested as this would not add as much delay as the discrete filter. In Figure 7.8, the result from the simulation with the implemented low-pass filter and lead compensation, without any position prediction, is shown.

Even without any filter, the latency between the driver and the vehicle model is big enough to give oscillations in the signals. The low pass filter with lead still adds some delay, therefore, the position prediction was implemented with the low pass filter and lead compensation as well. The prediction time was set to 50 ms and the results can be seen in Figure 7.9.

The prediction time was then increased to 100 ms for the low pass with lead compensation and the results can be seen in Figure 7.10.

As can be seen in the results from using the discrete and the low pass filter with lead compensation, the discrete gives the best results in terms of less noisy signals. However the discrete filter introduces a delay which increases the oscillations of the signals as the driver on the scenario computer receives the updated states of the vehicle far to late. Using the position prediction for the discrete filter gives similar results as with the low pass and lead compensation considering the oscillations. This means it could be possible to use the discrete filter and reduce the effects of the latency using the position prediction.

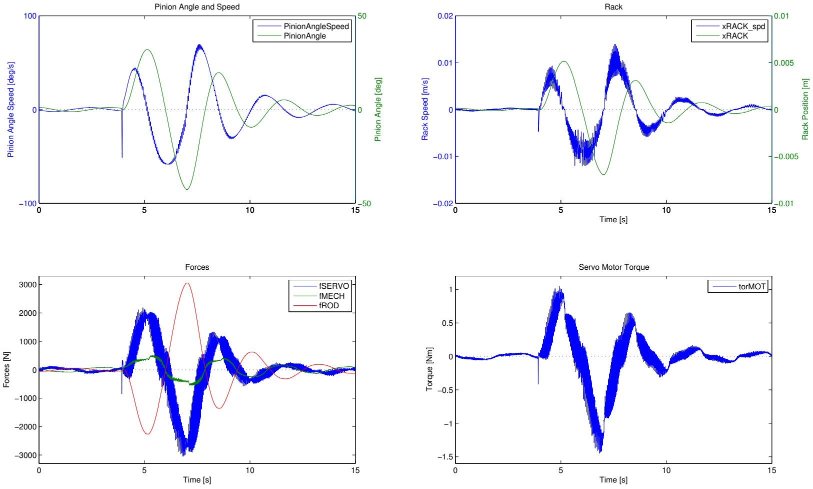


Figure 7.8: *Simulation results from simulation using the low pass filter and lead compensation for the steering signals from the driver on the scenario computer. No position predicton was used.*

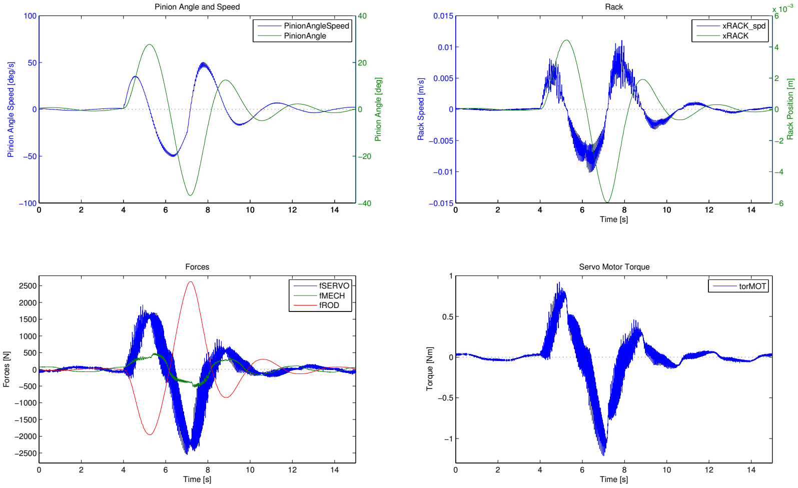


Figure 7.9: *Simulation results from simulation using the low pass filter and lead compensator for the steering signals from the driver on the scenario computer. Position prediction time 50 ms.*

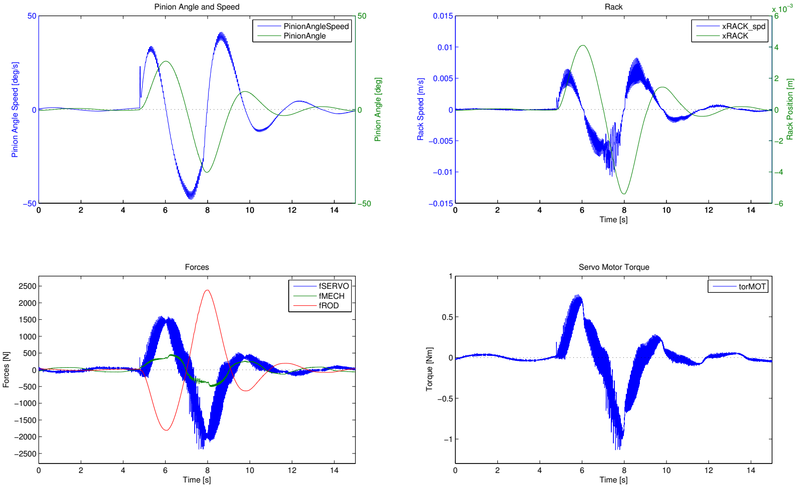


Figure 7.10: *Simulation results from simulation using the low pass filter and lead compensator for the steering signals from the driver on the scenario computer. Position prediction time 100 ms.*

7.4 Stability Analysis of Feedback System with Delay

As mentioned earlier, the communication between the scenario computer, with the driver, and the vehicle model includes latency. This latency, if not handled correctly, affects the stability of the complete system. In order to analyze how this latency affects the stability of the system, a simplified setup was developed. The vehicle model was replaced by a bicycle model and the driver was replaced by a Proportional Integral (PI) controller. The bicycle model was implemented as described in section 3.2 and the state space representation is shown below

$$A = \begin{bmatrix} \frac{-2C_{\alpha f}-2C_{\alpha r}}{V_x m} & -V_x - \frac{2L_1 C_{\alpha f}+2L_2 C_{\alpha r}}{V_x m} \\ -\frac{2L_1 C_{\alpha f}+2L_2 C_{\alpha r}}{V_x I_z} & -\frac{2L_1^2 C_{\alpha f}-2L_2^2 C_{\alpha r}}{V_x I_z} \end{bmatrix} \tag{7.1}$$

$$B = \begin{bmatrix} \frac{2C_{\alpha f}}{m} \\ \frac{2L_1 C_{\alpha f}}{I_z} \end{bmatrix} \tag{7.2}$$

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{7.3}$$

$$D = [0] \tag{7.4}$$

The delay on the feedback from the bicycle model to the driver was implemented using the following transfer function [10]

$$F_{delay} = e^{-\tau s} \tag{7.5}$$

The transfer function for the PI-controller was implemented as

$$F_{PI} = \frac{sK_P + K_I}{s} \tag{7.6}$$

The feedback from the bicycle model was the heading of the vehicle and the input was the steering angle. By feeding back the heading of the vehicle, the steering angle can be seen as a desired heading. A Nyquist plot of the so called loop transfer function [11], including the PI-controller, the bicycle model and the delayed feedback was done with the time delay *tau* alternated between 0 and 200 ms, and can be seen in 7.12a and Figure 7.12. The so loop transfer function was formed by breaking the closed loop right before the PI-controller and calculating the transfer function from one end to the other of the resulting open loop. In Figure 7.11 the studied system is shown together with marking of where the loop was opened.

As is shown in Figure 7.12, the delay on the feedback signal affects the stability of the closed loop. The only conclusion from this evaluation of the simplified model is the fact that the delay indeed decreases the stability of the feed back system. The Nyquist criteria says that the point (-1, 0) should not be encircled in order to have a stable closed loop system [11]. The reason the point (-1, 0) is the critical point is due to the fact that the loop transfer function will act as a gain with an amplitude of -1 and hence the feedback signal will be the same as the input when using the negative feedback, as seen in Figure 7.11. Studying Figure 7.12, the critical delay for the simple feedback system is around 180 ms. The actual system studied during the thesis has a latency of approximately 150 ms. This together with the discrete filter implemented in subsection 6.2.1 gives a delay close to the critical delay for the simplified system. For the real system, the complexity and the dynamics are far greater and stability can not be guarantied from this evaluation. Note also that this evaluation assumes continuous signals from the driver.

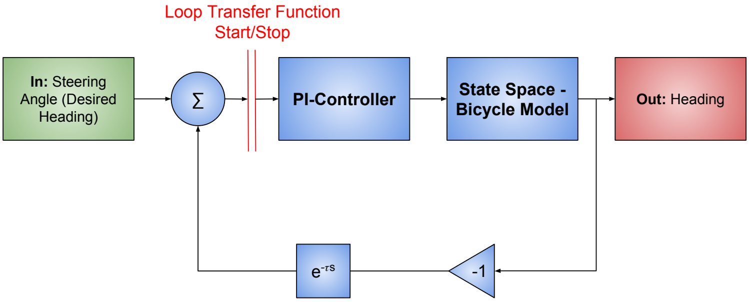
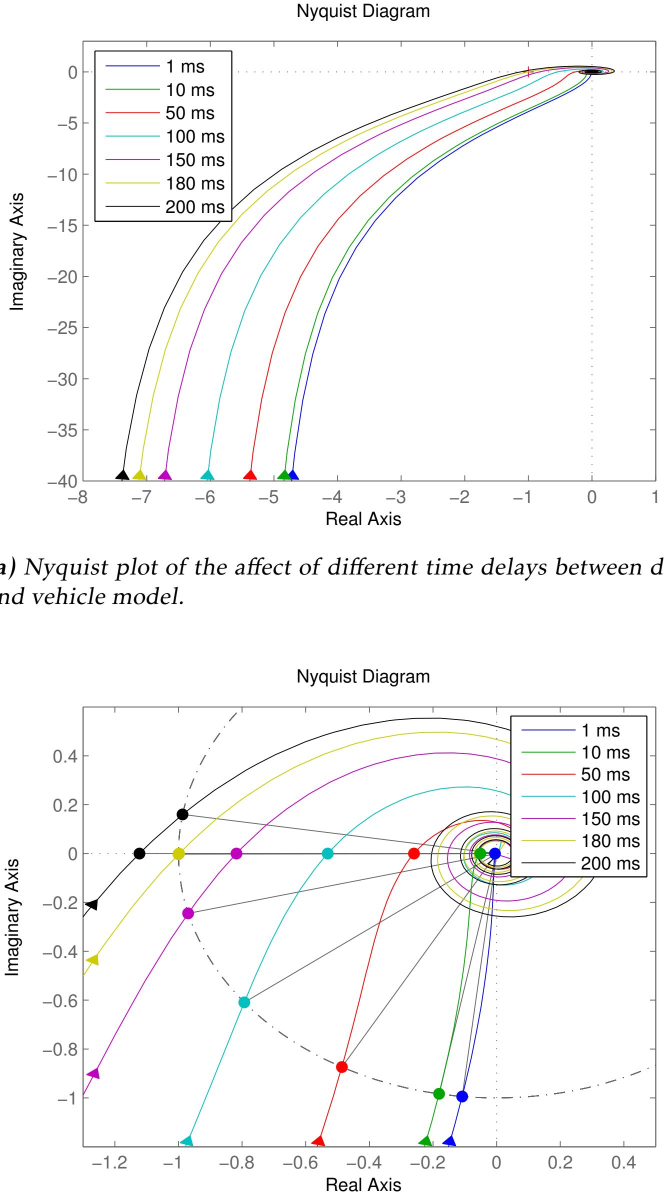


Figure 7.11: *Figure of the simple system used for stability analysis consisting of a bicycle model, PI-controller and a delayed feedback.*



***(a)** Nyquist plot of the affect of different time delays between driver and vehicle model.*

***(b)** Zoomed in - Nyquist plot of the affect of different time delays between driver and vehicle model.*

Figure 7.12

7.5 Discussion

The investigation in the HIL-environment indicates that there are several factors affecting the steering behaviour. One contributing factor is the latency in the simulation. This includes both the computational delay within the vehicle model as well as the latency in the feedback of the vehicle states to the scenario driver and computer.

By implementing a simple prediction trajectory the latency in the feedback is reduced, thus the behaviour of the signals in the simulations shows less oscillations. Even though this is a simplified prediction of the movement of the vehicle in the global directions, it is an effective way of showing the effect of the latency.

The latency further prevents use of models and functions which requires extensive computational time, for example discrete filters. The results from the simulation with the implemented discrete filter shows great improvement regarding the noise but this implementation also boosts the latency within the model, which increases the oscillations in the signals.

This result led to the implementation of the low pass filter with lead compensation. Although the results show improvement regarding the noise compared to simulations without any filter, the signals are not nearly as smooth as in the simulation with the discrete filter. The oscillations using the low pass with lead compensation is noticeable smaller than when using the discrete filter, so it is a matter of choice between obtaining smooth signals or accurate steering in terms of road position.

Before the implementation in the HIL-environment, the filter was tested in the open-loop offline environment with an input signal with a frequency rate of 50 Hz. Here, the low pass filter with lead compensation gave more effect. One big difference between these simulations is that within the HIL- environment, the input signal is not consistently sampled every 10 ms. As mentioned in section 6.2, the signals are kept constant during 20-40 ms. This makes it more difficult to tune the filter.

The results from the simulation where the driver on the scenario computer is bypassed and the signals are compared with validated data shows that the actual steering model behaves as expected. It is important to remember that the validated measurements are simulated in a MIL-environment and will thus naturally differ from the real environment. Validation against real measurements was not possible in this thesis due to confidentiality.

The problems associated with sending sampled signals into the steering model in combination with the stability issue with the latency between the driver and the vehicle model makes is quite hard to obtain satisfying results. It will be difficult to obtain good results in all aspects, thus the implemented improvements might not be enough for verification and study of the all the involved signals for lateral active safety functions.

When verifying the lateral active safety functions in the HIL, different aspects can be studied. One verification could be to test whether the controllers responds as expected in different situations. This means only the activation signals could be the signals to study. In this situation, the developed steering model would be enough. When studying signals from the servo motor in an active safety scenario, the behavior of the signal is more critical and more investigations would be required to obtain reliable results. However, if the latency introduced by the discrete filter is not crucial, the implementation using the discrete filter could be sufficient.

8

Conclusions and Future Work

The main goal of this thesis have been to investigate the integration and implementation of the new steering model in the HIL-environment, which includes an investigation on how the testing of the steering controller can be developed in the future. This has meant finding and researching the various problems and challenges with simulations in this kind of engineering system divided into multiple subsystems. The purpose of the various solutions have been to analyze what kind of problems have the biggest impact on the simulations. In this way, problems can be prioritized and this clarifies which solutions to investigate for future work.

The simulations with the implemented steering model gives successful results while simulated without the scenario driver, therefore the issues and challenges regarding developing a HIL-environment lays with the multi-rate signals and the latency in the coupled environment.

The biggest obstacle using this high dynamic steering model is the multi-rate problem regarding the input signals. Since the input signals from the scenario computer are kept constant at various sampling rates, more future investigation is needed within multi-rate methods. As mentioned earlier, using extrapolation that allows variable update frequencies on the input signals could be a solution to this.

Using the position prediction in order to reduce the effect of the latency between the vehicle model and the driver on the scenario computer, is a simple and efficient solution. However, this means that the position of the vehicle calculated by the vehicle model is overwritten and hence the vehicle behavior does not correspond to the simulation of the vehicle model. This should be avoided in the final simulation setup and was only implemented to highlight the fact that the oscillations in position is due to the latency. To make a better prediction of the position, a Kalman filter could be an option in order to correct the predictions.

As shown in section 7.4, the stability of a feedback system with a simple bicycle model can only be guaranteed for latency up to approximately 180 ms. The complexity in the simulation environment used in the HIL-environment is, as mentioned before, far greater. This means the latency in the system might be too big to guarantee stability using the implementation presented in the thesis.

Another option to improve the simulations is to replace the current environment with a product family that includes the vehicle model and the driver in the same subsystem. Hence, this would remove both the latency and the multi-rate problem between these two. There are several options on the market that offers this.

The extent of how much of the steering controller's performance can be tested in the HIL-environment is not known at this moment. It depends on the goal with the verification as well as the different aspects and signals which needs to be studied. Thus in order to fully use the HIL-environment for testing of the lateral steering, more investigation within the latency and the multi-rate methods are needed.

As discussed earlier, the road force into the steering model was not corresponding to the force from the validated MIL-environment. This must be investigated in order to get more accurate simulation results. The current vehicle model has an advance model for the road forces, but this needs to be handled correctly when sending it to the steering model. The road force needs to be translated into corresponding rod forces according to the current angle at the tie rods as well as only small delays should be allowed.

Bibliography

[1] Vivek Jaikamal. Model-based ecu development – an integrated mil-sil-hil approach. *ETAS Inc.*, 2009. Cited on pages 2 and 5.

[2] Martin Arnold. Multi-rate time integration for large scale multibody system models. *Martin Luther University Halle-Wittenberg*, 2007. Cited on pages 3 and 7.

[3] Naya M.Á. Luaces A. González, F. and M. González. On the effect of multirate co-simulation techniques in the efficiency and accuracy of multibody system dynamics. *Multibody System Dynamics*, 2011. Cited on page 3.

[4] Horn Martin Bededikt Martin Zehetner Josef Stettinger, Georg. Model-based coupling approach for non-iterative real-time co-simulation. *Control Conference (ECC), 2014 European*, 2014. Cited on pages 3 and 35.

[5] R. Kubler and W Schiehlen. Two methods of simulator coupling, mathematical and computer modelling of dynamical systems. *Taylor & Francis*, 2010. Cited on page 7.

[6] Watzenig Daniel Benedikt, Martin and Anton Hofer. Modelling and analysis of the non-iterative coupling process for co-simulation. *Taylor & Francis*, 2013. Cited on page 7.

[7] Manfred Harrer and Peter Pfeffer. *Steering Handbook*. Springer International Publishing AG Switzerland, first edition, 2014. Cited on page 9.

[8] How electric power assisted steering works, and why it's better than hydraulic. `https://www.carthrottle.com/post/electronic-power-assisted-steering-how-does-it-work/`. Accessed: 2017-06-02. Cited on page 10.

[9] J.Y Wong. *Theory of Ground Vehicles*. John Wiley & Sons, Inc., fourth edition, 2008. Cited on page 12.

[10] Torkel Glad and Lennart Ljung. *Reglerteknik - Grundläggande teori*. Studentlitteratur AB., fourth edition, 2006. Cited on pages 36 and 48.

[11] Johan Åström and Richard M. Murray. *Feedback Systems - An introduction for Scientists and Engineers*. Princeton University Press., - edition, 2008. Cited on page 49.

Index

ARMA
 abbreviation, x
ASDM
 abbreviation, x

ECU
 abbreviation, x
EPAS
 abbreviation, x

FOH
 abbreviation, x

HIL
 abbreviation, x

LKA
 abbreviation, x

MIL
 abbreviation, x

PID
 abbreviation, x
PSCM
 abbreviation, x

SIL
 abbreviation, x
STM
 abbreviation, x

ZOH
 abbreviation, x