HDV Simulink Real World Model for Testing

A Study on Model Simplicity

ROBIN LINDSTRÖM
HDV Simulink Real World Model for Testing

A Study on Model Simplicity

Robin Lindström

Master’s Thesis in Optimization and Systems Theory (30 ECTS credits)
Master Programme in Aerospace Engineering (120 credits)
Royal Institute of Technology year 2016
Supervisor at Scania CV AB: Gustav Åberg
Supervisor at KTH: Per Enqvist
Examiner: Per Enqvist

TRITA-MAT-E 2016:40
ISRN-KTH/MAT/E--16/40--SE

Royal Institute of Technology
School of Engineering Sciences
KTH SCI
SE-100 44 Stockholm, Sweden
URL: www.kth.se/sci
Abstract

The purpose of this thesis was to investigate the area of model complexity and see what precautions could be made to avoid the generation of complex models. These findings included five principles stated by M. Pidd which were implemented in the development of a new Simulink real world model of a heavy duty vehicle.

The developed model was integrated into a hardware-in-the-loop configuration and parameterized and validated by comparing the new model to the previous model in several advanced emergency brake test cases. Additional testing was done to prove the overall robustness and validity of the new model.

With an average relative error of 2% compared to measurement data of the previous model and the additional testing considered passed, the new model was deemed valid and representative of a heavy duty vehicle. The study recommends that further simplifications are done in the hardware integration framework, which have not been targeted in this thesis, to reduce the complexity of the overall model further.

All and all the new model met the deliveries and due to its less complex structure and maintained validity it was implemented at the REVt department at Scania CV AB.
Simulink omvärldsmodell av tungt fordon för test
En studie i enkelhet

Sammanfattning

Syftet med det här examensarbetet var att undersöka ämnet modell komplexitet och se vilka åtgärder som kan vidtas för att undvika skapandet av komplexa modeller. Studien inkluderar fem principer skrivna av M. Pidd vilka implementerades i skapandet av en ny Simulink omvärldsmodell av ett tungt fordon.

Den utarbetade modellen integrerades i ett hardware-in-the-loop ramverk och parametriserades och validerades genom att jämföra den nya modellen mot den gamla i ett flertal testfall för den avancerade nödbromsen. Ytterligare tester genomfördes även för att bekräfta hållbarheten och giltigheten hos den nya modellen.

Med ett snitt relativt fel på 2 % jämfört med mätdata från den tidigare modellen och godkänt resultat från de ytterligare testerna, ansågs den nya modellen vara en giltig och representativ avbild av ett tungt fordon. Studien rekommenderar att ytterligare förenklingar görs på hårdvaruramverket, som inte har berörts i det här arbetet, för att reducera komplexiteten hos modellen i helhet ytterligare.

Överlag så levde modellen upp till alla krav och på grund av dess reducerade komplexa struktur och bibehållna giltighet, implementerades den på REVT avdelningen på Scania CV AB.
Acknowledgements

This master thesis was carried out at the ECU system test department REVT at Scania CV AB, a part of the Research & Development in Södertälje, Sweden.

First of all I would like to thank my industrial supervisor Gustav Åberg at the REVT department for his invaluable guidance throughout the entire thesis. A special thanks goes out to Carl-Johan Sjöstedt for explaining and guiding me through the existing model framework. I would also like to thank the rest of the team at the REVT department as well as my fellow thesis worker Sophia Bäckström for their support and input. An additional thanks goes out to Tobias Roswall at the Scania NEVC department for his wide vehicle knowledge.

Finally I would like to thank my supervisor at KTH Per Enqvist for making it all possible.

- Robin Lindström, Stockholm June 2016
# Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>Anti-lock Braking System</td>
</tr>
<tr>
<td>AEB</td>
<td>Advanced Emergency Brake</td>
</tr>
<tr>
<td>CAN</td>
<td>Controller Area Network</td>
</tr>
<tr>
<td>ECU</td>
<td>Electronic Control Unit</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HDV</td>
<td>Heavy Duty Vehicle</td>
</tr>
<tr>
<td>HIL</td>
<td>Hardware-in-the-Loop</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>MAAB</td>
<td>MathWorks Automotive Advisory Board</td>
</tr>
<tr>
<td>MDL</td>
<td>Model</td>
</tr>
<tr>
<td>R&amp;D</td>
<td>Research and Development</td>
</tr>
<tr>
<td>SEI</td>
<td>Software Engineering Institute</td>
</tr>
<tr>
<td>SIL</td>
<td>Software-in-the-Loop</td>
</tr>
<tr>
<td>SUT</td>
<td>System Under Test</td>
</tr>
</tbody>
</table>
### Symbols

<table>
<thead>
<tr>
<th>Variable</th>
<th>Quantity</th>
<th>Unit of measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_{slope}$</td>
<td>Road grade</td>
<td>[rad]</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Relative error</td>
<td>[1]</td>
</tr>
<tr>
<td>$\mu_{brake}$</td>
<td>Frictional coefficient of the service brakes</td>
<td>[1]</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Air density</td>
<td>[kg/m$^3$]</td>
</tr>
<tr>
<td>$\omega_{diff}$</td>
<td>Rotational speed of the differential shaft</td>
<td>[rad/s]</td>
</tr>
<tr>
<td>$\omega_{drive}$</td>
<td>Rotational speed of the driveshaft</td>
<td>[rad/s]</td>
</tr>
<tr>
<td>$\omega_{eng}$</td>
<td>Rotational speed of the crankshaft leaving the engine</td>
<td>[rad/s]</td>
</tr>
<tr>
<td>$\omega_{maxstart}$</td>
<td>Maximum rotational speed limit in the start engine</td>
<td>[rad/s]</td>
</tr>
<tr>
<td>$\omega_{trans}$</td>
<td>Rotational speed of the propeller shaft leaving the transmission</td>
<td>[rad/s]</td>
</tr>
<tr>
<td>$A$</td>
<td>Vehicle front area</td>
<td>[m$^2$]</td>
</tr>
<tr>
<td>$C_D$</td>
<td>Drag coefficient</td>
<td>[1]</td>
</tr>
<tr>
<td>$C_{rr}$</td>
<td>Roll resistance coefficient</td>
<td>[1]</td>
</tr>
<tr>
<td>$F_{atr}$</td>
<td>Force due to air resistance</td>
<td>[N]</td>
</tr>
<tr>
<td>$F_{brake}$</td>
<td>Brake force</td>
<td>[N]</td>
</tr>
<tr>
<td>$F_{brake_{req}}$</td>
<td>Requested brake force</td>
<td>[N]</td>
</tr>
<tr>
<td>$F_{gravity}$</td>
<td>Longitudinal gravitational force</td>
<td>[N]</td>
</tr>
<tr>
<td>$F_{propulsion}$</td>
<td>Propulsion force</td>
<td>[N]</td>
</tr>
<tr>
<td>$F_{road}$</td>
<td>Longitudinal road forces acting on the tires</td>
<td>[N]</td>
</tr>
<tr>
<td>$g$</td>
<td>Gravitational acceleration</td>
<td>[m/s$^2$]</td>
</tr>
<tr>
<td>$i_{diff}$</td>
<td>Exchange ratio in the differential</td>
<td>[1]</td>
</tr>
<tr>
<td>$i_{gear}$</td>
<td>Exchange ratio in the transmission</td>
<td>[1]</td>
</tr>
<tr>
<td>$J_{clutch}$</td>
<td>Rotational inertia of the clutch</td>
<td>[kgm$^2$]</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Unit</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>$J_{crank}$</td>
<td>Rotational inertia of the crankshaft</td>
<td>[kgm$^2$]</td>
</tr>
<tr>
<td>$J_{drive}$</td>
<td>Rotational inertia of the driveshaft</td>
<td>[kgm$^2$]</td>
</tr>
<tr>
<td>$J_{gear_in}$</td>
<td>Rotational inertia of the ingoing gears</td>
<td>[kgm$^2$]</td>
</tr>
<tr>
<td>$J_{gear_out}$</td>
<td>Rotational inertia of the outgoing gears</td>
<td>[kgm$^2$]</td>
</tr>
<tr>
<td>$J_{start}$</td>
<td>Rotational inertia of the start engine</td>
<td>[kgm$^2$]</td>
</tr>
<tr>
<td>$J_{vehicle}$</td>
<td>Rotational inertia of the vehicle acting on the tires</td>
<td>[kgm$^2$]</td>
</tr>
<tr>
<td>$K_{damp}$</td>
<td>Gain due to damping in the clutch</td>
<td>[Nm/(rad/s)]</td>
</tr>
<tr>
<td>$K_{exhb}$</td>
<td>Input gain in the exhaust brake</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$m$</td>
<td>Vehicle mass</td>
<td>[kg]</td>
</tr>
<tr>
<td>$r_{tire}$</td>
<td>Tire radius</td>
<td>[m]</td>
</tr>
<tr>
<td>$T_{brake}$</td>
<td>Brake torque</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{diff}$</td>
<td>Torque entering the differential</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{eng}$</td>
<td>Engine torque</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{engloss}$</td>
<td>Engine torque loss</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{exhb}$</td>
<td>Brake torque generated by the exhaust brake</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{fric}$</td>
<td>Friction torque generated by the clutch</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{gear}$</td>
<td>Friction torque generated in the transmission</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{maxclutch}$</td>
<td>Maximum torque limit in the clutch</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{maxeng}$</td>
<td>Maximum engine torque</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{maxexhb}$</td>
<td>Maximum exhaust brake torque</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{maxstart}$</td>
<td>Maximum torque produced by the start engine</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{req}$</td>
<td>Requested engine torque</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{ret}$</td>
<td>Brake torque generated by the retarder</td>
<td>[Nm]</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Unit</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>$T_{\text{retloss}}$</td>
<td>Torque loss generated in the retarder</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{\text{retmap}}$</td>
<td>Available brake torque in the retarder</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{\text{start}}$</td>
<td>Torque produced by the start engine</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{\text{tire}}$</td>
<td>Torque acting at the tires</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$T_{\text{trans}}$</td>
<td>Torque entering the transmission</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$u_{\text{clutch}}$</td>
<td>Clutch input</td>
<td>[1]</td>
</tr>
<tr>
<td>$u_{\text{exhbrq}}$</td>
<td>Exhaust brake input</td>
<td>[1]</td>
</tr>
<tr>
<td>$u_{\text{ret}}$</td>
<td>Retarder input</td>
<td>[1]</td>
</tr>
<tr>
<td>$v$</td>
<td>Vehicle speed</td>
<td>[m/s]</td>
</tr>
<tr>
<td>$\dot{v}$</td>
<td>Vehicle acceleration</td>
<td>[m/s$^2$]</td>
</tr>
<tr>
<td>$v_0$</td>
<td>Wind speed, in direction of driving</td>
<td>[m/s]</td>
</tr>
<tr>
<td>$v_{\text{original}}$</td>
<td>Average derivative of the distance to the object of the original model</td>
<td>[m/s]</td>
</tr>
<tr>
<td>$v_{\text{project}}$</td>
<td>Average derivative of the distance to the object of the thesis model</td>
<td>[m/s]</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1. Shift in fault and cost distributions as process maturity increases. .............................................. 2
Figure 2. Radar (red) and camera (blue) coverage illustration of Scania Advanced Emergency Brake system. ................................................................................................................ 12
Figure 3. Acting longitudinal forces on a HDV. ......................................................................................... 16
Figure 4. Layout model of the powertrain and brake system. ................................................................. 17
Figure 5. Simulink block diagram example of the service brake .......................................................... 22
Figure 6. Simulink subsystem block example of the service brake illustrated in Figure 5. .......... 22
Figure 7. Block illustration of the model. ............................................................................................... 23
Figure 8. Illustration of model, ECU and hardware integration............................................................. 24
Figure 9. Illustration of a HIL simulation setup. .................................................................................... 25
Figure 10. Real-time model preparation flowchart. ................................................................................ 27
Figure 11. Real-time simulation flowchart. ............................................................................................. 28
Figure 12. Hardware-in-the-loop simulation flowchart. ....................................................................... 29
Figure 13. ControlDesk interface example. ............................................................................................ 31
Figure 14. Shows the difference in vehicle speed between both models. ........................................... 32
Figure 15. Shows the difference in how the distance to the obstacle varies between both models. ............................................................................................................................... 33
Figure 16. Shows the difference in engine speed between both models. .......................................... 33
Figure 17. Detailed view of the first obstacle in Figure 15 adjusted for time-shift. ............................ 35
Figure 18. Maximum available engine torque as a function of engine speed. ................................... 39
Figure 19. Engine parasitic losses as a function of engine speed. ....................................................... 40
Figure 20. Gear exchange ratio as a function of gear input. ................................................................. 40
Figure 21. Brake torque generated by the retarder as a function of lever input and engine speed. ........................................................................................................................................ 41
Figure 22. Torque loss generated by the retarder as a function of engine speed. ................................ 41
Figure 23. Exhaust brake torque as a function of engine speed. ............................................................ 42
Figure 24. Block structure illustration of the previous model. ............................................................. 43

List of tables

Table 1. Relative error for each appearing obstacle avoidance.............................................................. 34
Table 2. Relative error for each appearing obstacle avoidance.............................................................. 36
# Table of Contents

## Innehåll

Chapter 1 .................................................................................................................... 1  
  Background ........................................................................................................... 1  
  Problem description .............................................................................................. 2  
  Purpose ................................................................................................................... 3  
  Delimitations ......................................................................................................... 3  
  Method .................................................................................................................. 3  
  Report outline ....................................................................................................... 4  

Chapter 2 .................................................................................................................. 5  
  Complexity vs. simplicity ..................................................................................... 5  
  Five simple principles of modelling ..................................................................... 7  
    Principle 1. Model simple, think complicated .................................................... 7  
    Principle 2. Be parsimonious, start small and add .............................................. 8  
    Principle 3. Divide and conquer, avoid mega-models ...................................... 8  
    Principle 4. Do not fall in love with data ............................................................ 8  
    Principle 5. Model building may feel like muddling through ......................... 10  

Chapter 3 .................................................................................................................. 12  
  Advanced Emergency Brake .................................................................................. 12  
  Test cases in general ............................................................................................ 13  
  AEB functionality test cases ................................................................................. 14  

Chapter 4 .................................................................................................................. 15  
  Acting forces ........................................................................................................... 15  
    Propulsion force ................................................................................................. 16  
    Road forces ......................................................................................................... 16  
    Brake forces ........................................................................................................ 16  
    Air resistance ...................................................................................................... 17  
    Road grade ......................................................................................................... 17  
  Powertrain ............................................................................................................ 17  
  Engine ................................................................................................................... 18
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starter</td>
<td>18</td>
</tr>
<tr>
<td>Crankshaft</td>
<td>18</td>
</tr>
<tr>
<td>Clutch</td>
<td>19</td>
</tr>
<tr>
<td>Transmission</td>
<td>19</td>
</tr>
<tr>
<td>Gear train</td>
<td>19</td>
</tr>
<tr>
<td>Differential</td>
<td>20</td>
</tr>
<tr>
<td>Brakes</td>
<td>20</td>
</tr>
<tr>
<td>Simulink subsystem blocks</td>
<td>21</td>
</tr>
<tr>
<td>Model structure</td>
<td>22</td>
</tr>
<tr>
<td>Model and hardware integration</td>
<td>23</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>25</td>
</tr>
<tr>
<td>Real-time model preparation</td>
<td>26</td>
</tr>
<tr>
<td>Real-time model simulation</td>
<td>27</td>
</tr>
<tr>
<td>Hardware-in-the-loop simulation</td>
<td>28</td>
</tr>
<tr>
<td>Chapter 6</td>
<td>30</td>
</tr>
<tr>
<td>Model verification and parameterization</td>
<td>30</td>
</tr>
<tr>
<td>Component wise parameterization and verification</td>
<td>30</td>
</tr>
<tr>
<td>Reference data</td>
<td>30</td>
</tr>
<tr>
<td>Full-scale simulation</td>
<td>31</td>
</tr>
<tr>
<td>Additional testing</td>
<td>32</td>
</tr>
<tr>
<td>Simulated results</td>
<td>32</td>
</tr>
<tr>
<td>Error analysis</td>
<td>34</td>
</tr>
<tr>
<td>Discussion</td>
<td>34</td>
</tr>
<tr>
<td>Chapter 7</td>
<td>37</td>
</tr>
<tr>
<td>Conclusion</td>
<td>37</td>
</tr>
<tr>
<td>Future work</td>
<td>38</td>
</tr>
<tr>
<td>Appendix A</td>
<td>39</td>
</tr>
<tr>
<td>Appendix B</td>
<td>43</td>
</tr>
<tr>
<td>Bibliography</td>
<td>44</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Background

As technology increases\(^1\), systems tend to grow more and more complex. In the automotive industry the opportunities and challenges faced are immense. The automotive sector is Europe’s largest private investor in Research and Development (R&D) (European Commission, 2016). Today’s vehicles have everything from cruise control to driver support and automatic braking systems. The demand of functioning and reliable systems is increasing while the amount of systems in need of testing is growing at the same rate. These demands puts a lot of pressure on quality control and thereby the testing departments. In turn, this pressure increases the need to run more tests in a given time frame while still maintaining validity.

The process of testing a product is very time consuming and expensive if done manually. In order to shorten the process and reduce the cost, simulation is used. Advantages of using simulation in the automotive industry were discussed among the colleagues at the REVT department at Scania CV AB and the results are presented below.

**Time.** Simulated tests are executed by software tools and is therefore significantly faster. Let’s say one is to test an Electronic Control Unit (ECU) in several different setups. Doing this manually would require the tester to install it in one setup, run the tests, remove it and install it in a different setup, run another tests and so on. By simulation one can do all this automatically and swap setup with the click of a button or with the help of a script.

**Easy to reproduce.** Some tests are harder to reproduce than others, especially tests that are influenced by human error. However, through simulation, one can easily run the exact same test over and over until a sufficient amount of data is collected.

**No need for prototypes.** Prototypes are expensive and time consuming to make. By simulating one can easily change the dynamics of the model without having to build a new prototype, simply by changing the dynamics of the code.

**Emphasis on the part being tested.** Products are often dependent on each other. Sometimes it might be hard to test one specific component without being affected by the dynamics of other components. When simulating one can design the test such that it tests the desired part in its intended position without running a risk of damaging other components in ways that might not be doable manually.

**Safer/easier to test extreme conditions.** Some errors might only occur during extreme conditions. These conditions might be hard or even dangerous to meet. By simulation one can

---

\(^1\) The old but arguably still in effect Moore’s law states that computing would dramatically increase in power, and decrease in relative cost, at an exponential pace (Moore, 1965).
push a system to extreme limits without endangering personnel or damaging other parts of the system that might not be designed for the given conditions.

Another perk of testing in the development process is the ability to find errors and faulty assumptions at an early stage. Figure 1 illustrates the distribution of fault introduction and when in the development process a fault is discovered based on the maturity scale of the SEI’s Capability Maturity Model (Paulk, Curtis, & Chrissis, 1993). As seen in the bottom of Figure 1, the cost grows exponentially throughout the development process.

Within the automotive industry real world models, models used to simulate vehicles and their surroundings, are being developed and simulated in order to test and validate systems. Modelling and simulation mostly fall under the system test category in Figure 1 and the data presented shows the importance of reliable and accurate simulation models and test cases.

![Figure 1. Shift in fault and cost distributions as process maturity increases. Data and image from (Park, Goethert, & Florac, 1996, p. 58)](image)

Problem description

As of today, the models used during simulation are quite extensive and include areas that might not be of relevance to the system under test, SUT. This arguably unnecessary complexity results in an increased time frame of the testing process, due to longer computational times, more complicated debugging, higher requirement in maintenance and, in some cases, unnecessary errors due to impact from nonrelated aspects of the simulation model.
Purpose

The area of model complexity, despite its importance, appears to remain at the bottom of simulation research agendas (Chwif, Barretto, & Paul, 2000, p. 449). The purpose of this thesis is to investigate the complexity of simulation models and how this affects computational time and validity.

**Deliveries** The thesis consists of a literature study investigating what previous work has been made in the area, the advantages of using a less complex model in the testing process and how this affects the results of the simulation. A less complex model was developed guided by the principles stated in Five Simple Principles of Modelling (Pidd, 1996) and compared to the previously existing model in order to prove validity.

Delimitations

*In order to meet the deliveries of the thesis within the given time frame of 20 weeks, some delimitations were stated.*

The primary focus of the literature study was to investigate benefits of simplifying complex models within the automotive industry.

The study’s main focus was to investigate the case of an Advanced Emergency Brake (AEB) in a Hardware-in-the-Loop (HIL) simulation and how a simplified model of the Heavy Duty Vehicle (HDV), following Five Simple Principles of Modelling (Pidd, 1996), affect the validity and runtime of the simulation. The reason for this specific test case is the fact that AEB utilizes a lot of the HDV model dynamics in its simulation process. It is therefore considered to be a good representation of a test case that requires a more complex model than most. The idea is to determine what level of model complexity is required in order to validate it with results from the AEB test case.

All constants are referred to as non-specific values and all graphs are normalized and the axis values are removed to protect the integrity of the model, the test cases and the AEB functionality by request from Scania CV AB. The general behaviour of the model is considered enough to prove validity.

Method

This project started with a literature study of what previous work had been made in the area of model complexity and how complexity should be handled. The study focused on benefits of using a less complex model but also the disadvantages of reducing model complexity. Guidelines in how to handle complexity were found and decided to be the foundation on which the new model should be built upon. The literature search included conference papers, books, scientific articles and Master-/PhD-thesis.

After the research the previous Simulink model and the AEB test cases later used for parameterization and validation were analysed. The hardware interface connecting the model
to the HIL rig was analysed in order to find restrictions and requirements to ensure a smooth integration of the new model. Reference data was gathered from the previous model and prepared for parameterization.

The new model was developed component wise in Simulink based on basic dynamical knowledge of each component starting from the engine and following the powertrain to the wheels. Necessary input signals were constructed in order to parameterize and tested each component as thoroughly as possible.

All components were combined into a complete model which went through a final parameterization phase. The model was integrated into the hardware interface structure and validated by being compared to the previous model. The final product was uploaded in the HIL rig and ran through several test suites in ControlDesk and even overnight testing to ensure its robustness and validity.

The data was logged and analysed in CANalyzer and compared to the previous model in MATLAB. Any errors found were traced back to its source and sorted to the extent possible. Additional testing were made until the model was considered robust and valid and any remaining errors could be discarded or traced back to other sources than the model itself.

Report outline

- **Chapter 1** covers the background of the problem along with a description of the problem and the thesis method.
- **Chapter 2** investigates what previous work has been done in the area of model complexity and lists five principles used to help handle complexity when developing a model.
- **Chapter 3** focuses on the functionality of the AEB system in general and structure of the AEB test case.
- **Chapter 4** covers the components and dynamics included in the model and the process of structuring and integrating the model.
- **Chapter 5** explains hardware-in-the-loop and covers the process of preparing a model for HIL simulation.
- **Chapter 6** lists and discusses the results obtained during the thesis.
- **Chapter 7** summarizes the conclusions drawn in the thesis and discusses future work.
Chapter 2

Related work

This section covers related work done in the area and what conclusions were drawn from this earlier work. It also lists five principles stated by M. Pidd that were used as guidelines throughout the development process in this thesis.

Complexity vs. simplicity

Not much work has been done in the area of simulation model complexity. In the paper, On Simulation Model Complexity (Chwif, Barretto, & Paul, 2000), the authors speculate on the reasons for the lack of attention this area is receiving and why most modellers still tend to develop large and complex models. The authors start by investigating the concept of a complex model. The following section is dedicated to list the findings regarding model complexity made in On Simulation Model Complexity and discuss the reason why complexity is being used to the extent it is.

There is no single definition of a complex model. Two common interpretations are that model complexity is defined by the difficulty of understanding the system being modelled, or that model complexity is defined by the number of parts and elements included in the system. What both these conceptions share are the focus on “scope” and “level of detail”. As an example, consider the simulation of a production line. Scope is referred to as the focus of the model; is the model limited to a specific station in the production line or is it covering the entire line? The scope is reduced in the former case. Level of detail is referred to as the amount of detail included in the model; are assumptions and simplifications made or is every element of reality included to the best of the modeller’s ability? The level of detail is increased in the latter case.

Chwif, Barretto and Paul (2000) investigated the generation of complex models and why these models are, more often than not, still being the way to go by most developers. The authors have divided the reasons for increasing complexity into two categories; technical and non-technical reasons. The non-technical factors discovered are:

1. **“Show off” factor:** When showing a model to a manager a more complex model has more of an impact. Most people also think that a more complex model was more difficult to make, which is not always the case.

2. **“Include all” syndrome:** Inexperienced modellers might feel “insecure” about what to include in the model and as a safety measurement try to include everything that is possible in the model, following the maxim “It is better the excess than the lack”.

3. **“Possibility” factor:** With the increase in computation power, arguably still following Moore’s law (Moore, 1965), complexity and size is not a constraint as it used to be. Some modellers use this as an argument in order to create complex models just because computer power allows it.

The technical factors listed are:
1. **Lack of understanding of the real system**: In order to properly model a system one must first understand the system being modelled. If one's understanding of the system is lacking the resulting model will most likely be lacking in the same areas.

2. **Inability to model the problem correctly**: The lack of ability to model the problem correctly is common. Modelling is an abstraction of reality and the results should be close to reality, not necessarily the model itself. The reason to model a system in the first place is to avoid the complexity of reality.

3. **Inability to translate or code the conceptual model into a computerized model or lack of the simulation software knowledge**: Complex models may also be a result of the modeller’s lack of understanding the simulation software or lack of good programming skills. This might lead to the programmer generating code in an unnecessarily complex way and as a result a more complex computerized simulation model.

4. **Unclear simulation objectives**: Unclear or poorly defined objectives will most likely directly affect the complexity of the resulting model. “Where the overall aim is poorly defined, the anxious simulationist may draw the bounds of the model too wide, in the hope of including whatever it is that the user is really interested in.” (Salt, 1993, p. 2)

Chwif, Baretto and Paul (2000) also discussed advantages and disadvantages of using a less complex model. Some of the possible benefits are:

1. **Easy implementation, validation and analyzation**: A less complex model is easier to implement simply due to its self-explanatory lack of complexity compared to a complex model. It is also easier to trace and follow the “gear works” of a less complex models which helps in the validation and analyzation process.

2. **Easier to change or discard a less complex model**: A less complex model is more adaptable and easier to change if the conditions and hypothesis of the system are changed throughout the development process. It is also easier to let go and discard a less complex model if discovered that it is wrong or unreliable simply because it is easier to accept the failure of a cheaper model than a complex and expensive one.

3. **Time to complete a simulation study is reduced**: A less complex model often results in a shortened simulation study. It might also be the way to go if one is required to produce results within a short time frame; “it is infinitely better to have results (even approximate) of a simpler model before the deadline of the simulation study than to have the results of a highly complex model after the deadline” (Pegden, Sadowski, & Shannon, 1995).

Some drawbacks of using a less complex model are:

1. **Problem of validity**: A model must be complicated if necessary, but just enough. An oversimplified model may result in a validity loss. If a phenomenon requires certain aspects in order to be a representation of reality these aspects must be included. Discarding these aspects may result in a highly unstable model that is invalid in cases where the parameters deviate from their initial conditions.
2. **Problems of scope reduction:** One way to achieve simplicity is to reduce the scope of the model. By limiting the scope one also limits the flexibility, meaning that the model quickly becomes very specific and sensitive to changes. Changing one component in a narrow scoped model may result in loss of validity or in the worst case changing a component might not even be possible without having to rework the entire model.

3. **Difficulties in understanding:** Sometimes simple models are achieved by using abstraction. As long as everyone know what kind of abstraction has been made this will not result in a problem, but as soon as someone new is handled the model that person might fail to grasp what kind of abstraction has been made. This reduces the overall understanding of the model.

As seen in the list above, most drawbacks of using a less complex model comes from oversimplifying it. There is a fine line when simplifying a model further just makes it more complicated. There is a saying;

"Everything should be made as simple as possible, but not simpler."

– Albert Einstein (1879 – 1955)

This saying compactly articulates the principles of parsimony stated in Occam’s razor, often compactly summarized as *Entities are not to be multiplied beyond necessity* (Baker, 2013). Salt (1993, p. 1) also stresses the principles of parsimony by writing: “The whole modelling process consists in capturing the important elements of a system, and leaving out the rest”. Udi Dahan (2012) takes it one step further by writing: “Don’t try to model the real world, it doesn’t exist… Recognize that, like models, all perceptions are wrong, but some may be useful. Model the perceptions. Forget about reality.”

In other words, a model should be as simple as possible but still complex enough to maintain validity and help the understanding for future applications. It is impossible to include every aspect of reality, focus on the what can be perceived and what is needed. With all this in mind, constructing a model is not an easy task.

**Five simple principles of modelling**

In order to add some structure to the modelling process M. Pidd stated guidelines in a paper called *Five Simple Principles of Modelling* (1996). These guidelines were meant to help developers tackle complexity in a constructive fashion.

**Principle 1. Model simple, think complicated:** There is a widespread misconception among modellers that complex systems require complicated models. Pidd believes that this misconception has is origin from the principle of variety, due to Ashby (Ashby & Pierce, 1957), which states that “variety must match variety”. The reason for why this misconception is false is the simple fact that “models are not just built, they are also used” (Pidd, 1996, p. 722). By this statement Pidd refers to the fact that a model alone must not match the system being modelled, but the model and the user(s) combined must. In this case the model and the user(s) form a system. As long as the model requires a user to operate, the relation holds. The idea of viewing the model and user(s) as a system is the reason for the title of the first principle; “model simple, think complicated”.

A simple model with the correct interpreting user does not require as much complexity as long as the system and the user forms is a valid representation of reality. Models can be seen as a “tool for thinking”.

There is however a contradiction to this view on models; a highly complex model might not be in need of much critical thinking. In the case of autonomous systems, which carry on their day-to-day operation without human intervention, the use of complexity is essential. It is very important that an autonomous system is independent of a thinking user, for self-explanatory reasons. Testing systems (the area of focus in this thesis), however, are highly dependent on a thinking entity as the results are analysed and interpreted by the user(s). The view on modelling as a tool for thinking is therefore considered to be a valid principle to use as a guideline when developing a model for testing.

**Principle 2. Be parsimonious, start small and add:** When developing a simulation model one problem might be knowing how simple or complicated it should be. In the second principle, Pidd suggests employing the Principle of Parsimony, also known as Occam’s razor; *Entities are not to be multiplied beyond necessity* (Baker, 2013). It is sometimes also known as the acronym K.I.S.S., which stands for *Keep It Simple, Stupid*. The idea of the principle is to develop the model gradually; start small and add. Instead of assuming the level of complexity needed one should start with simple, manageable assumptions and add, gradually, from there whenever this is necessary.

**Principle 3. Divide and conquer, avoid mega-models:** Complex systems tend to look rather chaotic. A common advice, and main idea this principle, is to divide the model into smaller and more manageable models. Powell (1995) refers to this as *decomposition*, as well as *divide and conquer*. Pidd also quotes Raiffa’s (1982) summarization of the idea to divide and conquer:

> “Beware of general purpose, grandiose models that try to incorporate practically everything. Such models are difficult to validate, to interpret, to calibrate statistically and, most importantly to explain. You may be better off not with one big model but with a set of simpler models.”

In conclusion, “the best way to face complexity is to break it down into manageable chunks” (Pidd, 1996, p. 723).

**Principle 4. Do not fall in love with data:** A lot of modellers rely heavily on the existence of data. They assume that progress cannot be made unless data is available. This, however, may be a mistake. Exploratory data analysis might be a very valuable technique. Today’s software easily enables plotting of large amount of data and patterns are quickly discovered. But, to quote Pidd, “sometimes these patterns exist, but at other times they are, like beauty, in the eyes of the beholder.” (1996, p. 724)

It is very important that the model drive the data collection and not the other way around. This is a common misconception from most case-style teaching where students are presented with all the data they may need in the papers issued with the case. When in reality data must be requested, justified and collected. The process of collecting and analysing data is not cheap, and it is therefore important that the modeller knows what data is needed before requesting it.
Pidd states that the best way to link data to the parsimonious approach is to, if circumstances permit, develop a simple model and then request or collect data in order to parameterise and validate it. This may prove that the less complex model will suffice for the intended purpose or that it needs to be refined. If the model needs further data the modeller knows the cost of this data and can present a cost-benefit calculation justifying the collection of more data. This process can go on until the cost-benefit ratio reaches a point where the cost outweighs the benefit.

Pidd addresses some specific issues regarding data that he chooses to divide into five subprinciples.

1. **Data mining and data grubbing:** Data mining is the name of the process of trying to develop statistical models from available data. The data is searched for patterns with the use of powerful computer packages. Data grubbing is the debasement version of data mining. It is when many different data series are unthinkingly collected and then analysed by these powerful statistical packages.

   The problem with this type of analysis is that it can be done very quickly by someone who know close to nothing about the statistical tools being used or the origin of the data. Pidd compares this to “an attempt to bake a cake by collecting ingredients that look interesting, then mixing them together until boredom sets in, popping the resulting melange into the oven and waiting for the smoke to appear.” (1996, p. 724) Data should not be a substitute for thought. Just because data is available does not mean that it is useful.

2. **Data is useful in model building:** In spite of the criticism against data grubbing and the reliance of available data, data is a very strong tool in model building. Pidd considers it helpful to divide data and information into three groups.

   First is the preliminary or contextual data and information. This is the kind of data that answers the questions what, why, when, how, where and who, Kipling’s six honest serving-men (The Elephant’s Child, 1912, p. 83). The focus of this data and information collection lies on understanding more about the context of the problem, rather than gathering information used to develop a detailed model. This type of research might prove to be enough to decide whether or not to take a project any further.

   The second group of data and information is that which might be needed in order to develop the model in some detail. This type of data and information falls under the process of model parameterisation, or model realisation; the process of “determining how to make the model fit the data and how to get answers from the model” (Willemain, 1995, p. 921).

   The third group of data is discussed under the heading Avoid using the same data to build and to test the model.

3. **Beware of data provided on a plate:** Pidd writes that, “for modelling purposes, data is best ordered a la carte rather than table d’hôte” (1996, p. 725). By this statement he is referring to the fact that most companies today collect almost
every conceivable type of data about what is going on. This data may, however, be incomplete or only cover a specific aspect of the process.

Pidd illustrates this problem by using the process of modelling customer demand as an example. Most companies have a lot of data from their sales department and it might seem as a good idea to use this data. There are however some problems in doing that. Firstly, most companies only store data from actual deliveries to customers. If the product in demand is out of stock or otherwise removed from the list it might not be represented in the data. Secondly, if the customer assumes that the product in demand is unavailable, due misleading information or other miscommunications, this demand will most likely not be represented in the data. Finally, the entire reason for the research might be to increase sales in areas where customers normally go to other suppliers. According to the existing data the demand in this area will be considered low. In this case it would be wiser to order a new collection of data for the model; data that represent a more realistic picture of customer demand.

If pre collected data is presented it might be wise to conduct a small scale data collection in order to verify how representative the provided data is. Another way to handle presented data is to apply so called data massage. This is the process of conducting a small scale data collection and then mould the provided data to match this newly gathered data. It is however important to bear in mind that data massage implies using a model of the data itself as a part of the modelling process.

4. Data is just a sample: It is important to remember that data is just a sample and not an exact representation of the reality. Reality is complex and ever-changing. What data collection does is taking a sample of a given area in a given time frame. Pidd compactly describes this by saying, “if we say that data is believed to be representative then we are implying that it is representative of a larger population which displays some regularity through time” (Pidd, 1996, p. 725). Data runs the risk of changing through time and space. This is an important aspect to bear in mind, especially when extrapolating data into the future. Data is a set of observations and a sample which was gathered given limited time and resources; one can always gather more data.

5. Avoid using the same data to build and to test the model: Most models uses data in one form or another. Parameterisation is usually based on data. The third group of data, continuing the list under the heading Data is useful in model building, is validation data. This type of data is used to validate the model. A common mistake is to use the same data, as which was used during the parameterisation process, in the validation process. This method usually guarantee good results but that is not the only point of modelling, to get a good match with one set of data. It is important that the model remains valid for more than one set of data.

Principle 5. Model building may feel like muddling through: No matter how it might seem, modelling is not a linear nor highly rational process. There is no golden rule, no specific method that works for everyone. Instead Pidd describes the process as “people
seem to ‘muddle through’, making use of insights, perhaps taking time away from modelling, trying to look at things from different perspectives and so on” (1996, p. 726). This is, however, not how it works for everyone. People think differently and the fact that modelling is a tool for thought makes the process of modelling vary from person to person.

Pidd refers to an investigation made by Willemain (1994) where he gathered a group of twelve experienced modellers and asked them to describe themselves, their view on their own approaches to modelling and to describe important aspects of modelling, based on their own experience. Willemain (1994, p. 214) concluded that they mainly work in areas where they “pursue specific objectives towards fundamental changes in complex, existing systems”, and that they also “develop a unique model for each problem, though all their models involve extensive computation”. Their approach to model building is to “develop their models, not in one burst, but over an extended period of time marked by heavy client contact”, and they are “guided by analogies, drawing and doodling, they develop more than one alternative model, in each case starting small and adding”. The conclusions drawn here by Willemain are in line with the principles stated by Pidd. Pidd concludes by writing:

“Perhaps it is an exaggeration to say that modellers muddle through. However, it is equally an exaggeration to assert that modelling proceeds as a linear step-by-step process.” (Pidd, 1996, p. 727)

The principles stated by Pidd are not step-by-step instructions in how to develop a model but rather guidelines in how to handle complexity. When being listed like this they may seem obvious but they are easily forgotten in the mess that is a model development process. The reason these principles were chosen was to add some form of structure to the development process and to show that when complexity is being considered from start to end it is easier to handle than when remembered in the last stage.
Test case - Advanced Emergency Brake

As was stated in the delimitations section, one of the main focuses of the thesis was to investigate the AEB and use the AEB test case to verify the model. This chapter is dedicated to the AEB system in general and the structure of the AEB test case.

Advanced Emergency Brake

Advanced Emergency Brake, AEB, is the commercial name for the driver assisting system developed by Scania CV AB to reduce the risk of collision (Scania CV AB, 2016a). The AEB system is fully integrated into the HDV and uses a setup of long-distance radar technology and a forward-facing camera to assess potential collision danger. The sensor setup is illustrated in Figure 2. The radar system measures the distance to and the relative speed of and obstacles in front of the HDV while the camera determines the width, the lateral position and the nature of each obstacle. This information is then processed by a central processing unit along with additional data from the vehicle, such as the speedometer and the engine control system. A program is then able to make necessary judgements and act accordingly. The system is designed to filter out false targets as these annoyances might increase the risk of an accident rather than the prevention of it. It is important to note that AEB is a driver support system and not an autonomous system.

The system acts in three stages. Once the vehicle exceeds low speed velocity and an obstacle is detected in the path of the vehicle the AEB system assesses whether or not the driver is in control by examining the change of position of the brake and accelerator pedals. If the AEB
system senses that the driver is not reacting it activates a collision-warning signal and informs the driver via the driver’s display and by sound. A special brake-assist system is also activated, increasing the brake-pedal sensitivity.

If there is still no response from the driver the AEB system applies the service brakes and illuminates the brake lights. Only a portion of the vehicles total braking capacity is applied as a mean of attracting the driver’s attention. These actions are executed when the distance to the obstacle has roughly halved since the initial warning.

If the AEB system judges that a threshold based on relative speed and distance to the obstacle has been exceeded, and there is still no reaction from the driver, an emergency braking is commenced. Based on the situation there might be too late to avoid a collision, but the system continues to lower the speed of the vehicle in order to reduce the force of impact.

AEB is aborted if the obstacle diverts from the trajectory of the vehicle, if it accelerates away from the vehicle or if the system assesses that the driver is in control. The driver can ignore the warnings by either depressing the accelerator, which is done when overtaking another vehicle for example, activate the brakes or using the turn signal, indicating an intended turn from the trajectory. There is also a switch where the driver can deactivate the function completely.

Test cases in general

A test case in general is a set of conditions which are used to determine whether the features of an application or software system are working as originally intended. The mechanism determining whether a test was passed or failed are referred to as an oracle. An oracle can be anything from a requirement, a use case or a heuristic (requirements and use cases are more common than heuristics). The requirements can be anything from the product specification to an ISO standard. A use case often refers to a list of actions with known outcome. A heuristic is an approach to solving or learning about a problem so that enough information may be gathered to be able to employ a method sufficient for the immediate goals.

A test case in this sense is a written script that automatically executes different inputs to allow tests to be performed under equal conditions each time and allows automatization. It can be anything from just turning the key in the ignition to identifying an obstacle at a given distance and monitor the output of instruments.

Test cases are often designed to test and inspect a specific functionality in the system. The script is then designed to make sure that all the initial conditions required to test this specific functionality are met, then executes whatever manoeuvre needed to trigger the functionality being tested and finally monitoring the outcome, determining whether or not the test is passed and resetting the initial conditions so that a new test case may be run.

It is also possible to trigger several test scripts in a row in a so called test suite. The suite then runs multiple test cases in a row summarizing the information from each case in a final report. This feature is especially useful if one needs to run a batch of tests taking a long time overnight.
AEB functionality test cases

The AEB functionality test cases is the benchmark for the thesis. The exact layout of the test cases is classified but what they do is to check and make sure that all the functionality mentioned under the Advanced Emergency Brake section is working. This includes testing all the abort criterion and full emergency brake intervention without collision. This testing is done by having several obstacles appearing and disappearing, aborting the system with all the different triggers (avoiding, overtaking, manually braking, activating the turn signal and so on) and checking that the system is responsive enough to brake in time to avoid a collision. The general behaviour of the test case can be seen in Figure 14 and 15.
Chapter 4

The model

The model developing process was carried out with Pidd’s five simple principles of modelling in mind and what was known to be needed in form of functionality in order to run the AEB test cases. These two sources of information form the foundation upon which the model was built. The structure of the model was somewhat limited by the interface to which it was to be connected in order to run it in the HIL configuration. The modelling process was divided into external and internal factors and the common interface between these two are the tires, which serves as the vehicles connection to its surroundings. The external factors are referred to as the acting forces on the vehicle while the internal factors are the dynamics which take place within the vehicle powertrain. All these factors were then converted to system blocks in Simulink and combined to form the final model.

Acting forces

While driving the vehicle is affected both by external forces and forces originating from the vehicle itself. These forces acts in vertical, lateral and longitudinal directions. The most prominent vertical forces acting on the vehicle are gravity, downforce and normal force. The dominant lateral forces are those that appear during steering, from side wind and due to road tilt. Forces acting in these two directions are irrelevant for the thesis as they do not have any considerable impact on the acceleration and deceleration of the vehicle and the test case for AEB is carried out in a straight line. The thesis focuses on the forces that act in the third direction; the longitudinal forces.

The longitudinal forces that act on the HDV considered in the thesis are the propulsion force, road forces, brake forces, air resistance and gravitational forces due to road grade, all illustrated in Figure 3. The resulting force balance equation can be seen in (1). This equation was used to determine the velocity of the vehicle once all other factors were known. The basic dynamics and origin of the acting forces are discussed individually below.
Figure 3. Acting longitudinal forces on a HDV. The scale of the forces are just for illustration; this is not the actual proportion. Inspiration taken from (Jonhed, 2013).

\[ m\ddot{v} = F_{\text{propulsion}} - F_{\text{air}} - F_{\text{brake}} - F_{\text{road}} - F_{\text{gravity}} \] (1)

**Propulsion force**
The propulsion force is produced by the engine, transmitted through the drivetrain to the wheels and through the wheels to the road. The origin of the force is discussed in the engine section and the exchange between the different components in the drivetrain is discussed throughout the powertrain section. The force in relation to the torque transmitted to the tires is seen in (2).

\[ F_{\text{propulsion}} = \frac{T_{\text{tire}}}{r_{\text{tire}}} \] (2)

**Road forces**
Road forces in this case refers to those forces acting when the tires roll against the road. In the thesis these forces are limited to roll resistance which is expressed in (3), (Hibbeler, 2007, pp. 441-442). Roll resistance is a portion of the normal force acting in the longitudinal direction due to friction between the tire and the road. Roll resistance is, on horizontal ground, a constant force and the reason an object remains still until this force is surpassed and the object start to roll. It is not affected by the vehicle speed in contrary to the air resistance.

\[ F_{\text{road}} = F_{\text{roll}} = mgC_{rr} \cos \alpha_{\text{slope}} \] (3)

**Brake forces**
Brake force is the sum of all acting brake forces. These does not necessarily act on the tires directly, but as they are means of deaccelerating the vehicle, they can all be traced to the tires. A HDV make use of several braking systems which are all essential in the AEB test cases. These systems are: service brakes, retarder and exhaust brakes. All the braking systems act on different parts of the drivetrain and are therefore explained separately under the Brakes section.
Air resistance
The air flow acting on the vehicle in the longitudinal direction when driving is referred to as air resistance or drag force (Torenbeek & Wittenberg, 2009, p. 151). The force is expressed in (4) first as a general expression accounting for wind speed in the direction of driving, as having tailwind reduces the impact of air resistance. However, as accounting for wind speed is not a part of the AEB test cases and just adds to complexity, wind speed is assumed to be zero throughout the thesis.

\[ F_{\text{air}} = \frac{1}{2} \rho (v - v_0)^2 C_D A = \{v_0 = 0\} = \frac{1}{2} \rho v^2 C_D A \]  (4)

Road grade
On a non-horizontal road gravity is no longer limited to the vertical direction. The gravitational component that acts in the longitudinal direction is dependent on the road grade according to (5).

\[ F_{\text{gravity}} = mg \sin \alpha_{\text{slope}} \]  (5)

Powertrain
Powertrain refers to the group of components that generate power and deliver it to the wheels. In this case it consist of engine, starter, clutch, transmission and differential. The crankshaft here serves as a connection between the engine, the starter and the clutch while the gear train connects the transmission, differential and clutch. The clutch is the common factor serving as an interface between the crankshaft and the gear train. A system of brakes is also connected to the powertrain, reducing the amount of power reaching the wheels. The powertrain with the added brake system is illustrated in Figure 4.

![Figure 4. Layout model of the powertrain and brake system.](Image)
The components described below were modelled simple and the complexity comes in the form of data tables, block integration and the final size of the model. Complex dynamics were replaced by a measured function curve in accordance with Pidd’s first principle, Model simple, think complicated. This gain in simplicity comes at a loss in flexibility. The loss in flexibility is reduced by normalizing the curves and multiplying with the specific component’s peak value. As long as the general behaviour of each alternate component is similar and just varies in maximum performance this method allows for some flexibility. If the alternate component varies too much it is even possible to overwrite the entire data table when running the model with that from a different version of the component.

**Engine**

The engine serves as the source of power for the powertrain. The amount of torque delivered is limited by the available torque based on the current engine speed and parasitic losses according to (6). The behaviour of the limiting torque is seen in Figure 18 and the corresponding parasitic losses are seen in Figure 19, both located in Appendix A. These curves vary somewhat from engine to engine. The most important factor in this case is the peak value of Figure 18, as this is what limits the source of power for the entire system.

\[
T_{eng} = \min( T_{maxeng}(\omega_{eng}) - T_{engloss}(\omega_{eng}, T_{req}) - T_{exh} ) \tag{6} 
\]

**Starter**

The starter is mounted on the engine and is what cranks the engine from its initial zero value into idle speed. When on, it is modelled to deliver a peak torque of \( T_{maxstart} \) limited by the current engine speed and the constant maximum speed allowed by the starter according to (7). The torque influence from the starter is approximated to be zero when the starter is considered off. The on/off states are controlled by the ignition key. \( T_{maxstart} \) and \( \omega_{maxstart} \) are both constant values changing from engine to engine.

\[
T_{start} = \begin{cases} 
T_{maxstart}\left(1 - \frac{\omega_{eng}}{\omega_{maxstart}}\right), & \text{on} \\
0, & \text{off}
\end{cases} \tag{7}
\]

The inertia of the starter is later used to calculate the engine speed in (9) and is approximated to be constant when on and zero when off according to (8).

\[
J_{start} = \begin{cases} 
\text{const}, & \text{on} \\
0, & \text{off}
\end{cases} \tag{8}
\]

**Crankshaft**

The crankshaft here refers to the shaft going through the engine and connects it to the clutch. It is what converts the reciprocating motion from the pistons into rotational motion. In order to reduce pulsation characteristics from the strokes of the pistons and to maintain rotation it is also connected to a flywheel. The flywheel is responsible for the majority of the constant rotational inertia, \( J_{crank} \), of the crankshaft. The angular velocity of the crankshaft is calculated by
extracting the angular acceleration from Newton’s second law for rotation and integrating the expression, which results in (9).

$$\omega_{\text{eng}} = \int \frac{\sum T}{\sum J} \, dt = \int \frac{T_{\text{eng}} + T_{\text{start}} - T_{\text{fric}}}{J_{\text{crank}} + J_{\text{start}}} \, dt$$

(9)

**Clutch**

The clutch serves as an interface connecting the crankshaft and the gear train. Its primary purpose is to match the transmission speed, $\omega_{\text{trans}}$, with the engine speed, $\omega_{\text{eng}}$. This is done by taking the mean of the sum of the torque from both the crankshaft and the gear train and accounting for damping within the clutch due to difference in speed. The resulting torque is seen in (10) and is referred to as friction torque, $T_{\text{fric}}$, and is fed back to both the crankshaft and the gear train.

$$T_{\text{fric}} = \frac{T_{\text{eng}} + T_{\text{start}} + T_{\text{trans}}}{2} + K_{\text{damp}}(\omega_{\text{eng}} - \omega_{\text{trans}})$$

(10)

The clutch input, $u_{\text{clutch}}$, is handled as a dynamic Simulink saturation block (MathWorks, 2016a), limiting the friction torque by a factor of $T_{\text{maxclutch}}$ according to (11). When the clutch pedal is fully pressed $u_{\text{clutch}} = 0$ which results in $0 \leq T_{\text{fric}} \leq 0 \rightarrow T_{\text{fric}} = 0$. The clutch discs are fully separated and no torque is transferred.

$$\|T_{\text{fric}}\| \leq u_{\text{clutch}} T_{\text{maxclutch}}, \quad u_{\text{clutch}} \in [0,1]$$

(11)

**Transmission**

The transmission is responsible for providing controlled speed and torque conversions from the engine to the differential, which in turn drives the wheels. In this case the transmission is limited to the gearbox and refers to the exchange taking place in said component. The gearbox modelled in in the thesis is a six-speed automatic gearbox, meaning that it has six different gears with positive ratio, one reverse gear ratio and one neutral gear giving zero exchange, i.e. separated. The different ratios are shown in Figure 20 starting with reverse, neutral and then the six speeds. The torque and speed exchange follow (12) and (13), which means that a higher ratio gives a greater torque exchange and a lower transmitted speed to the differential.

$$T_{\text{diff}} = T_{\text{trans}i_{\text{gear}}}$$

(12)

$$\omega_{\text{diff}} = \frac{\omega_{\text{trans}}}{l_{\text{gear}}}$$

(13)

**Gear train**

The gear train refers to all the shafts and otherwise rotating components following the clutch leading up to the wheels. In this setup this includes all the components in Figure 4 except the engine and the starter. The angular velocity of the gear train is calculated by extracting the angular acceleration from Newton’s second law for rotation and integrating the expression, which results in (14).
\[
\omega_{\text{trans}} = \int \frac{T_{\text{fric}} - T_{\text{diff}} - T_{\text{brake}} - T_{\text{gear}}}{J_{\text{drive}} + J_{\text{gear\_out}} + J_{\text{vehicle}} + J_{\text{gear\_in}} + J_{\text{clutch}}} \, dt \quad (14)
\]

\(J_{\text{vehicle}}\) here refers to the rotational inertia acting on the wheels from being in contact with the ground. This rotational inertia is the dominating source of inertia acting on the gear train and is responsible for more than 99% of the total rotational inertia. Depending on where on the gear train different torques and rotational inertias act, they are multiplied with the corresponding gear ratios. The gear ratios are dependent on where in the gear train they are used. In the thesis model the torque balance is taking place right after the transmission which results in \(J_{\text{vehicle}}\) having to pass through the differential, resulting in the expression seen in (15).

\[
J_{\text{vehicle}} = \frac{mr_{\text{tire}}^2}{i_{\text{diff}}^2} \quad (15)
\]

**Differential**

The main purpose of the differential is to allow the outer wheel to rotate faster than the inner wheel during a turn. This feature is however not important for the thesis as the individual wheel speed is not accounted for and the model instead calculates and average speed for both wheels, i.e. \(\omega_{\text{drive}}\). The important feature of this component is the exchange in torque and speed this coupling provides, \(i_{\text{diff}}\). By increasing the torque according to (16), and lowering the speed according to (17), the differential provides the torque required to accelerate the vehicle. A typical value for the exchange ratio, \(i_{\text{diff}}\), is somewhere between 2 and 4.

\[
T_{\text{diff}} = \frac{T_{\text{tire}}}{i_{\text{diff}}} \quad (16)
\]

\[
\omega_{\text{drive}} = \frac{\omega_{\text{diff}}}{i_{\text{diff}}} \quad (17)
\]

**Brakes**

The HDV modelled in the thesis has three mechanical sources providing brake torque to the system. These sources are the service brakes, the retarder and the exhaust brake and are explained further below.

*Service brakes*

Service brakes are the brakes most commonly referred to when talking about vehicle brakes. The HDV uses hydraulic brakes pressing on discs mounted behind the wheel. The theoretical function of these brakes are expressed in (18). A brake force, \(F_{\text{brakereq}}\), is requested based on the position of the brake pedal. This force is delivered until an upper limit is reached based on the friction coefficient of the brake, \(\mu_{\text{brake}}\), and the normal force, \(mg \cos \alpha_{\text{slope}}\). However, as the brakes are an important factor of these tests, the resulting torque goes through a black box block accounting for ABS efficiency, hydraulic pressure and other measured factors and deliver the *actual brake torque*. This black box block is provided by Scania CV AB.
\[ T_{\text{brake}} = \min\left( mg \mu_{\text{brake}} \cos \alpha_{\text{slope}}, F_{\text{brakereq}} \right) r_{\text{tire}} \]  \hspace{1cm} (18)

**Retarder**

The primary purpose of the retarder is to assist the service brakes at higher speeds. The retarder slows the rotational speed of the propeller shaft, the shaft exiting the transmission, with the use of viscous drag forces. It consists of a rotating disc mounted to the propeller shaft with vanes rotating in a fluid-filled chamber with static vanes. The pressure in this chamber is then increased and the distance between the static and dynamic vanes reduced in order to increase the viscous drag in the chamber and achieve retardation. This type of braking is highly dependent on the speed of the propeller shaft and is much more efficient at higher speeds when the fluid exchange is greater. The main use of the retarder is therefore to slow the vehicle at higher speeds and help maintain constant velocity when driving downhill. The amount of braking torque generated by the retarder is a highly complex process to model. Test data (Li, Yang, & Zhang, 2015, pp. 270-271) is therefore used in the thesis for both the retarder torque, \( T_{\text{retmap}} \), and the torque loss, \( T_{\text{retloss}} \). The general behaviour of these torque functions are seen in Figures 21 and 22 in Appendix A. The total torque generated by the retarder is expressed in (19).

\[ T_{\text{ret}} = T_{\text{retmap}}(u_{\text{ret}}, \omega_{\text{trans}}) - T_{\text{retloss}}(\omega_{\text{trans}}) \]  \hspace{1cm} (19)

**Exhaust brake**

The exhaust brake works by closing off the exhaust path from engine and thereby causing the exhaust gases to remain in the cylinder. No additional fuel is applied and the exhaust gases keeps compressing in the cylinders causing the engine to work backwards, slowing the crankshaft and thereby the vehicle. The amount of brake torque generated by the exhaust brake is proportional to the back pressure of the engine, forming an upper limit. The behaviour of this limit is seen in Figure 23. The torque delivered by the exhaust brake is modelled as the requested input, \( u_{\text{exhreq}} \), multiplied with a gain, \( K_{\text{exh}} \), scaling the requested input to the maximum brake torque. The requested torque is then limited by the upper limit, \( T_{\text{maxexh}} \), based on the engine speed, and the final torque is expressed in (20).

\[ T_{\text{exh}} = \min\left( K_{\text{exh}}u_{\text{exhreq}}, T_{\text{maxexh}}(\omega_{\text{eng}}) \right) \]  \hspace{1cm} (20)

**Simulink subsystem blocks**

“Simulink is a block diagram environment for multidomain simulation and Model-Based Design. It supports simulation, automatic code generation, and continuous test and verification of embedded systems.” (MathWorks, 2016b)

The idea is to convert the dynamics and components listed in the acting forces and powertrain section into subsystem blocks in Simulink. This idea is the embodiment of Pidd’s third principle, Divide and conquer, avoid mega-models, by dividing the components into minor subsystems and later combining these subsystems into one large system.
The conversion strategy from mathematical formula to a block diagram used in the thesis may be considered a contra intuitive process, starting by identifying the outputs and trace it back to the inputs. In the case of the service brakes, expressed in (18), there is one output, $T_{brake}$, and three time varying inputs; $F_{brakereq}$, $\mu_{brake}$ and $\alpha_{slope}$. One might argue that the mass might vary with time as the vehicle consumes fuel, but the test cases are too short and the total mass of the vehicle too large for this to be of any noticeable impact; and in terms with Pidd’s second principle, Be parsimonious, start small and add, it is better to avoid unnecessary complexity at first and then add it if needed. The relation between the output and the inputs may be modelled as illustrated in Figure 5.

![Figure 5. Simulink block diagram example of the service brake. This is just an example of how this component may be modelled in Simulink, not the actual setup used in the model developed in the thesis.](image)

The block diagram illustrated in Figure 5 form a subsystem block with the corresponding inputs and outputs, seen in Figure 6, which in turn can be connected to other. This system block conversion was done for each component.

There are some general guidelines provided by MathWorks to keep in mind when developing a vehicle simulation model. These guidelines are referred to as MathWorks Automotive Advisory Board, MAAB (MathWorks, 2016c). The MAAB is a 136 page document covering everything related to automotive vehicle modelling, from guidelines regarding layout (i.e. to avoid crossing signals and keeping inputs to the far left and outputs to the far right) to naming parameters and generating code. Luckily there is an automatic check tool included in Simulink called Simulink Verification and Validation. It automatically compares the selected model to the MAAB and lists and/or executes recommended changes.

Model structure

Each component and dynamic listed under the acting forces and powertrain section were converted to subsystem blocks following the strategy stated in the Simulink subsystem blocks section. These subsystem blocks were then connected to form the dynamic structure of the layout model illustrated in Figure 4. The resulting block model system structure is seen in Figure 7. Each subsystem block is dependent on at least one other subsystem block forming a network of dependence. This dependency is a good example of Pidd’s fifth principle, Model
building may feel like muddling through. Modelling is not always a linear process and there is not necessarily a clear starting point.

Figure 7. Block illustration of the model. The braking subsystems blocks were excluded from this illustration and were instead represented as an input torque to the gear train subsystem. The actual model is more complex, but this illustration covers the basics.

Model and hardware integration

The model created in the model structure section is still dependent on a lot of external inputs in order to function properly. These inputs can be anything from the clutch pedal position to communication with the window mounted radar. The input signals originate from each component, each lever and each button and are fed to different ECUs throughout the vehicle depending on which subsystem the signal is a part of. Together these ECUs monitors and control everything in the vehicle and form the vehicle’s graphical user interface, GUI. In the thesis only one real ECU is integrated and the rest is being simulated. This ECU is the one that handles everything related to AEB and some additional functionality.

It is therefore important that the model is connected to the ECU interface in order to receive these signals and inputs. This integration was done in the model with the use of controller area network buses, referred to as CAN buses, I/O pins and Simulink GOTO blocks (MathWorks, 2016d). CAN buses are responsible for transferring the information to and from the model to the ECU and the hardware I/O, i.e. the HIL communication. The information being transferred on the CAN buses are ranked based on importance and high priority information is ranked higher and prioritised. The priority ranking is structured around what information is most crucial to maintain functionality and guarantee the safety of the driver and passengers.
The information transferred on the CAN buses are then divided into Simulink GOTO blocks in the *To MDL* block illustrated in Figure 8. This block serves as the bridge between the model and the hardware. GOTO blocks functions as such that they makes a signal accessible globally in the model by giving it a variable name and a path. This variable name and path can then be called upon wherever needed throughout the model.

The *From MDL* block illustrated in Figure 8 is the exact opposite of the *To MDL* block. It collects all the GOTO block information being sent from the model and converts it back to CAN signals. Together, the *To MDL* and *From MDL* blocks serve as the interface between the model developed in the thesis and the hardware it was tested in. A major part of the thesis has been identifying what signals were needed, finding their variable name and their path and mapping them into the model. Pidd’s fourth principle, Do not fall in love with data, has been in the back of the mind throughout this process, finding what data is available and tracking its roots in order to validate the source.

![Figure 8. Illustration of model, ECU and hardware integration.](image)

The hardware integration part of the model is the interface the model developed in the thesis is connected to and was not developed in the theses.
Chapter 5

Hardware-in-the-loop

All tests related to the thesis were done in real-time with partially real hardware. This type of simulation is referred to as Hardware-in-the-loop, HIL, simulation. The point of a HIL simulation is to add the complexity of the plant under control to the test platform. All electrical signals are emulated between the plant and the embedded system under test by the HIL rig. In the model this conversion was handled by the Hardware I/O block illustrated in Figure 8, converting signals to and from I/O pins.

The idea is to test how the controller responds in real-time to realistic virtual stimuli. The HIL configuration is setup as illustrated in Figure 9. The controller in this case is the ECU being tested and the plant model, yellow in Figure 9, is the HDV model developed in the thesis. This setup allows for real hardware integration and a level of testing which otherwise would require full-scale testing in an actual HDV. The HIL rig converts modelled signals into real I/O signals which are sent to and processed by the ECU. The ECU then responds to this information as it would in a real vehicle and acts accordingly. This response information is then sent back and converted from I/O signals into signals processed by the computer and model.

Some preparations have to be made to a model before it is ready for HIL simulation. The preparation process is best described as a workflow consisting of three stages; real-time model
preparation, real-time model simulation and hardware-in-the-loop simulation, all illustrated in Figures 10, 11 and 12.

Real-time model preparation

The first stage is to prepare the model for real-time simulation. This preparation is done by following the workflow illustrated in Figure 10 and starts with obtaining reference results and validating the model; making sure that it is a valid representation and matches reality. Validation was an essential part of the thesis and is covered in chapter 6.

Once a valid model is created that works well in a fix-time frame it is important to evaluate the overrun risk. This refers to the risk of having a too long computational time to cope with real-time simulation, resulting in the system getting overrun and losing accuracy and validity. Simulink has built in tools that evaluate and warn of any overrun risk.

If there is an overrun risk the next step is to adjust the model fidelity or the scope. This refers to improving the simulation speed by for example reducing numerical stiffness, reducing zero crossings, reducing computational cost or partitioning the model for parallel processing. Another approach is to evaluate the results obtained with a variable-step solver. This information is used to find the maximum step size available to increase accuracy or exact times when the simulation is slowed down.

Once the model accuracy is evaluated and the results are comparable to the reference results, i.e. the model is considered valid, the next step is to perform the real-time model simulation workflow stage.
Real-time model simulation

The second stage when preparing a model for hardware-in-the-loop simulation follows the workflow illustrated in Figure 11 and starts by performing a fixed-step, fixed-cost simulation. This refers to running a simulation at a fixed computational cost. Increasing the number of steps increases the accuracy but also the computational cost. Lowering the number of steps lowers the computational cost but may also lead to loss of validity. Finding a balance while still maintaining validity might prove being difficult and is hopefully achieved by adjusting solver settings. In some cases it is required to return to the real-time model preparation workflow for additional fine-tuning.

Once again it is important to evaluate the overrun risk as adjusting the step-size most likely had an impact on the computational time. If Simulink evaluates the risk being too high the simulation speed may again be improved by adjusting the solver settings. If adjusting the solver settings turns out to not be enough there may again be need to return to the real-time model preparation workflow.

If everything looks fine the model is real-time viable and the next step is to perform the hardware-in-the-loop simulation workflow stage.
Hardware-in-the-loop simulation

The final stage when getting a model ready for hardware-in-the-loop simulation is to test the model and evaluate its robustness. The testing and evaluation is done according to the workflow illustrated in Figure 12 and starts by generating and compiling the code on the development hardware, seen in Figure 9, downloading the real-time application on the target hardware and executing the real-time application remotely from the development computer using a software called ControlDesk (dSPACE, 2016a).

The accuracy of the model may once again be evaluated to see that it matches the reference data. If this is not the case there may again be need to return to the real-time model simulation workflow for additional fine-tuning.

If the accuracy is sufficient all that is left is to check for any overrun risk. If Simulink evaluates the risk being too high the simulation speed may again be improved by returning to the real-time model simulation workflow.
Once the accuracy is considered sufficient and there is no real overrun risk, the model is hardware-in-the-loop viable.

Figure 12. Hardware-in-the-loop simulation flowchart. Image taken from (MathWorks, 2016h).
Chapter 6

Validation

Model verification and parameterization

Component wise parameterization and verification
The model was developed component wise, meaning that each component was developed, parameterized and tested individually to the extent possible before being integrated into the complete model. This means that the parameterization and verification process was partly integrated into the development process, once again proving the nonlinear work process described in Pidd’s fifth principle, Model building may feel like muddling through.

Parameterizing and verifying the different blocks separately requires construction of realistic input signals. To some extent these signals were already available. One huge advantage of working with an actual vehicle model is that most parameters are set beforehand by the specification of each component being modelled. For example, when modelling the engine, parameter values such as maximum deliverable torque, parasitic losses, rotational inertia and so on are already available in the Scania CV AB database for that engine. The difficulty comes in understanding the dynamics of the component and putting everything together.

Some input signals could not be constructed to the level of detail required in order to parameterize and verify all component blocks individually. These component blocks had to be integrated into the model before being tested, relying on previously gathered data and rough estimations. Once the components were in place each incoming and outgoing signal could be monitored and compared to realistic test data and the component could be parameterized such that the simulated data matched the reference data.

Reference data
The amount of reference data available for the thesis at Scania CV AB is close to endless. With today’s technology with ECUs monitoring most of the vehicle data a lot of information can be stored from the field testing for later use. As Scania CV AB develops most of vehicle components themselves, a lot of benchmarking data and other relevant data is also available.

There are many difficulties in choosing good reference data when developing a product. The guidelines in Pidd’s fourth principle, Do not fall in love with data, state that it is important to not parameterize and validate the model with the same set of data. The parameterization and verification process were restricted to reference data from the previous simulation model gathered alongside the development of the thesis model. The reason for using this data was the fact that full access to the previous model was given, allowing for the ability to trace all reference data back to its source. By having access to the previous model also allowed for gathering new data whenever needed. Specific data from a certain component or signal could easily be gathered at any point. By having the previous model as reference it was simple enough to guarantee that all data was as up-to-date as it could be. The data gathering process could also
be restricted to the component being parameterized at the given moment reducing the amount of unnecessary data being collected.

When working in a HIL environment it is important to know if the data gathered originates from the model or the ECU. The data gathered from the ECU has gone through the to hardware conversion and may not be as reliable as data gathered from the model directly. When parameterizing it is easier to use the model data as this type of data better matches the model. When validating the model it is important that the data being used it gathered from the CAN buses as this is the data logged when running HIL tests. This division in source guarantees that the data used to parameterize the model was not the same data as was being used to validate the model in its entirety.

As described in the Hardware-in-the-loop simulation section the model was ran in a software called ControlDesk (dSPACE, 2016a). The interface of ControlDesk, seen in Figure 13, allows for monitoring any signal throughout the model, both model signals and CAN signals. It also allows for easy swapping between the original model and the model developed in the thesis for quick comparison. This tool has been essential for tracking down errors by tracing them back to the source.

![Figure 13. ControlDesk interface example. Image taken from (dSPACE, 2016b).](image)

**Full-scale simulation**

Once all components had been parameterized, tested and combined to form the final model structure seen in Figure 7, the full-scale verification process was started. This refers to running the AEB test cases for both the previous model and for the new model developed in the thesis. Running the test case was done by loading the model being tested into the ControlDesk (dSPACE, 2016a) software and running the test script calling the selected test cases. A program called CANalyzer (Vector, 2016) then logs all CAN data for later use. The relevant CAN signals are then selected and stored in .csv files. A MATLAB script loads the .csv files for both models
and lines up the starting point of the simulations for a good comparison. The results are seen in the simulated results section and discussed under discussion section.

**Additional testing**
The model was also exposed to an overnight multiple test suite session testing all the rig tested functionality covered by the REVT department where the thesis was carried out. When the suite is completed a report is structured with pass/fail checks and corresponding information for each test case and condition.

**Simulated results**
The first full-scale simulations were done as described in the full-scale simulation section. From all the gathered data the vehicle speed, distance to obstacle and engine speed were decided to be the most relevant. The resulting comparison between the original model (blue) and the model developed in the thesis (red) can be seen in Figures 14, 15 and 16.

![Vehicle speed graph](image)

*Figure 14. Shows the difference in vehicle speed between both models. Original model (blue) refers to the model used by the department today while project model (red) refers to the model developed in the thesis. Axis values are removed to protect the integrity of the function.*
Figure 15. Shows the difference in how the distance to the obstacle varies between both models. Original model (blue) refers to the model used by the department today while project model (red) refers to the model developed in the thesis. Axis values are removed to protect the integrity of the function.

Figure 16. Shows the difference in engine speed between both models. Original model (blue) refers to the model used by the department today while project model (red) refers to the model developed in the thesis. Axis values are removed to protect the integrity of the function.

From the additional testing 97% of the conditions were passed. Among the failed conditions 85% were due to missed target speeds, 7.5% were due to expected collision warnings not being
Chapter 6

triggered properly and 7.5% were due to non-matching coolant levels. Apart these failed conditions there was an error due to a missing pathway. This missing pathway is due to the change in model block structure and has to be changed in the test script search way rather than in the model.

Error analysis

The average derivative of each obstacle case seen in Figure 15 was approximated according to (21), where \( N \) refers to the number of time steps for each obstacle case.

\[
\frac{\Delta f(t)}{\Delta t} = v = \frac{\sum_{i=1}^{N} f(t+1) - f(t)}{N}
\] (21)

The relative error of the average derivative of each obstacle case seen in Figure 15 was calculated according to (22) and presented in Table 1.

\[
\eta = \left| \frac{v_{\text{original}} - v_{\text{project}}}{v_{\text{original}}} \right|
\] (22)

In (22) \( v_{\text{original}} \) and \( v_{\text{project}} \) refers to the average derivative of each obstacle case for both the original model and the model developed in the thesis.

<table>
<thead>
<tr>
<th>Obstacle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \eta )</td>
<td>0.0737</td>
<td>0.0189</td>
<td>0.0202</td>
<td>0.0069</td>
<td>0.0077</td>
<td>0.0127</td>
<td>0.0303</td>
</tr>
<tr>
<td>Obstacle</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>( \eta )</td>
<td>0.0278</td>
<td>0.0388</td>
<td>0.0075</td>
<td>0.0290</td>
<td>0.0154</td>
<td>0.0262</td>
<td>0.0298</td>
</tr>
</tbody>
</table>

Discussion

The important characteristics from the results presented in the simulated results section is not how overlapping the curves are but rather that the general behaviour of both models are similar and the gradient of all curves are matching. A slight shift between both curves just indicate that an obstacle was generated faster in one case compared to the other. This kind of shifting behaviour is expected. The gradient however indicates how fast the model accelerates and decelerates. As the test case is using extreme conditions, i.e. full gas or no gas, and cruise control make sure that the initial velocity of each obstacle appearance is the same by a factor of \( \pm 1 \text{ km/h} \), a difference in deceleration would indicate loss in validity of the new model. This is most clear in Figure 15 as this figure only accounts for the deceleration parts of the test case.

According to Table 1 there is a maximum relative error of 7.4% in the first obstacle case while most other cases have a much smaller relative error. The reason for this larger error can be one of many. The obstacle appearing has a constant low speed velocity, resulting in the obstacle...
accelerating away from the HDV when the HDV has decelerated to a velocity lower than that of the obstacle. This relation is much clearer in Figure 17 where the first obstacle case is in focus and the shift in time of the obstacle appearance for both models has been adjusted for. The reason this increase in distance is considered a source of error is the fact that the obstacle is not removed directly after the HDV reaches a standstill.

In order to observe the relative error of the derivative of the distance to the obstacle without being influenced by the appearance and disappearance of the obstacle the derivate was also calculated when removing the first and last hundred samples of each obstacle case and focusing on the actual slope of the curve. The resulting relative errors, seen in Table 2, apart from case 5 and 13, show some signs of improvement. The relative error is much smaller in some cases and even rounded to zero in case 10 (actual calculated value were $1.08 \cdot 10^{-5}$).

When analysing the failed additional testing further it becomes clear that missed target speeds is always within 1 km/h from the target speed. I.e. if the target speed is expected to be within the region of 49-51 km/h the actual speed at the time of the target was 51.25 km/h. After some discussion with my supervisor at Scania CV AB, this behaviour is to be expected as some of the test cases has a very narrow region of tolerance. The actual speed is acceptable and the error is to be ignored. The reason for this deviation is most likely due to having real equipment rather than faulty modelling. The reason for the non-matching coolant levels is because the cooling system was removed and replaced by a torque loss. The coolant level can therefore not be measured and this triggers the error. Further inspection of the errors due to the non-triggering collision warnings were traced back to non-matching target speeds (actually speed relative to the obstacle). It also became clear that the collision warnings not being triggered were actually not collision warnings being activated but collision warnings supposed to be deactivated due to the obstacle being removed. From a point of safety this error is not of any major concern.

![Figure 17. Detailed view of the first obstacle in Figure 15 adjusted for time-shift. Original model (blue) refers to the model used by the department today while project model (red) refers to the model developed in the thesis. Axis values are removed to protect the integrity of the function.](image)
Table 2. Relative error for each appearing obstacle avoidance.

<table>
<thead>
<tr>
<th>Obstacle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>η</td>
<td>0.0675</td>
<td>0.0410</td>
<td>0.0004</td>
<td>0.0072</td>
<td>0.0289</td>
<td>0.0145</td>
<td>0.0210</td>
</tr>
<tr>
<td>Obstacle</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>η</td>
<td>0.0071</td>
<td>0.0191</td>
<td>0.0110</td>
<td>0.0130</td>
<td>0.0456</td>
<td>0.0153</td>
<td></td>
</tr>
</tbody>
</table>

Another source of error is the fact that the data sample is discrete. The approach used to tackle the discrete behaviour of the signal was to use a larger step-size when calculating the derivative to avoid averaging over these flat areas. As the data consisted of over a thousand samples for each obstacle case this method was considered to still give a good estimation of the actual derivative.

The model developed in the thesis also handles parasitic losses in the engine differently than the original model. It uses an approach where the losses are known for any given engine speed. While this method result in the same final torque being delivered from the engine it is a faster method than accounting for the actual inertias. This behaviour is clearly seen in Figure 16 where the thesis model tend to reach the low engine speeds faster and in general has narrower peak values.

The HIL rig also adds another dimension of complexity. It is no longer just parameters in a computer being evaluated but also real equipment with real electrical circuits. This hardware integration is a source of uncertainty as this is actually a platform to test the hardware rather than testing the model. The ECU is the actual SUT but is here used to verify the model. The hardware also adds small deviations in each input due to the fact that each signal is not an exact value but rather a tolerated range of values around the expected value.

On a final note the model developed in the thesis is built on a branch of the original model. The original model is in constant development and though most changes are included in the new model not everything is. Minor tweaks were made to the original model during the two month development process. The change-list of the original model was reviewed before comparing the two but as the very nature of the model has changed not everything could be implemented.
Chapter 7

Conclusion and future work

Conclusion

The five principles stated by Pidd has remained in the back of the mind throughout the development of the model. Some examples of situations where each principle have been used are the following:

- The first principle was the main approach when modelling all the components. The idea was to take the complex dynamics of each component and handle them in a form where the complexity remains but the model becomes simple. This is where the use of function curves listed in Appendix A has come in handy. By normalizing these curves and simply multiplying with the peak value from the specific component being used flexibility was maintained while still using a simple approach.

- The second principle is more of a tool of approach rather than a concrete feature. The fact that components that existed in the previous model could be removed is proof that this principle has been included.

- The third principle is included in the structure of using Simulink blocks. It is hard to prove the extent of the use of this principle without going through the entire model but the fact that it has been used is clear simply by observing the block illustration in Figure 7.

- The use of the fourth principle is clear in the verification process. Instead of using previously generated data as a reference both the new model and the previous model has been run back to back throughout the entire parameterization and verification process. The source of the data is therefore surely known, which adds a higher level of trust to the results.

- The fifth principle is more of a statement rather than a guideline to simply help the modeller come to term with the fact that modelling is not a linear process but rather a ball of yarn slowly untangling itself when being approached from different angles.

From the Simulated results and Discussion section it is clear that the resulting model is comparable to the previous model, with an average error of 2%, even though many simplifications were made. The main goal was to develop a less complex model that remained valid throughout the test cases. But how does one quantify complexity? Especially in this case, where the only way to run the model is in real-time, meaning that no runtime comparisons can be made. In order to argue that the model indeed is less complex than the original model a block structure illustration of the previous model was made and is seen in Figure 24 in Appendix B. By comparing Figure 24 and 7, illustrating the newly developed model’s block structure, it becomes quite clear that simplifications were made. To start with the entire cooling system has been removed as the impact from it, apart from the torque loss it provided, was insignificant. The transmission system went through a complete overhaul and the block structure in general is more in line with Pidd’s third principle. Instead of using large chunks it is divided into smaller components included in the subsystem it belongs to.
The additional test suite proves the model’s validity even beyond the frames of the AEB test cases. The reason the AEB was chosen as reference was the fact that it requires a lot of functionality from the model, resulting in enough functionality to cover the additional test cases as well.

All and all the new model met the deliveries and due to its less complex structure and maintained validity it was implemented at the REVT department at Scania CV AB.

Future work

This work has compared using a less complex Simulink model during HIL test cases compared to the more complex model used previously. Due to limited time the developed model was bound by the integration between model and hardware. An area of improvement is to also look into the model to hardware integration where no improvements has been made in the thesis. This part of the HIL configuration is highly complex and is bound to have areas where simplifications can be made.

The next step is to run the model in Software-in-the-Loop, SIL, simulation. The intended use of SIL is to simulate the ECU and focus the tests on the ECU software. Hardware is removed from the process which means that real-time is no longer a requirement. When the demand for real-time is removed simplicity really starts to pay off. By simplifying a model runtime can be reduced, resulting in more tests in a given timeframe. With the use of SIL it is easier to evaluate simplicity in saved time. This evaluation allows for a better comparison between the previous model and the model developed in the thesis.

A final approach might be to abandon the usual Simulink platform and instead look into physical modelling in Simscape (MathWorks, 2016i). Physical modelling allows the creation of physical component blocks with a more comprehensive layout. Instead of linking all the required parameters between two blocks, the physical model platform allows to simply connect the engine and transmission, as an example, with a shaft and all the information needed will be transferred through this connection. It covers more than 10 physical domains so integration between electrical, hydraulic and mechanical schematics are much clearer.
Appendix A

Component data

This appendix contains the plots for the component data used in the powertrain section in chapter 4.

Figure 18. Maximum available engine torque as a function of engine speed. The curve is based on measured data provided by Scania CV AB. Axis values are removed to protect the integrity of the model.
Figure 19. Engine parasitic losses as a function of engine speed. The curves are based on measured data provided by Scania CV AB (Rideskär, 2009, pp. 10-16). Axis values are removed to protect the integrity of the model.

Figure 20. Gear exchange ratio as a function of gear input. The curve is based on measured data provided by Scania CV AB. Axis values are removed to protect the integrity of the model.
Figure 21. Brake torque generated by the retarder as a function of lever input and engine speed. The surface is based on test data taken from (Li, Yang, & Zhang, 2015, pp. 270-271). Axis values are removed to protect the integrity of the model.

Figure 22. Torque loss generated by the retarder as a function of engine speed. The curve is based on test data taken from (Li, Yang, & Zhang, 2015, pp. 270-271). Axis values are removed to protect the integrity of the model.
Figure 23. Exhaust brake torque as a function of engine speed. The curve is based on measured data provided by Scania CV AB. Axis values are removed to protect the integrity of the model.
Appendix B

Previous model illustration

This appendix contains the block structure illustration used for comparison in the conclusion section in chapter 7.

Figure 24. Block structure illustration of the previous model. This illustration is just for comparison to indicate what simplifications have been made in the thesis. The green blocks remained, the yellow blocks were simplified or otherwise included in the green blocks and the red blocks were removed. The actual model is more complex, but this illustration covers the basics.
Bibliography


Dahan, U. (2012, March 5). *Don't try to model the real world, it doesn't exist*. Retrieved from The Software Simplist: http://udidahan.com/2012/03/05/dont-try-to-model-the-real-world-it-doesnt-exist/


