Development of a Simulation Model of an Automatic Gearbox

Ludvig Wendelius

Master's Thesis

Master of Science in Engineering Physics

Supervisor:	Afram Kourie
Examiner:	Martin Servin

August 19, 2012

Umeå University
Faculty of Science and Technology
Department of Physics
SE-901 87 UMEÅ
SWEDEN

Abstract

A simulation model for an automatic gearbox with primary retarder has been constructed and implemented, in this thesis. Together with other modelled vehicle components, this model could for example be used for fuel consumption estimation or optimizing vehicle parameters.

The mechanical components and the control system inside the automatic gearbox were modelled separately and then assembled into the final gearbox model, using the object-oriented programming language Modelica. Modelica ensures that each individual component can be reused in other models.

The gearbox model was validated through a number of test cycles designed to capture different vehicle behaviours. The test cycles were recreated in the simulation environment and the simulation results could be compared to a real vehicle performing the same tests.

Validation showed that the model succeeded in its goal, that the implemented model is reproducing similar behaviour as the real gearbox. With gear shifts taking place in about the same situations and converter locking/unlocking occurring the same time in the simulations as in the real vehicle testing.

Keywords: Drivetrain, Simulation, Modelica, Dymola, Automatic Gearbox, Torque Converter, Primary Retarder.

Sammanfattning

I det här examensarbetet har en simuleringsmodell för en automatisk växellåda med primär retarder utvecklats och implementerats. Tillsammans med andra modeller från fordon och drivlina skulle denna simuleringsmodell kunna användas för att uppskatta ett fordons bränsleförbrukning eller till att optimera olika fordonsparametrar.

De olika mekaniska komponenterna samt kontrollsystemet i växellådan modellerades separat. Dessa modeller kunde sedan sammanfogas för att bygga den slutliga växellådsmodellen. Alla modeller implementerades i det objektorienterade programmeringsspråket Modelica, som tillåter en stor återanvändningsbarhet till vardera enskild komponent.

Den implementerade modellen verifierades genom ett antal provcykler, utformade för att fånga olika beteenden hos växellådan. Dessa cykler har återskapats i simuleringsmiljön och med det kunde resultat från simuleringar jämföras mot data från ett verkligt fordon som utförde samma prov.

Från verifieringen har slutsatsen dragits att modellen uppfyllde målen med projektet. Målen var, att den slutliga simuleringsmodellen visar ett liknande beteende som en växellåda i ett verkligen fordon. Växlingar och låsning/upplåsning hos momentomvandlaren inträffande vid ungefär samma situationer i simuleringarna som i provningen med det verkliga fordonet.

Nyckelord: Drivlina, Simulering, Modelica, Dymola, Automatisk växellåda, Moment Omvandlare, Primär Retarder.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Scania is one of the leading manufacturers of trucks, buses and industrial engines on the international market. Even if trucks is the largest area of production, bus operations have a significant role in the company. Scania produce both city buses, made for shorter journeys, and coaches, made for longer journeys. No matter the bus type, the need for optimizing the drivetrain is important.

The group *Bus Powertrain performance* is responsible for shaping, specification and validation of drivetrain characteristics with respect to noise, fuel consumption, comfort and performance. Their knowledge of computer simulation, design and testing of the drivetrain and its components allows them to make calculated judgements about bus development.

In recent years the legal and customer requirements in automotive industry have escalated rapidly. Together with new emission standards, regarding carbon-dioxide, NOx[1] and other particulates, are continuing to come from customers and the EU. Along with the tough market, that is selling buses, demands on the development departments have never been higher.

During the last years, simulations have taken a larger and larger role in the development in the automotive industry. As computer performance continues to grow and new modelling tools arrive, vehicle manufactures are developing their own programming libraries consisting of their very own products, and using them in tests and trials. Using computer simulation is a good way of testing new product designs and/or vehicle varieties without building a vehicle prototype. Simulations can also be used in pre-studies and optimization of different features and components of the drivetrain. It allows developers to quickly try new ideas and come to a first decision about them. In short, it can be said that simulations can save manufacturers a lot of development time and money.

In city buses, it is popular for manufacturers to install an automatic gearbox. For that reason, a physical model of it should be implemented and present in the simulation libraries, so it can be used in the development process.

In this thesis, the main objective is the development of such a gearbox model, together with all of its components and its control system. The model was constructed and implemented, then the performance of the model was tested against a real vehicle. After assessing the models performance regarding shift points, vehicle speed and fuel consumption, a verdict is made regarding the model and if it shows the proper behaviour relative to the real gearbox.

[1] Nitrogen oxides.

Chapter 2

Problem Description

The need for simulating the vehicle drivetrain is pivotal during development of heavy vehicles. For simulations to give accurate results all modelled components of the drivetrain has to be implemented and up to date.

The development and validation of one of the drivetrain components is carried out in this thesis, and in this chapter the objectives and goals of the project is described, together with delimitations.

2.1 Purpose

The purpose of this thesis is to develop a physical model for an automatic gearbox so that it can be used in simulations with other modelled bus components. The finished model could for example be used for estimating fuel consumption for different cases, but could also be used for vehicle optimization.

It is therefore important that the model is detailed and that it reproduces the correct behaviour of a real gearbox.

2.2 Objectives

The project aim to achieve the following objectives:

- To create a model of a heavy vehicle automatic gearbox complete with physical and mathematical relationships.
- To implement a simplified model of a gear shift strategy, using road slope and engine speed calculations to capture the behaviour of a real gearbox. The shift strategy shall also prevent the gearbox from entering fail modes, such as 'gear hunting' [1] .
- Implement the model with the modelling software DYMOLA®[2], and if possible add it to existing libraries at Scania.
- Study the reliability of the gearbox model by simulation together with a complete drivetrain and comparing its performance to a real vehicle.

[1] 'Gear hunting' is a phenomenon that arises when optimizing shifts points. See Section 3.3.3.
[2] DYMOLA stands for Dynamic Modeling Library.

2.3 Delimitations

Listed below are the main delimitations that affect the work of the thesis:

1. No reverse gears need to be simulated in the gearbox.
2. The losses in the planetary gears are calculated as a percentage loss which are taken from a table. This is done instead of modelling the friction losses.
3. The hydraulics of the gearbox does not need to be solved with CFD[3] calculations but can be estimated through a system of DAE's[4].
4. For the shifting strategy only diesel combustion engines are considered.
5. The performance of most real drivetrain components are highly dependent on the component's internal temperature and the gearbox is no exception. In this project, internal temperature variations has been left out of the calculations and it is assumed that the gearbox is working at optimal temperature.
6. Tire radius is assumed to be fixed.
7. Tyre slippage is excluded from the traction calculation.
8. All translational motions are assumed to be longitudinal.
9. During validation the simulated vehicle does not need to include *auxiliary* systems such as, air conditioner or heater etc.

[3]CFD stands for Computational Fluid Dynamics.
[4]DAE - Differential Algebraic Equations.

Chapter 3

Theory and Methods

This chapter describes the theory of the mechanical components in the gearbox and the tools used to implement and simulate the model. As well as the method for validating the performance of the gearbox model.

3.1 Modelica

This section is dedicated to present the programming language used to implement the modelled automatic gearbox, *Modelica*. The goal is not to give the reader the sufficient knowledge to start programming with the language on his/her own, but rather give a description of how it is different from other languages and how it can be used for modelling.

Modelica is an object-oriented equation-based physical programming language, often used for modelling large, complex and heterogeneous physical systems [3, p. 19]. The same general class concept that other object-oriented languages uses is the same. Modelica also uses *graphical model editors* to define models by drawing a composition diagram by positioning icons representing the *components*[1] inside the model [3, p. 73].

To describe how a model is built, consider the icon view and source code of the simple vehicle model, Fig. 3.1. Consisting of only an engine and a resistance component.

[1] A component is an instance of a model, analogous to an object in object-oriented programming [11].

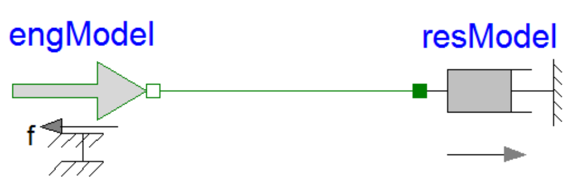

Figure 3.1: The *icon view*, of the `Vehicle`-model. Consisting of an engine component connected to a resistance component.

```
1 model Vehicle
2   ModelicaResistanceModel resModel "air, road and other resistances";
3   ModelicaEngineModel engModel "very simple engine model";
4
5   Real a "acceleration";
6   Real v(start=0.0) "velocity";
7   Real F "driving force";
8   parameter Real m = 0.5 "mass";
9
10 equation
11   //equations
12   a = der(v);
13   m*a = F;
14   connect(resModel.flange, engModel.flange);//connect the two Modelica models
15 algorithm
16   //statements
17   when sample(0, 0.1) then
18     // code here will be executed every 0.1 seconds (simulation time)
19     F := engModel.F_forward - resModel.F_res;
20   end when;
21 end Vehicle;
```

Studying the example code of the `Vehicle`-model together with Fig. 3.1 the following can be noted, line numbers refers to the example code:

– A model consists of both a graphical representation, or *icon view* (Fig. 3.1), and Modelica source code.

– The connection between the two components, `engModel` and `resModel` is represented in both the icon view as a line, and in the source code with the function `connect(...,...)`, line 14 and the connecting line between the two models in Fig. 3.1

– The component `engModel` is of the model type `ModelicaEngineModel`, analogous with being an instance of the class, line 3.

– The extent of the model begins with `model Vehicle` and ends with `end Vehicle;`, lines 1 and 26.

– All of the model's attributes are declared. In the model four variables are of type `Real` (same as double in C) line 5-8, and two other Modelica models added as object attributes, lines 2 and 3.

– Initial values can be set for variables, line 6.

– `engModel` has atleast the two variables `flange` and `F_forward`, line 14 and 19.

– Variables in other Modelica models can be accesses using dot-notation[2], line 14 and 19.

– The constant `m` is declared as a `parameter` and assigned a value.

– The keyword `equation` initiates the section where the model's defining equations are written, line 10.

– The keyword `algorithm` initiates a section where assignment statements can be written.

[2]dot-notation referring to the dot between the object and the accessed variable, e.g. `object.variable`.

– Code inside the `when`-loop is executed every tenth of a second (0.1) of simulation time, starting at $t = 0$, line 17.

– An explaining comment for variables can be written inside quotation marks `"..."`, other comments after `//`.

– Time derivative is given by `der(...)`, line 12.

3.1.1 Equations

As stated before Modelica is among else an equation-based programming language. In Modelica an equation does not describe assignment but equality, therefore they may have expressions on both right and left sides of the equality sign. It is possible to write assignment statements as well, this is done using the `:=` operator.

– Examples of Modelica equations: `x^2+y^2=R^2`, `m*der(v)=F` [3], `tan(phi)=X/Y`

– Examples of Modelica assignments: `R:=sqrt(x^2+y^2)`, `a:=F/m`, `phi:=atan(X/Y)`

An Equation do not dictate a certain data flow or execution order, this makes them more flexible than assignments and they are the key to the physical modelling capabilities and the increased potential for reusing Modelica classes [4, chap. 2.6]. Equations also makes models more versatile, for example Newton's second law, written above as `m*der(v)=F` can be used in three ways corresponding to all the possible assignment statements that can be made by rearranging the equation: `F:=m*der(v)`, `der(v):=F/m`, `m:=F/der(v)`. Additionally, by writing `der(v)` instead of `a` for acceleration both acceleration and velocity will be determined by the *simulation engine*.

An important note is that equations in Modelica *always* holds, meaning that declaring an initial value for a variable is different from other languages. While writing `Real position=0.5;`[4] means that `position` is effectively a constant, to initialize properly in Modelica one writes `Real position(start=0.5);`.

DAE's - Differential Algebraic Equations

A Modelica model is expressed by a system of equations consisting of the following two types of equations:

– *Differential equations* that contains time derivatives, the time derivative dx/dt is denoted by $\dot{x}$.

$$\dot{x} = a \cdot x + b \tag{3.1}$$

– *Algebraic equations*, involving algebraic formulas but no differentiated variables. Including non-linear equations.

$$x^2 + y^2 = R^2 \tag{3.2}$$

An equation system containing both differential and algebraic equations is called a system of *differential algebraic equations*, or DAE [4, p. 33].

DAE's are declared in a model after the keyword `equation`. If a user want to use assignment statements instead this can be done below another keyword, `algorithm`, see

[3] `der(...)` in Modelica is the time derivative of the argument variable, therefore this equation is a differential equation.

[4] Real is the declaration for a double in Modelica, that is a floating number of 64-bits.

the source code belonging to the vehicle example in section 3.1 line 15. Lifting demanding calculations into `algorithm` sections speeds up simulations since the user can control the frequency of the execution of the code.

3.1.2 Connectors

While models are defined by equations in Modelica, it is impractical to write all equations for a large system in only one model [11]. In the `Vehicle` example above the engine and resistance part of the system is divided into two separate component models, `engModel` and `resModel`. These components are *connected*, meaning that there is a dependence between them. Dividing physical models into smaller sub-models and modelling them separately is one of Modelica's greatest strengths and ensures model reusability.

Modelica has a special tool used to connect models, namely *connectors*. A connector is a model itself, that has an interface used to connect it with other connectors of the same type[5]. Is has already been stated that two sub-models can be connected through the function `connect(...,...)`. When connected, regular variables in the connectors are set as equal, while variables marked with `flow` are added to zero. Consider the case where two `Flanges` (from the example code below) are connected. This means that voltage is set as equal in both, while current is added up to zero, i.e. flowing from one `Flange` to the other [6].

```
connector Flange
  Real U;       //voltage
  flow Real I;  //current
end Flange
```

To create a model for an input/output signal, a connector with no `flow` marked variables can be used. Instead the variables can be marked with `Input` or `Output` which meaning is self-explanatory.

3.1.3 Acausal Modelling

In equations the variables that are input and output do not need to be stated, in contrast to assignments statements where the output is the variable on the left-hand side and the inputs are the variables on the right-hand side. The causality of equation-based models that allows inputs and outputs to be unspecified in the implementation is called *acausal modelling*.

With the acausal modelling style, the solution order of the equations will adapt to which variables that the *simulation engine* states as inputs and outputs to the computed solution. This is done continously during simulation and is the main advantage of acausal modeling, as well as making the models more reusable than traditional ones with fixed input and output [4, chap. 2.7].

Example: A Drivetrain From an Acausal Point of View

Figure 3.2 shows the direction of power flow and loads in the drivetrain components of a vehicle. Which one of power or load that is dominating is changing in different driving cases.

When a vehicle accelerates, the net power flow will go through all components from the engine to the wheels. But when the vehicle engine-brakes then loads are dominating and

[5] There are many types of connectors, electric, rotational, translational, etc.

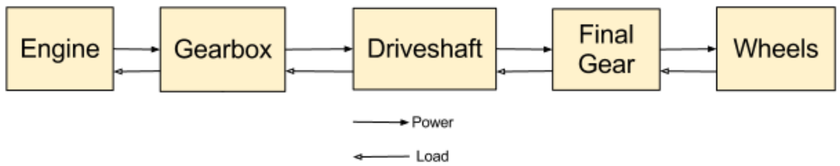

Figure 3.2: The power flow and load direction in a vehicle drivetrain model.

power flow is reversed. This means that the connections between the components must be able to act as both input and output.

This makes *acausal modelling* a viable strategy for modelling physical systems like this.

3.1.4 Executing Modelica Code

When executing a Modelica program things are a a bit different compared to other programming languages. As explained earlier, Modelica focuses on describing physical and mathematical systems with equations and not assignement statements. This would not make much sense to a compiler, which is why Modelica classes are not really compiled, but translated into objects by a simulation engine. This engine, manipulates the equations symbolically to determine the order of execution and which equations that signifies input/output [3]. Figure 3.3 and Algorithm 1, explains and illustrates the process of the simulation engine for executing a Modelica model [4, chap 2.20].

It is not required to perform this entire process for every simulation. The executable file (`dymosim.exe`) allows for parameter values to be changed and then a new simulation can be runned. Meaning that all steps in Algorithm 1 up to step 8 can be skipped if only parameters needs to be changed.

3.1.5 Numerical Integration in Modelica

In order to run a time dependent simulation a numerical integration algorithm is needed and Modelica offers the user a choice between many different solvers [3]. Most of the solvers use *variable time-step*[6] algorithms, which estimates the *local error*[7] after every time-step and then choose the next time-step such that the local error does not exceed the maximum local error. Maximum local error is defined by the relative and absolute error, which may be user specified. [3]

For this thesis the solver LSODAR, *Livermore Solver for Ordinary Differential Equations with Automatic Method Switching and Root Finder*, was used. LSODAR is an implicit ODE solver that uses an automatic switching method that determines if the system is most efficiently solved using methods suited for stiff or non-stiff problems [7].

Sometimes it is required to fetch data from some external data-file, this is done using so called *look-up* functions and Modelica have such functions implemented. Look-up functions can be slow performing and therefore they are best placed in the `algorithm`-section of model. That way the rate at which they are executed can be controlled by the user.

[6]The opposite to static or constant time step.
[7]Local error, is an accuracy measurement to the numerical solution over one time-step.

Algorithm 1 Stages of translating and executing a Modelica model.

1. Modelica source code is parsed and converted to an internal representation, usually an abstract tree.
2. The representation is analyzed and translated, resulting in a flat set of equations, constants, variables and function definitions. By this time the object-oriented structure is all but gone.
3. The flattened equations are topologically sorted through analysis of the data flow dependencies between them.
4. An *optimizer* module eliminates most equations through algebraic simplification algorithms, symbolic index reduction methods and the like.
5. The equations written on explicit form are converted to assignment statements. Since the system is already sorted and an execution order for the numeric solver is already established.
6. C code is generated and linked together with a numeric equation solver that solves the equation system.
7. The C code is compiled and a executable simulation file, usually called `dymosim.exe`, is generated.
8. During simulation the numeric solver computes the values for the system's variables during a user specified interval $[t_0, t_{fin}]$. If necessary the user also specifies parameter values.
9. The result is a set of functions of time that can be plotted or saved to a file (standard).

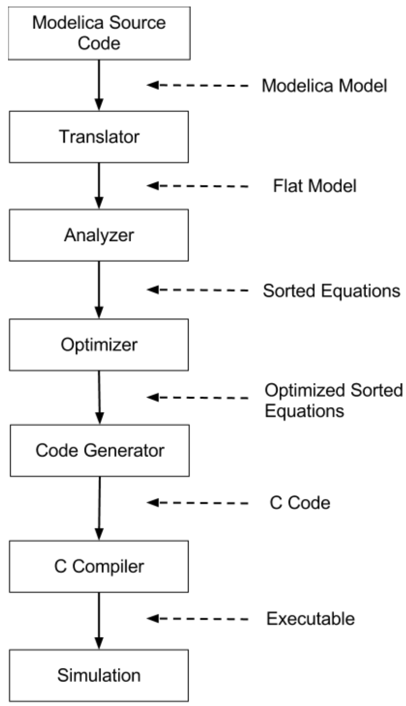

Figure 3.3: All the stages that a Modelica model goes through during execution, from source code to simulation.

3.1.6 Dymola

The program used to implement the model is Dymola®, *Dynamic Modelling Library*, a product from Dassault Systémes™. It provides the user with a *graphical user interface*, or GUI, to combine component model as well as a text editor for Modelica implementation. From Dymola the Modelica source code can be executed and simulated from inside the GUI, and it also provides tools for post processing the results.

3.2 Automatic Gearboxes in General

Combustion engines usually operate at high speeds which is not ideal for starting, stopping or travelling slowly. The gearbox's purpose is to manipulate the torque and speed ratios from the engine, giving the vehicle wider driving range. Beginning with gear one (1) with the highest torque conversion rate $r_1 = r_{max}$ as the gearbox shift to higher gears (2,3,...) the gear ratio decreases, even so far that it becomes less then one. This called an *overdrive* since speed increase and torque is reduced.

In the *ideal*[8] case the resulting torque, T_{eng}, and speed, ω_{eng} on the propshaft, after the gearbox, for some gear i with ratio r_i follows the following relations.

$$T_{eng} \cdot r_i = T_{prop} \tag{3.3}$$

$$\frac{\omega_{eng}}{r_i} = \omega_{prop} \tag{3.4}$$

[8]Ideal meaning that there is no power loss, that is $P_{eng} = P_{prop}$.

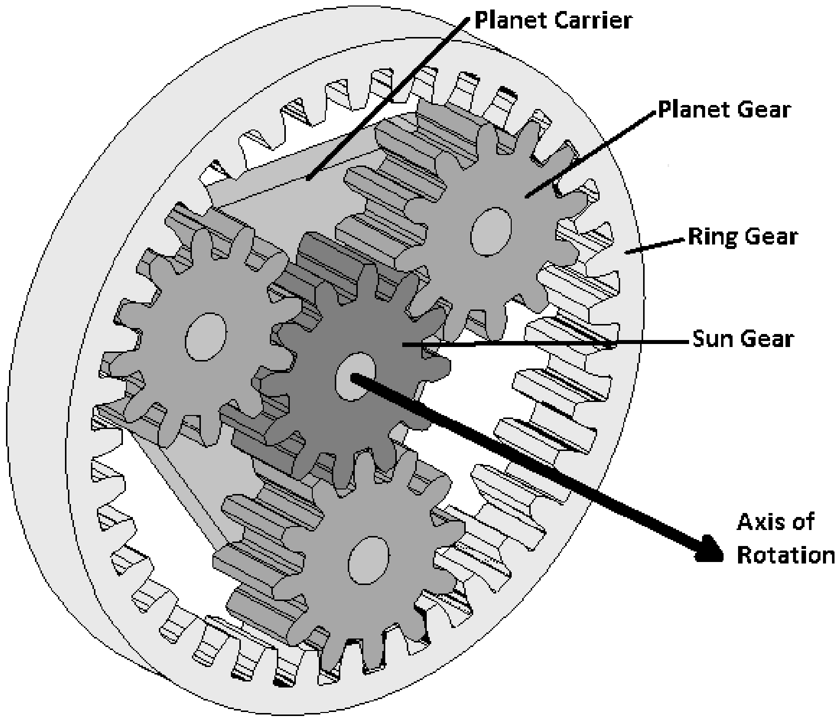

Figure 3.4: A planetary-gear set with its components identified.

Where T_{eng} and ω_{eng} is torque and speed from the engine. Power, P, for a rotational source is given by:

$$P = T \cdot \omega \tag{3.5}$$

3.2.1 Planetary-Gear Sets

The majority of the automatic gearboxes in commercial vehicles use a planetary-gear train to shift between gear ratios. It is made up by several planetary-gear sets, in some cases regular gears as well. A planetary-gear set consists of three major components, the *sun*-gear, internal *ring* gear and the planet gears together with the planet *carrier*. Each component may act as either input, output or be held stationary. The layout of the planetary-gear set is such that all components share the same rotational axis, see Fig. 3.4. This makes it ideal to use automatic gearboxes, where clutches and brakes can be used to engage or fix the individual components, allowing the engagement pattern to be changed without interrupting the torque flow [1, p. 670].

The lesson learned from Table 3.1 is that with only one planetary-gear set, torque amplification, overdrive and reverse gear can be achieved. It is common that there are a few planetary-gear sets mounted after each other, building a planetary-gear train with a wide range of gear ratios, with clutches and brakes in between. This allows a shift in gear ratios without interrupting the torque flow.

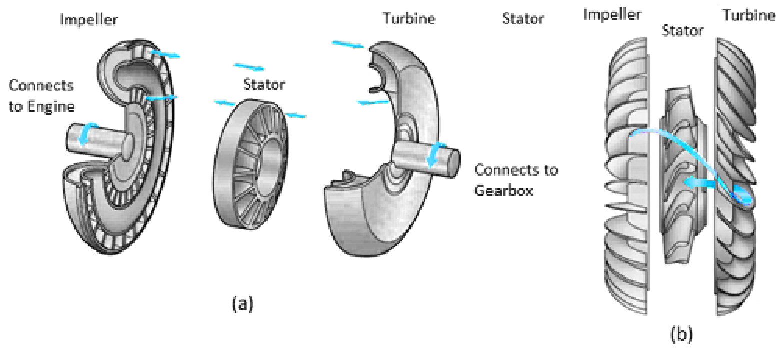

Figure 3.5: Cross section of a torque converter (a), and its blades (b), hydraulic flow and rotation denoted by the arrows.

Clutches

Even though a vehicle equipped with an automatic gearbox does not have a clutch pedal, this does not mean that there are no clutches in the gearbox. On the contrary, there are multiple clutches used in the gear train.

A clutch is made up of two friction plates and a mechanism that either presses them together or pushes them apart. In the latter case, this means that the connection between the plates is broken and the plates do not effect one another. The connection can also be partly broken, meaning that the two plates are slipping against each other and only parts of the power is transferred.

To name a clutch *stiff*, implies that there is no slippage between the plates when they are pressed against each other to the maximum capability.

3.2.2 Torque Converter

The torque converter is a type of hydraulic fluid coupling that connects the gearbox to the engine flywheel. There are three major functions that the torque converter brings to the gearbox. Firstly, when travelling at low engine speed, or even idling speed, the amount of torque travelling through the converter is very small allowing the vehicle to stop without killing the engine. Secondly, the torque converter gives the vehicle more torque during take-off's [9, chap. 4.2.1]. Finally, the hydraulics in the torque converter helps in reducing torque spikes that can occur in a drivetrain. This makes it more comfortable to ride in the vehicle.

Table 3.1: Equations of one planetary-gear set with different the different output, input and fixed components, Z_i denotes the number of cogs on the component i.

Input	Output	Fixed	Conversion ratio	Remark
Sun	Carrier	Ring	$1 + Z_{Ring}/Z_{Sun} > 1$	Torque amplification
Ring	Carrier	Sun	$1 + Z_{Sun}/Z_{Ring} > 1$	Torque amplification
Carrier	Sun	Ring	$\frac{1}{1+Z_{Sun}/Z_{Ring}} < 1$	Overdrive
Carrier	Ring	Sun	$\frac{1}{1+Z_{Ring}/Z_{Sun}} < 1$	Overdrive
Sun	Ring	Carrier	$-Z_{Ring}/Z_{Sun} < 0$	Reverse gear
Ring	Sun	Carrier	$-Z_{Sun}/Z_{Ring} < 0$	Reverse gear

The torque converter consists of an *impeller*[9], a *turbine*, a *stator* and a *lock-up clutch*, the first three can be seen in Fig. 3.5. The component that connects the torque converter to the engine is the impeller[10]. The impeller works as a centrifugal pump converting the rotational power from the engine to the hydraulic fluid, which is flung from the inside of the converter to outside, then entering the turbine. In the turbine, curved blades catch the fluid and changes its direction, see Fig. 3.5.

The reacting force from braking and redirecting the fluid causes the turbine (and the rest of the gearbox) to spin. The fluid then exits the turbine and flows into the stator. The stator also uses curved blades to change the fluid direction, but the stator does it much more aggressively than the turbine. The sharply bent blades of the stator can be seen in Fig. 3.5(b). This helps in guiding the fluid back to the impeller. There is a mechanism preventing the stator to spin with the fluid, only in the opposite direction, this forces the fluid to change direction.

In short, during the driving phase the stator redirects the fluid, while the impeller accelerates it and the turbine decelerates it. Even if the power losses are large, this actually means that the turbine torque is larger than the engine torque [1, 9]. The measurement used to denote the torque increase is the *conversion ratio* and it is given by

$$\mu = \frac{T_T}{T_P} \tag{3.6}$$

where T_P is the impeller torque and T_T is the turbine torque. Notable facts about the conversion ration is that it is always $\mu \geq 1$,[11] and it is at its maximum when the vehicle is standing still, then the turbine speed $\omega_T = 0$. Just like in the case where the driver is pressing the brake pedal and the braking torque is enough to keep the vehicle standing still. At this point the relative power loss between the impeller and turbine is 100%. Power loss in the torque converter is given by[12]

$$P_{loss} = \omega_P T_P - \omega_T T_T \tag{3.7}$$

and the relative power loss is

$$\eta_{rel} = \frac{P_{loss}}{\omega_P T_P}. \tag{3.8}$$

As the driver then releases the brakes the turbine will pick up speed, that is ω_T increases and as the turbine speed approaches the speed of the impeller, that is $\omega_T \to \omega_P$, the torque conversion drops. The power loss also decreases when this happens. This continues until the speed relation between the impeller speed ω_P and turbine speed ω_T, given by

$$\nu = \frac{\omega_T}{\omega_P}, \tag{3.9}$$

approaches a value ν_{coup},[13] where the torque conversion ratio has decreased to $\mu = 1$. In other words there is no torque increase. Looking at Eq. (3.7) this means that $P_{loss} = (1 - \nu_{coup})\omega_P T_P$ at this point. At this point the converter has served its purpose and it would be more efficient if the impeller and turbine would be locked in relation to each other, eliminating all losses inside. This can be achieved by a *lock-up clutch* (LuC), locking the

[9] Sometimes called a pump [9, chap. 4.2.1] giving it the subscript P in equations.
[10] Giving the impeller the same speed as the engine, $\omega_{eng} = \omega_P$.
[11] This means that $T_T \geq T_P$.
[12] Eq. (3.7) is basically $P_{loss} = P_{in} - P_{out}$.
[13] Typical value, $\nu_{coup} = 0.9$ [9].

torque converter and effectively allows the engine speed and torque to effect the gearbox directly.

As stated above the impeller is connected to the engine flywheel, meaning that they will have the same speed $\omega_P = \omega_{eng}$, but the hydrodynamic torque T_P produced by the impeller is not necessarily equal to the engine torque. Bosch et al. [1] gives the following equation for calculating the impeller torque:

$$T_P = \sigma \rho_{fluid} D_P^5 w_P^2 \tag{3.10}$$

where σ is the performance index of the converter, ρ_{fluid} is the density of the hydraulic fluid and D is the diameter of the impeller.

It has already been stated that the maximum torque conversion μ_{max} is achieved when the vehicle is at a stop, then $\omega_T = 0 \Rightarrow \nu = 0$ and as the turbine speed increases the speed ratio $\nu \to 1$ the conversion ratio will drop. This drop can be modelled as *linear* until reaching a coupling point ν_{coup} [1], where $\mu(\nu_{coup}) = 1$. Resulting in the following expression for μ.

$$\mu = \begin{cases} \mu_{max} - \frac{\mu_{max}-1}{\nu_{coup}}\nu & 0 \leq \nu < \nu_{coup} \\ 1 & \nu_{coup} < \nu \leq 1 \end{cases} \tag{3.11}$$

3.2.3 Retarder

A retarder is a non-wearing auxiliary brake used to reduce the wearing on the regular road-wheel brakes. It is activated by the brake pedal just like a normal brake and it is especially used on heavy vehicles for its non-wearing property. Retarders can either be hydrodynamic or electrodynamic and can be fitted either on the engine side, called a primary retarder, or the propshaft side, called a secondary retarder. [1, p. 672]

Following Eq. (3.3) the braking torque of the retarder is also effected by the gear ratio. Since it is improbable that the driver is pressing the accelerator pedal as well as the braking pedal, it is then safe to assume $T_{eng} = 0$ when the the retarder is active[14]. Then Eq. 3.3 can be rewritten as

$$T_{retader} \cdot r_i = T_{prop}. \tag{3.12}$$

From Eq. (3.12) it follows that, if $r_i > 1$, which is the case for low gears, the braking torque is magnified and if $r_i < 1$ braking torque is reduced. This means that for vehicles that are more likely to experience largely low speeds, like public-transport buses, a primary retarder is much more effective. Secondary retarders are widely used for vehicles that are known to travel at high speeds for long times.

3.2.4 Disengaging the Drivetrain

Buses are prone to experience many starts and stops in traffic, either from queuing, traffic lights or planned bus stops. Developers have found that disengaging the drivetrain will reduce fuel consumption in these traffic scenarios [13, 14].

The disengaging works similar to a *decoupling*, in a manual gearbox. Regularly during engine braking there is no to very little fuel injected. But, in the case when the engine speed, ω_{eng}, come close to the idling speed, ω_{idle}, the engine must inject fuel in order to keep from failing. Then, it is more fuel efficient to disengage the drivetrain and let the engine go idle. Relieving the engine from driving the internal loads of the drivetrain.

[14] T_{ret} and T_{eng} are parallel but with opposite signs.

3.3 GMS - Gearbox Management System

The purpose of an automatic gearbox is that the driver should not need to worry about gear shifting, shifting should be an automatic function. For this to work some control system is needed, that replace the drivers intuition and judgement for shifting between the different gears. This control system will hereafter be referred to as the *Gearbox Management System*, or GMS.

The GMS is also responsible for the control of the gearbox, like the locking/un-locking of the torque converter and disengaging the drivetrain. It also communicates with the other control systems via the vehicle's *CAN bus*. Including the control systems for en engine, and for the brakes.

3.3.1 CAN bus - Controller Area Network

A *CAN bus system* or *CAN bus protocol* is a network for data communication, and are widely used in today's motor vehicles. A *CAN bus* is a standard way for connecting various components such as sensors, actuators and/or computers in a motor vehicles. Allowing them to communicate and share signals with each other. [1]

3.3.2 Theory of Shifting Strategy

To understand why a change of gear is needed at an arbitrary time the engine torque curve has to be studied. Fig. 3.6 show a sketch for some arbitrary diesel engine's torque distribution with respect to its rotational speed. Every diesel engine has its own distinct torque curve, but most of them share the same principle. This principle can be summarized by the three regions labelled A, B and C in Fig. 3.6.

In *region A*, torque grows as speed increases, in *region B* torque remains more or less constant, and finally in *region C* torque drops as speed continues to rise. Never mind that the slopes look linear, Fig. 3.6 is a conceptual sketch and not a real torque curve. Remember that power is given by, $P = T\omega$, so there is power to gain from higher engine speeds even if torque remains constant or sometimes decreases to some extent.

Some important aspects about the torque curve is that the engine speed should never go below ω_{idle}, otherwise it dies, and also for any speed $> \omega_{idle}$, torque T_{eng} can assume any value in the shaded area under the curve. The curve is actually representing maximum torque $T_{max}(\omega)$.

The estimated engine speed $\hat{\omega}$ after a shift, $i \rightarrow i'$, can then be derived, from Eq. (3.4) (see Appendix B) to be:

$$\hat{\omega}_{eng,i'} \begin{cases} < \omega_{eng,i} & \text{if } r_{i'} < r_i\text{, up-shift} \\ > \omega_{eng,i} & \text{if } r_{i'} > r_i\text{, down-shift} \end{cases} \tag{3.13}$$

From Eq. (3.13) it can seen that ω_{eng} will drop when a shift to a higher gear takes place, that is $\hat{\omega}_{eng,i'} < \omega_{eng,i}$, and vice versa for shifting to a lower gear. This means that there might be a change in potential torque, T_{max} after a gear shift. For example, engine speed is reduced after shifting to a gear with to a lower ratio and in the case when the vehicle accelerates there might not be enough torque to continue accelerating. If a shift is taken at a wrong time. High engine speeds also means higher fuel consumption due to internal power losses. Therefore to find the best gear at any given moment is difficult to estimate from the torque curve alone, even if it is an important aspect.

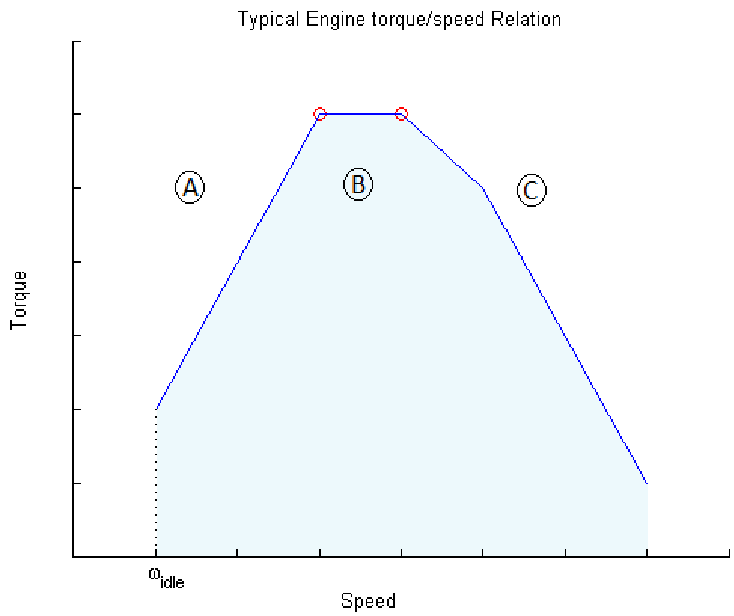

Figure 3.6: Conceptual sketch of an arbitrary diesel engine's torque distribution with respect to the engine speed.

The estimated $\hat{\omega}_{eng,i'}$ can then be mapped to the engine curve to get an estimate of the maximum available torque $\hat{T}_{eng}(\hat{\omega}_{eng,i'})$. With this a certain *shift limit* can be determined for the engine speed $\omega_{eng} \in [\omega_{\downarrow}, \omega_{\uparrow}]$. Which is calculated such that there is always enough power available after an intended up- or down-shift. If the engine speed would step outside this limit a gear shift could occur.

To estimate how much torque the engine is required to produce at a certain time the vehicle's equation of motion can be used. The total force on the vehicle can be summed up by all driving forces and running resistances resistances to be [1]:

$$F_{tot} = \underbrace{F_{wheel}}_{\substack{\text{Driving force at}\\ \text{tire footprint}}} - \underbrace{mgC\cos\theta}_{\substack{\text{Rolling}\\ \text{resistance}}} - \underbrace{mg\sin\theta}_{\substack{\text{Climbing}\\ \text{resistance}}} - \underbrace{cA\frac{\rho_{air}}{2}v^2}_{\substack{\text{Aerodynamic}\\ \text{drag}}} \tag{3.14}$$

See Appendix A for explanation about variables and subscripts. The driving force F_{wheel} can be calculated from, the wheel torque T_{wheel} and wheel radius R:

$$F_{wheel} = \frac{T_{wheel}}{R} \tag{3.15}$$

where

$$T_{wheel} = T_{eng} r_i r_{FD} \eta_{tot} \tag{3.16}$$

η_{tot} is the efficiency of the entire drivetrain and r_{FD} is the ratio of the drivetrain's final drive. Combining Eq. (3.14) & (3.15) together with Newton's second law[15] the following equation is achieved.

$$ma = \frac{T_{wheel}}{R} - mgC\cos\theta - mg\sin\theta - cA\frac{\rho_{air}}{2}v^2 \tag{3.17}$$

Eq. 3.17 is the vehicle's equation of motion and that needs to be taken into consideration when determining shifting.

Rajesh Rajamani et al. [9] shows, in his book, an example of a shift schedule for an automatic gearbox. Noting the up-shift for each gear shift occurs at higher speeds as the accelerator pedal (throttle)position, $p_{acc} \in [0, 100]\%$, is more pressed. This means that the driver requests more torque from the engine.

3.3.3 Gear Hunting

Gear hunting is a phenomenon that can arise when optimizing gear shifting points. It occurs when shifting limits are set to close for two neighbouring gears. An illustration of how gear hunting may look is found in Fig. 3.7. Note, there is not necessarily something wrong with taking any of the shifts,individually. Rather it is the switching back-and-forth that causes performance issues.

It would be most uncomfortable for both the driver and passengers to experience this behaviour and because of the lower efficiency during shifts, it becomes fuel consuming. Modern gearboxes have control systems protecting them from entering these kinds of situations.

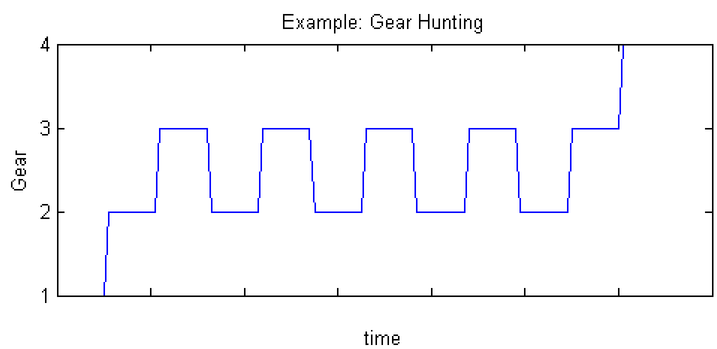

Figure 3.7: Sketch of how *gear hunting* may look like. Here the engaged gear is plotted against time.

3.4 Validation Method

To build a model of a system that represents reality is easy, the difficulty lies in making it accurate [6, chap. 1.6]. To be able to use and rely on the results of a model, it first has to be verified and validated. In order to do this, in this thesis, a comparison is made between the behaviour and fuel consumption of the model against the behaviour and fuel consumption of a real vehicle.

The validation of the gearbox model was made by performing certain test cycles with a vehicle equipped with an automatic gearbox with a primary retarder. During the test,

[15] Newton's second law of motion $F_{tot} = ma$.

Table 3.2: The test cycles designed to validate the gearbox model.

Cycle	Description
Test 1	Using full throttle to get through all the gears, then slowing down to a standstill.
Test 2	Full throttle until vehicle reaches 30 km/h then brake to a standstill.
Test 3	Full throttle until vehicle reaches 50 km/h then brake to a standstill.
Test 4	Using constant throttle trying to reach some equilibrium speed. Throttle positions: 30%, 40%, 60%, and 80%. (Four cycles)
Test 5	Starting from standstill in 16% and 18%-slope respectively and then driving up the hill. (Two cycles)

different signals were recorded from the vehicle's CAN-network. The driving tests were designed to capture different vehicle behaviours, that the model is required to handle in order to reproduce reliable simulations. Table 3.2 gives descriptions to all test cycles that were designed to evaluate the model.

By giving some of the recorded signals as input to the simulation environment the tests could be recreated and runned. This way a comparison between the CAN-signals and the simulation variables could be made. When comparing simulation results to reality the following aspects were evaluated.

1. The overall shift behaviour.
2. Locking and unlocking of the torque converter's LuC.
3. The engine speed at the shift points.
4. The vehicle speed.
5. Road slope calculation.
6. Fuel consumption.

3.4.1 SORT - Standardized On-Road Test Cycles

Fuel consumption is simulated and compared because if it shows poor results then there must be something wrong with the gearbox model. Since all other components are assumed to be correct. For this reasons the fuel consumption is validated.

SORT is a set of bus test cycles developed by the UITP[16] to evaluate fuel consumption [5]. The standardized cycles makes it easy for buyers to compare the performance of different products. But they are also used by manufacturers to see how competitive their products are in fuel efficiency aspect. Therefore, SORT was chosen as the method for validating the fuel consumption.

There are three different SORT cycles, each of them representing one specific type of urban driving scenario. Every cycle is composed of three *sections*, each section consisting of four *parts*. In the first part, the vehicle accelerates from a stand still. Second, the vehicle

[16] UITP - International Association of Public Transport.

travels at constant speed, Then, follows a deceleration part to a stop, and finally the vehicle have a certain *idling* time, meant to reflect traffic dependent stops. During the last idling time the bus doors are opened and closed to represent passenger boarding. The velocity distribution for SORT 1 can be seen in Fig. 3.8 and Table 3.3 shows the design for all three cycles [5].

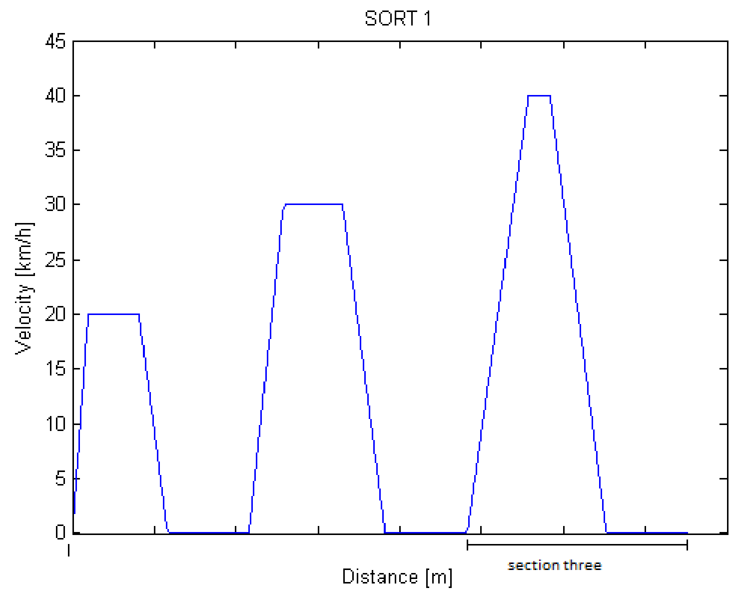

Figure 3.8: Target velocity vs. time for SORT 1.

Table 3.3: Design of the three SORT cycles. [5]

	SORT 1 "Urban"	**SORT 2 "Mixed"**	**SORT 3 "Suburban"**
Rated average speed [km/h]	12.6	18.6	26.3
1st section v-const. [km/h]	20	20	30
2nd section v-const. [km/h]	30	40	50
3rd section v-const. [km/h]	40	50	60
Length of stops [s]	20 / 20 / 20	20 / 20 / 20	20 / 10 / 10
Total cycle length [m]	520	920	1 450

Chapter 4

Modelling

4.1 Mechanical Modelling of Components

The general design of how the components are situated in an automatic gearbox is pretty straight forward. Since the torque converter always connects to the engine, the retarder is placed before the planetary gear-train and the control system, GMS, that governs their functions. The general description of how the modelled automatic gearbox looks was sketched see Fig. 4.1.

The final DYMOLA model was constructed using Fig. 4.1 and by looking at the graphical final icon view of the model in `Dymola`, Fig. 4.2, similarities can be found. Like the GMS connecting to most components through connecting *ports* (the connections between the arrowhead-like ports) and that torque converter connects to retarder and planetary geartrain. Differences lie in all the inertia components and the clutch between *inertia2* and *inertia3*. All mechanical connections (*flanges*) are of the *rotational* type. Which means that the rotational speed is the same between two connections and torque `flows` from one connection to the other.

From Fig. 4.2, observed that *flange_ a* will connect the gearbox to the engine model and *flange_ b* connects the gearbox to the propshaft model. There is also one more external port, the *six pointed star* at the top if the figure, this is the CAN-port connecting the GMS to the other control systems of the vehicle. This allows the GMS to communicate with other control systems. Remember, that torque can flow in both directions from *flange_ a* to *flange_ b* or from *flange_ b* to *flange_ a*, depending on the situation. Each of the major components are given an individual description in the following sections.

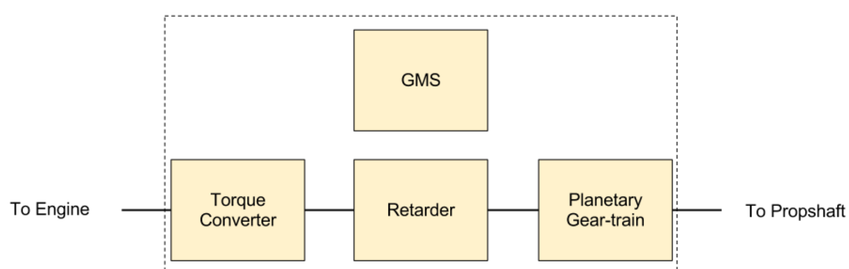

Figure 4.1: General sketch of an automatic gearbox with primary retarder and control system.

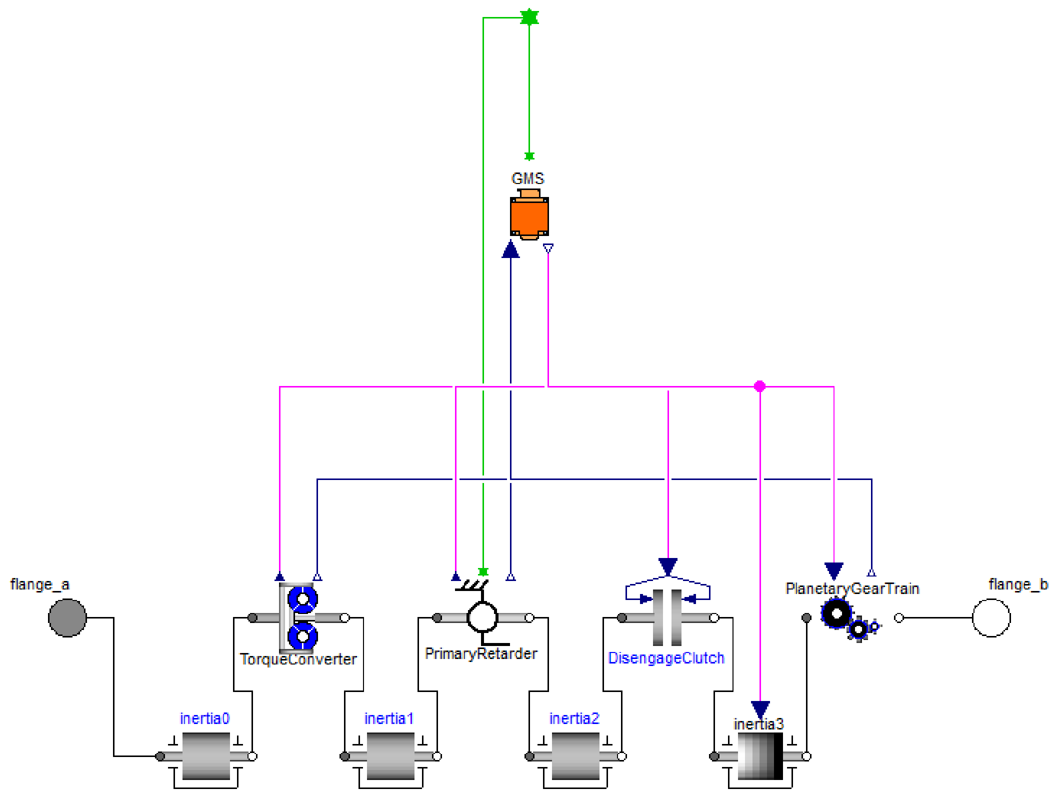

Figure 4.2: DYMOLA icon view of the automatic gearbox model.

4.1.1 Torque Converter Model

The *TorqueConverter* model contains other sub-models and not just source code. This means that it is built up by other DYMOLA components just like the gearbox in Fig. 4.2. This solution was practical since Modelica already have a *clutch*-model in its standard library that was used to model the converter's LuC (*lockUpClutch* in Fig. 4.3). Besides the clutch the converter consists of two connectors (*flanges*), two ports (*inport*/*outport*) and a hydraulics component.

The LuC is controlled by the GMS through the *inport*. The clutch is assumed to be stiff, which mean that the friction coefficients are very big. The threshold value for locking the LuC was set to $\nu_{lock} = 0.9$ (Rajamani et al. [9]). Other parameters that need to be set by the user include, ρ_{fluid}, D_P and σ.

The flanges are connected to both the LuC and the hydraulics component, meaning that torque can flow through either of them. If the clutch is closed then torque will only flow through the LuC, and because the clutch is stiff this results in $T_{flange\_a} = T_{flange\_b}$. Vice versa, if the clutch is open, torque only flows through the hydraulic component. The final case is when the clutch is in the process of opening or closing, then torque flow is split and divided between both components.

When the LuC is opened or partly opened, the hydraulics of the converter kicks in and torque conversion is calculated. The general approach for calculating the hydraulic dynamics of the torque converter with Modelica is found in Algorithm 2.

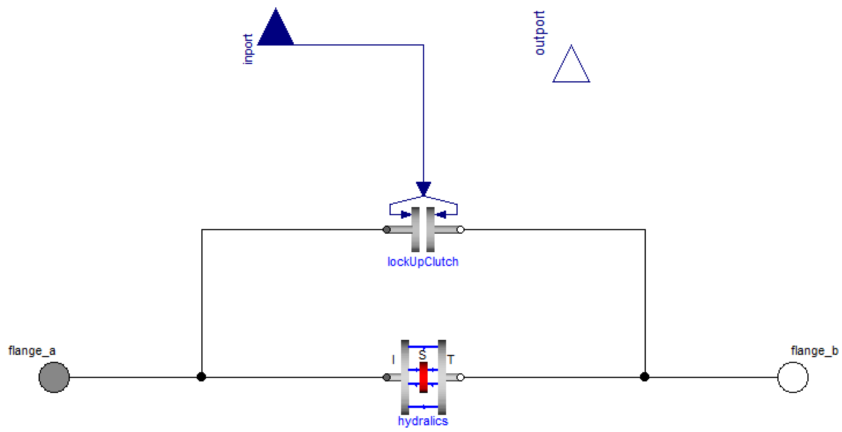

Figure 4.3: DYMOLA icon view of the torque converter model.

Algorithm 2 Calculation of the converter hydraulics.

---

equations

Torque and speed equations
$T_{flange\_a} = T_P$, Impeller torque
$T_{flange\_b} = T_T$, Turbine torque
$\omega_{flange\_a} = \omega_P$, Impeller torque
$\omega_{flange\_b} = \omega_T$, Turbine torque
$\nu = \dfrac{T_T}{T_P}$

algorithm

if $\nu \leq \nu_{coup}$ **then**
Case, when hydraulics are active
Calc. $T_P := \sigma \rho_{fluid} D_P^5 w_P^2$
Calc. $\mu := \mu_{max} - \dfrac{\mu_{max} - 1}{\nu_{coup}} \nu$
Calc. $T_T := \mu \cdot T_P$
else if $\nu > 1$ **then**
Case, when braking against the converter
$T_T = T_P$
end if

4.1.2 Retarder Model

During braking the driver signals requested braking torque by pressing the brake pedal. The retarder uses the position of the brake pedal to calculate how much of its full potential that should be used. This calculation is user defined, the maximum retarder torque $T_{ret,max}$ can also be set by the user, since every retarder has its own potential.

The braking torque the retarder produces is determined from a message on the vehicle's CAN-network, describing the required retarder percent torque $\lambda \in [0, 100]\%$ [12]. The final expressions for the retarder brake torque become:

$$T_{brake,GB} = \frac{\lambda}{100} T_{ret,max} \tag{4.1}$$

and

$$T_{brake,prop} = r_i \frac{\lambda}{100} T_{ret,max}. \tag{4.2}$$

4.1.3 Planetary Gear-Train Model

To keep the gearbox model as general as possible it was decided to implement the planetary gear-train as a whole (one unit). Then, the individual gear-sets and mechanisms to lock the ring gear, planet carrier and sun would not need to be modelled. A too detailed model would also lose its re-usability since the structure may vary between different gearboxes. Modelling it as one unit also leaves the possibility to be used as a gearbox that uses a combination of planetary and mechanical gears or only mechanical gears.

The remaining parameters that the chosen model would require is the ratios for each gear and some way to get the correct energy losses. Pelchen and Schweiger et al. [8], states that frictional losses in a planetary-gear set originates from both the *bearings* and through *meshing*[1]. Since the entire geartrain is modelled as one unit, the efficiency[2] of gear i, η_i, for each gear step is needed. These could be found experimentally or calculated through simulations of the entire gear-train, and then used in this model as a reduction of torque, see Eq. (4.3).

$$T_{out} = \eta_i \cdot T_{in} \tag{4.3}$$

In Eq. (4.3) T_{in} is the turbine torque T_T, of the torque converter. Which is the same as the engine torque T_{eng} if the converters LuC is closed and the impeller and turbine are locked together.

4.1.4 Inertia Model

The different inertias in Fig. 4.2 all represent the inertias for a specific real component. Which component that belongs to each inertia is composed in Table 4.1

As stated in the section above the planetary gear-train is modelled as one component. The effective inertia of the gear-train will change depending on how the planetary gears are engaged. To model this the component seen as `inertia3` was implemented and its inertia variates with each gear. It has the engaged gear as input and sets the inertia according to a specified value. The other inertias are taken from Modelica's standard library and kept constant through the simulation.

[1]Losses that originates from the friction between the teeth of gear wheels are called meshing losses.
[2]Efficiency is calculated by: $\eta = P_{out}/P_{in}$.

Table 4.1: Description of the model's inertia components.

Model Name	Description
inertia0	Converter impeller inertia.
inertia1	Converter turbine inertia.
inertia2	Retarder inertia.
inertia3	Inertia of the planetary gear-train. Varies with each gear.

4.1.5 Disengaging the Drivetrain Model

To model the mechanism that disengages the drivetrain, a stiff clutch was introduced. *DisengageClutch* in Fig. 4.2. The position of it was chosen so the geartrain is still connected to the propshaft. That way the engine do not waste power to drive the geartrain, together with its energy losses. Resulting in reduced fuel consumption when the clutch is open.

The parameters and conditions that activates this feature was never examined during this thesis. Because of time shortage.

4.2 Shift Strategy

The *GMS* interact with most of the mechanical components through the *arrowhead* connections in Fig. 4.2. It is the GMS that is responsible for all controlling aspects of the gearbox, including when a gear shift should take place. This section presents an overview of the strategy of determining a shift in gears.

The following variables require calibration or user specification for the shift strategy:

- The engine maximum torque vs. speed curve, together with the idling speed ω_{idle}.
- The shift limits depending on accelerator pedal pressure for each gear, Rajamani et al. [9].
- All the gear ratios of the gearbox.
- The time for one shift to finish, τ.
- Set values for the parameters: m, C, c, g, A, ρ_{air} and r_{FD} for Eq. (3.17) (for notations see Appendix A).

At any moment, the shift limits are first determined by the accelerator pedal pressure for gear i. The limits creates a boundary $[\omega_{\downarrow}, \omega_{\uparrow}]$ for the engine speed. While ω_{eng} is inside this limit no shift will occur. But, if the engine speed should reach $\omega_{eng} > \omega_{\downarrow}$, an up-shift occurs or if $\omega_{eng} < \omega_{\uparrow}$ a down-shift occurs. This is not the entire truth however. There are two more important aspects that effect the shifting limits.

First is the road gradient, θ. Since buses are heavy vehicles the gradient force becomes dominating even for relatively small angles, this makes the road slope an important parameter. It turns out that θ is the only unknown in the vehicles equation of Eq. (3.17), after parametrization. Vehicle speed is know through the CAN-network and Modelica can compute acceleration with `der(v)`. Hence, in this model Eq. (3.17) is used to determine the road slope.

The second is the rate of change in vehicle momentum, i.e. acceleration/deceleration. After a shift to higher gear the available engine power is almost always less than before.

Because acceleration is very energy consuming shifting should occur at higher engine speeds, resulting in more available power after the shifting is done.

Acceleration and road slope are taken into consideration by creating an offset β to the shift limits. The new limit can be written as $[\omega_{\downarrow} + \beta_{\downarrow}, \omega_{\uparrow} + \beta_{\uparrow}]$. The method or function for determining β needs to be set by the user.

The method for determining an actual gear shift, for gear i can be summarized by Algorithm 3.

Algorithm 3 Process for determining shift points, for gear i.

Get $\omega_{\uparrow}(p_{acc}, i)$ from table
Get $\omega_{\downarrow}(p_{acc}, i)$ from table
Calculate the offset, β, through:

- Road slope, θ, calculated from Eq. (3.17).
- The vehicle acceleration, a.
- Current gear, i.

if $\omega_{eng} > \omega_{\uparrow} + \beta_{\uparrow}$ **then**
  Shift to $i + 1$
else if $\omega_{eng} < \omega_{\downarrow} + \beta_{\downarrow}$ **then**
  Shift to $i - 1$
end if

4.2.1 During a Shift

In previous chapters it is stated that a planetary-gear train can switch gears without interrupting the torque flow. What really happens with the torque between a gear change is unknown. In the gearbox model it was chosen to run the new ratio through a filter (see Appendix C) with half of the shift time as its time constant, $T = \frac{\tau}{2}$.

4.2.2 Torque Reduction

During a shift the model issues a torque reduction request to the engine. This torque request, T_{req}, is used to combat torque spikes that originates from the sudden speed change in the engine. The request depends on the shift progress $t_{prog} \in [0, \tau]$ and the time for the shift τ

$$T_{req} = \begin{cases} T_{eng}\left(1 - \frac{t_{prog}}{\tau}\right) & , 0 \leq t_{prog} \leq \tau/2 \\ T_{eng}\left(\frac{t_{prog}}{\tau}\right) & , \tau/2 \leq t_{prog} \leq \tau \end{cases} \tag{4.4}$$

4.2.3 Dealing with Gear Hunting

To deal with the gear hunting phenomenon, a time delay, Δt, was introduced in the model. The delay is introduced to deal with the shift limits that may arise from using the offset, β.

For example, for some gear i the limit for shifting to $i+1$ may be lower then the gear's $i+1$ limit to shift to i.

$$(\omega_{\uparrow} + \beta_{\uparrow}|_i < (\omega_{\downarrow} + \beta_{\downarrow}|_{i+1}$$

Resulting in switching back and forth between the two gears, see Fig. 3.7. The delay counteracts this by preventing shifting back to the previous gear for Δt seconds. This allows the engine speed to stabilize and leave the crucial area that would otherwise create a gear hunting situation. Algorithm 4 shows the process of this delay.

Algorithm 4 The gear hunting delay.

if Last up-shift time $<$ current time $+ \Delta t$ **then**
  do not allow down-shift
else if Last down-shift time $<$ current time $+ \Delta t$ **then**
  do not allow up-shift
end if

Chapter 5

Results

One of the most important results of the thesis project is the implemented gearbox model explained in Chapter 4. This chapter presents the simulation results using the gearbox model. The chapter begins with the SORT cycle simulations used to validate the fuel consumption. Then moving on to the tests designed to validate the model behaviour explained in Table 3.2. The data for the real vehicles used in the test are presented in Table 5.1, all tests were performed with the same vehicle except for the SORT cycles.

Table 5.1: Vehicle data for the test cycles

Parameter	SORT tests	Other tests
Weight	14 500kg	12 000kg
Engine power	170kW	228kW
Number of gears ratios	6	6
Torque Converter, ν_{coup}	0.9	0.9

5.1 SORT Cycles

Table 5.2 presents the fuel consumption results from the real testing and the simulations of the SORT cycles. The design of the SORT cycles can be found in Table 3.3.

Table 5.2: Results from SORT cycle simulations.

Cycle	Real Vehicle[l/100km]	Simulation[l/100km]
SORT 1	48.0	42.7
SORT 2	41.7	35.8
SORT 3	-	32.4

5.2 Other Test Cycles

In the following figures, Fig. 5.1-5.10, are the results from the simulations and real testing that was performed in the thesis. In all figures the solid line is the data from the test with the real vehicle test and a the dashed line is from the simulations. The list below gives a short description of the variables that were chosen for the plots.

Vehicle Speed : Gives a picture of the vehicle behaviour and help to understand the current gear choice.

Engine Speed : A shift point can be clearly seen as a jump or a dip in the engine speed.

LuC : The converter's lock-up clutch. Value of 1 means closed, that the impeller and turbine are locked in place, and vice versa for value 0.

Current Gear : The engaged gear. Ideally the test and simulation should have the same gear engaged, if the test cycle is perfectly reproduced.

Accelerator Pedal : Ranging from 0-100%, shift points dependent on the position of the pedal. Where 100% means full throttle and 0% is no throttle at all.

Slope : Test cycle 5 was designed to validate the calculations of the road gradient. Therefore it is only plotted for this test, the other tests were performed on flat ground.

Test cycles 2 and 3 were even repeated multiple times which gave a small difference in the results from the real vehicle and in turn the simulation as well. The time scale is not necessarily the same between two figures, but it is for all plots in one individual figure.

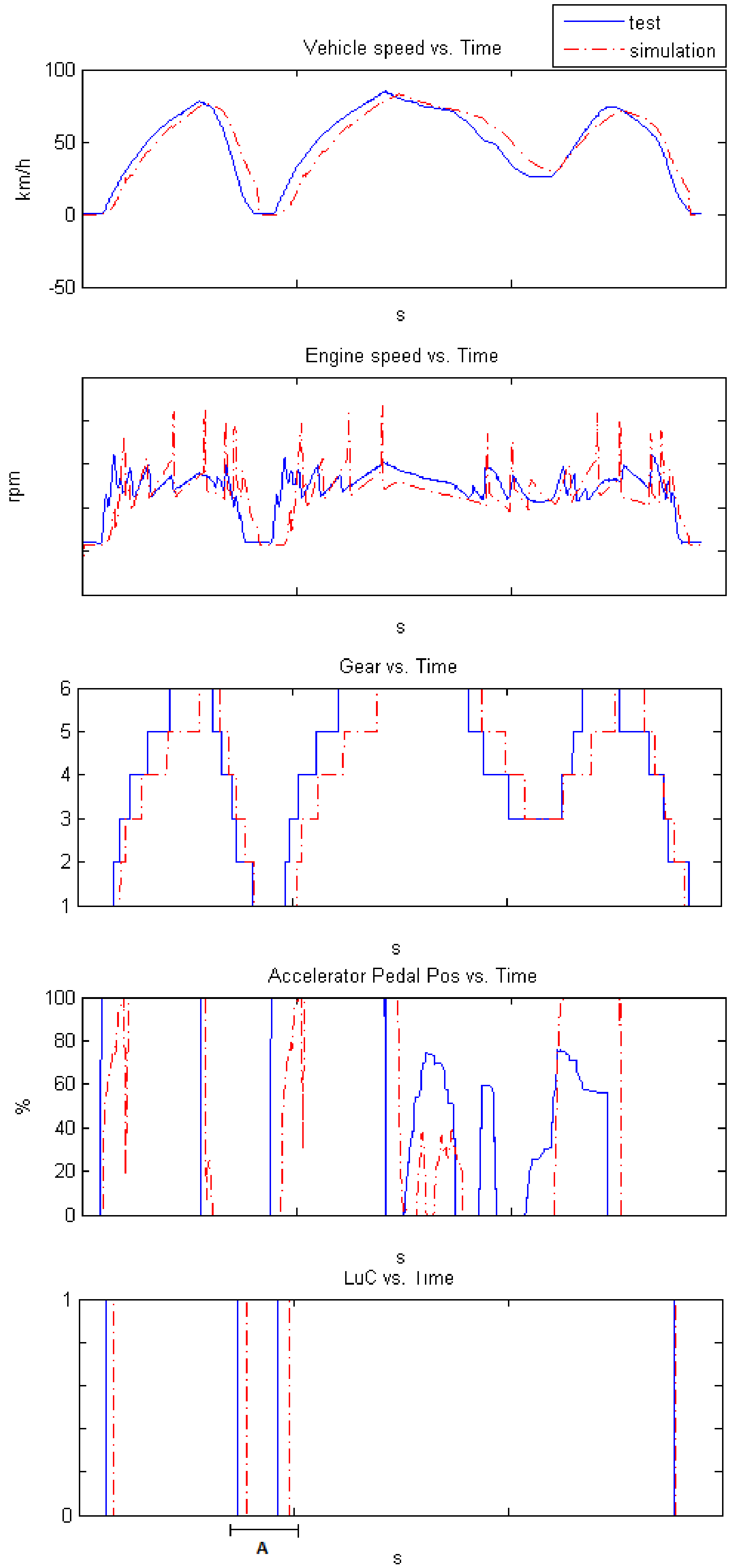

Figure 5.1: Results from test cycle 1.

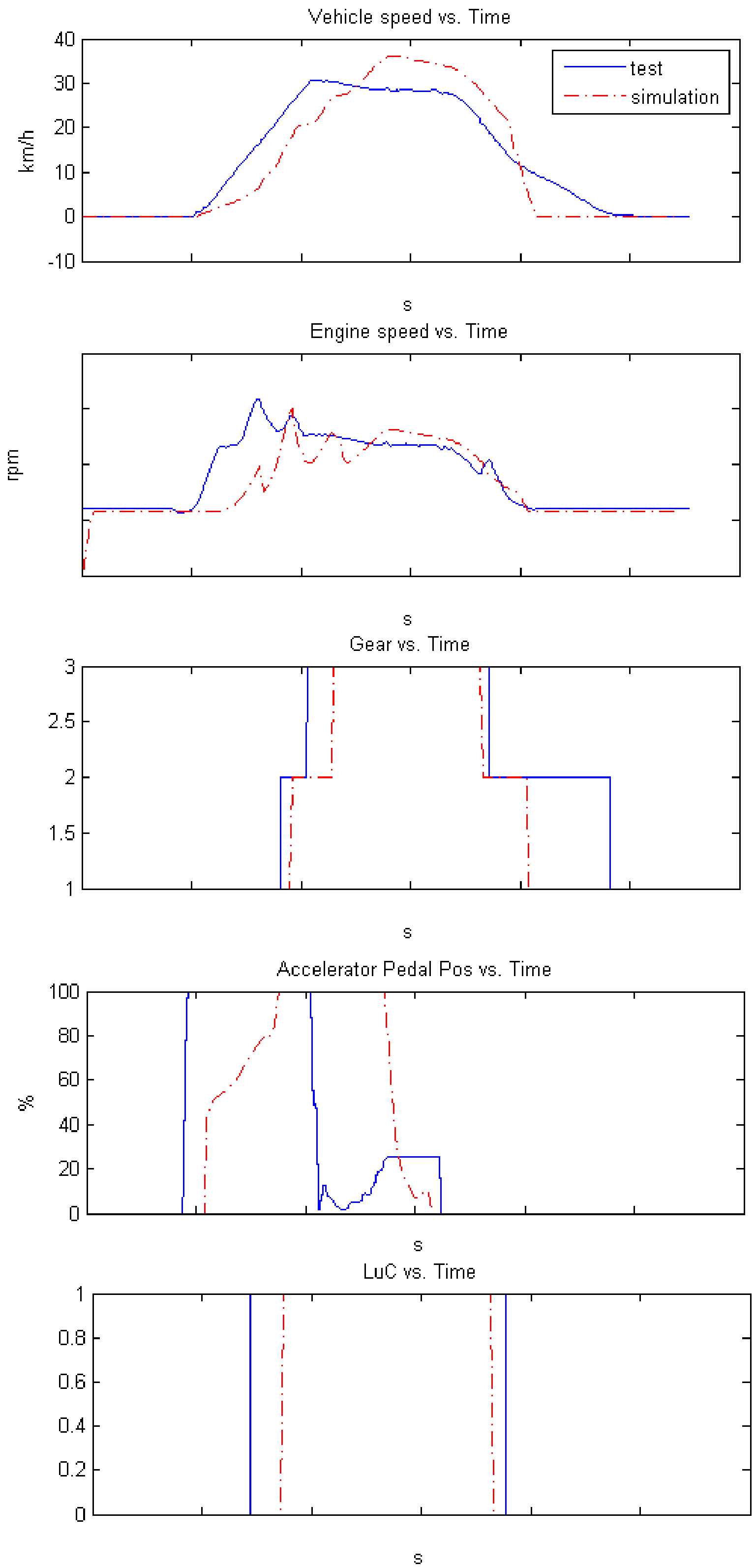

Figure 5.2: Results from the first attempt of test cycle 2.

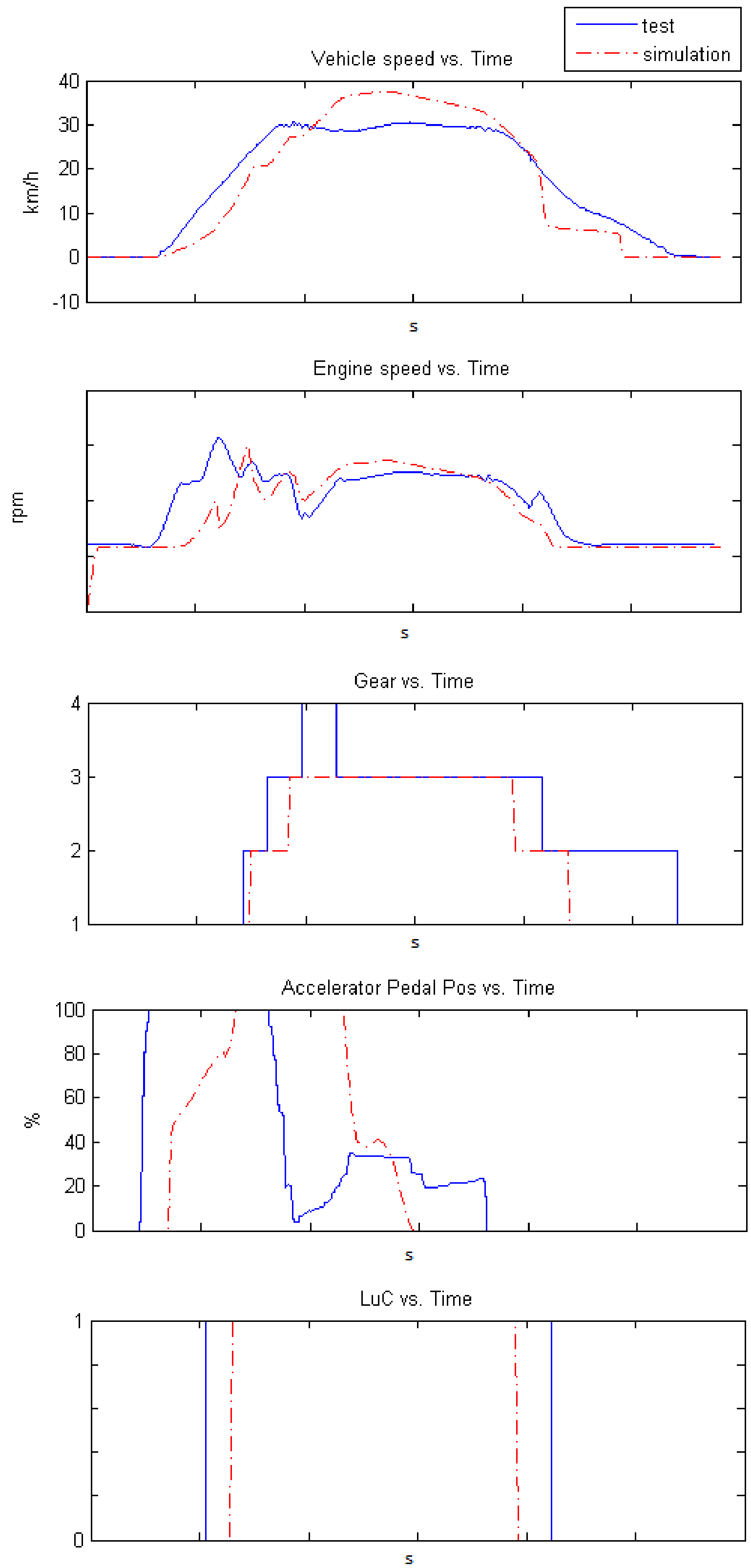

Figure 5.3: Results from the second attempt of test cycle 2.

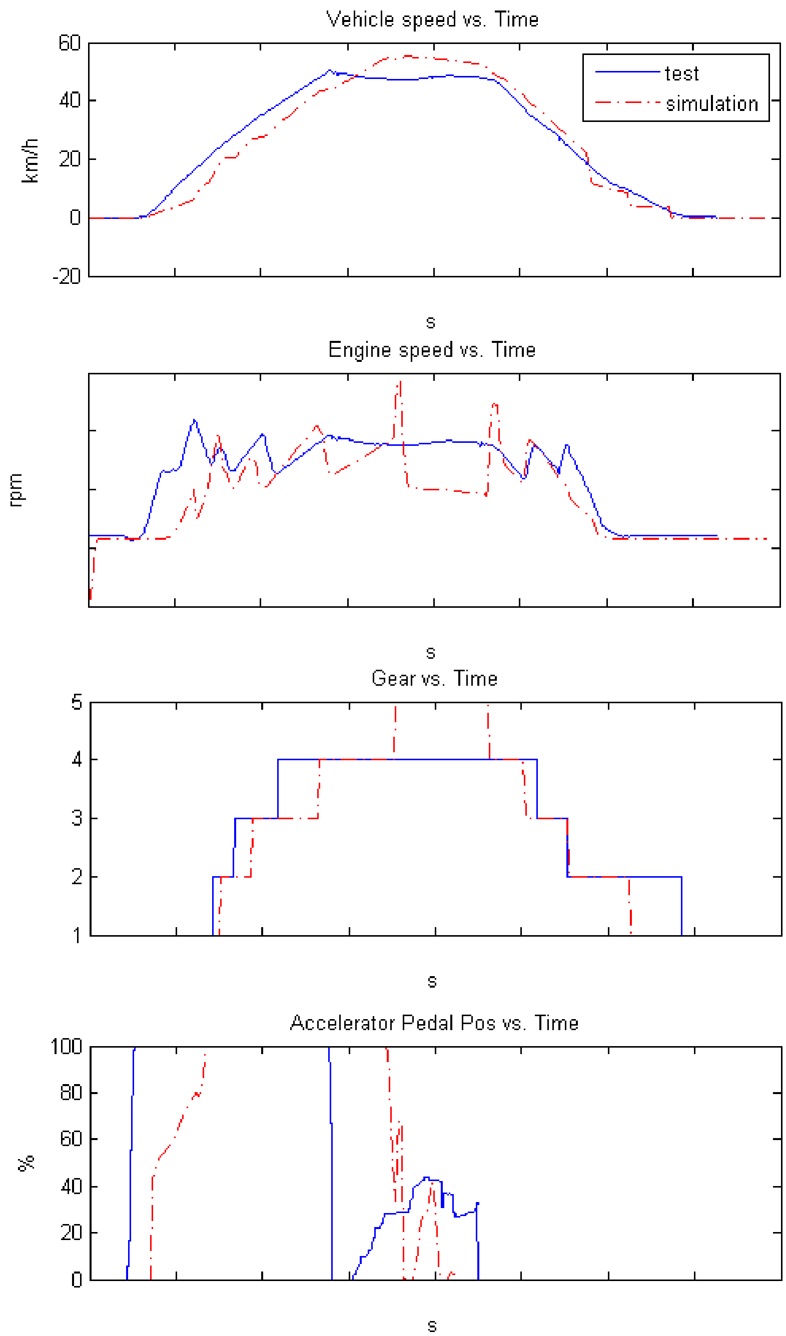

Figure 5.4: Results from the first attempt of test cycle 3.

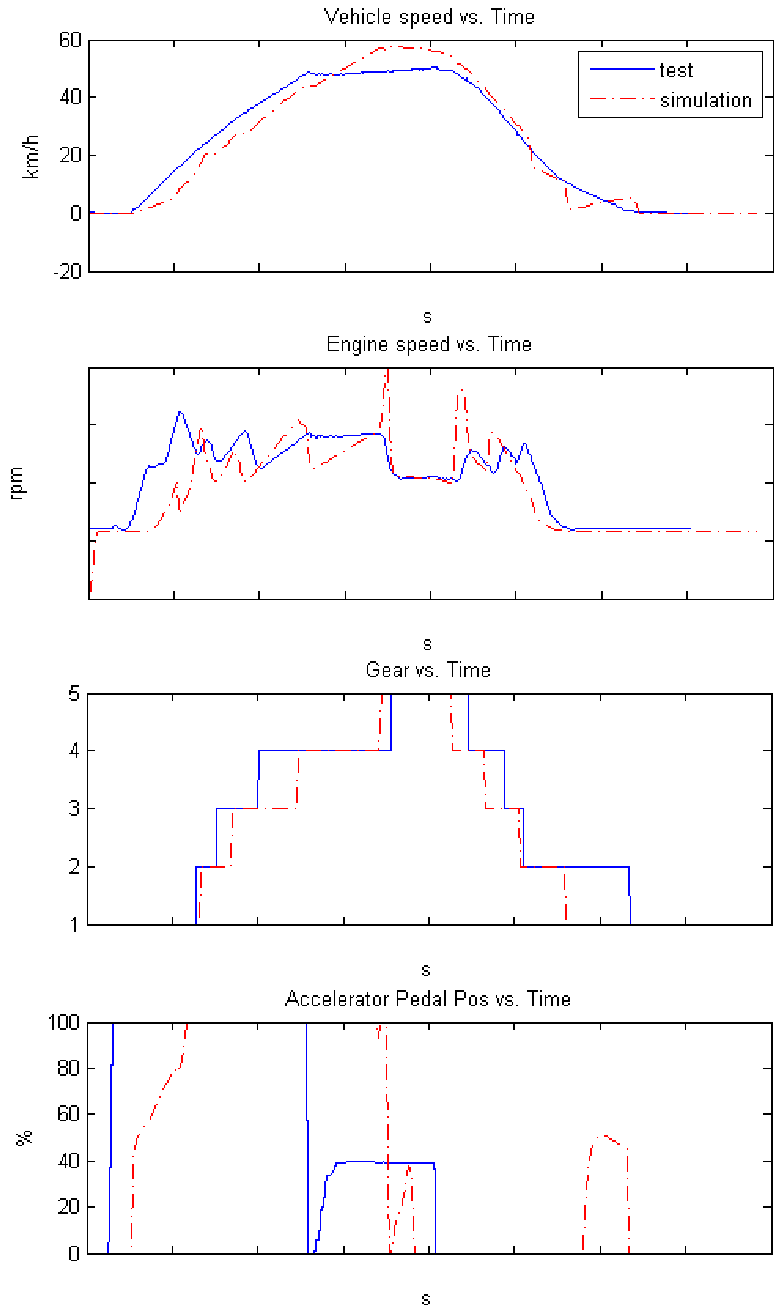

Figure 5.5: Results from the second other attempt of test cycle 3.

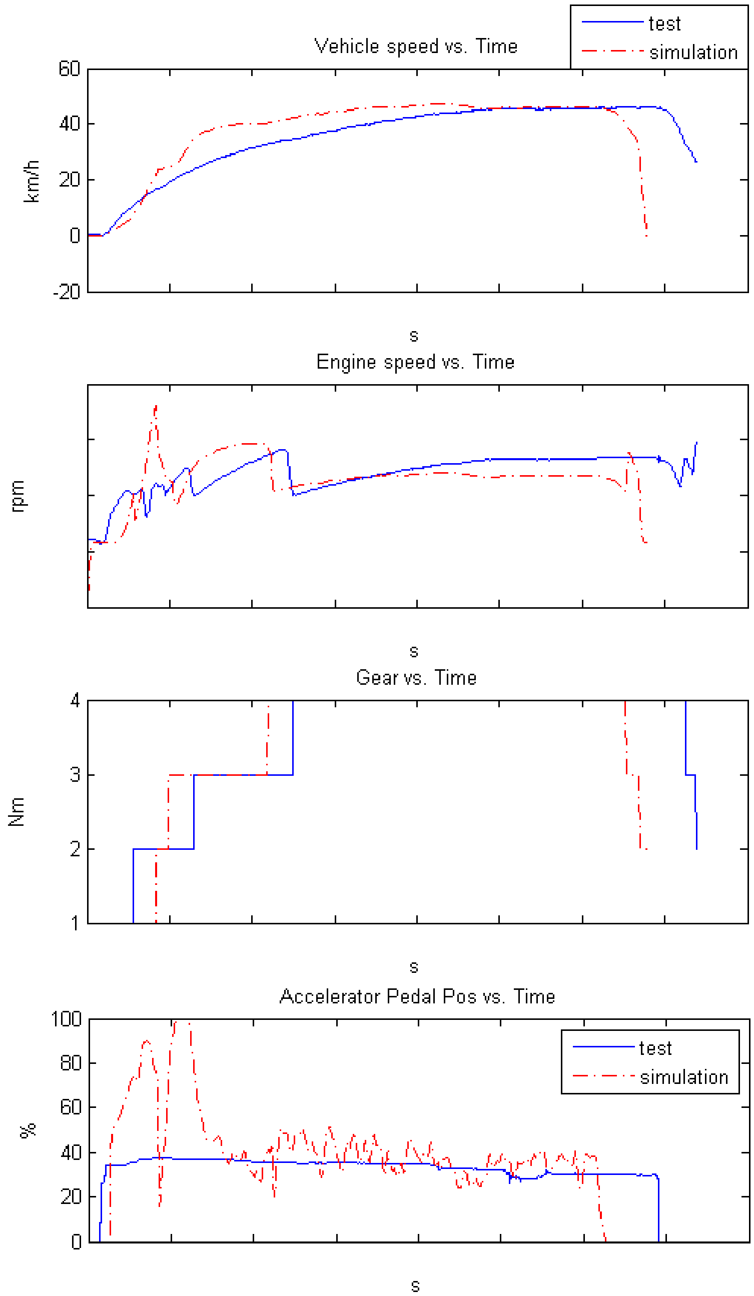

Figure 5.6: Results from test cycle 4, with 30% throttle.

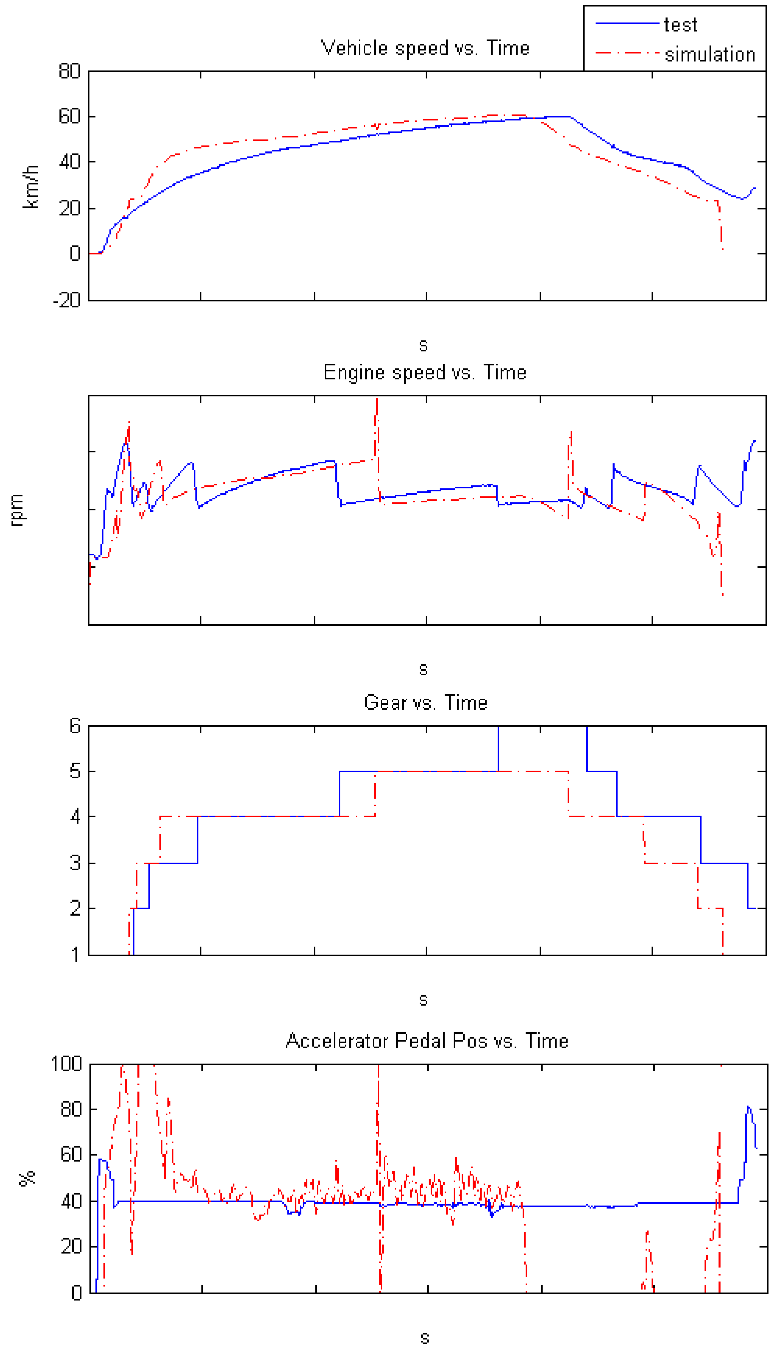

Figure 5.7: Results from test cycle 4, with 40% throttle.

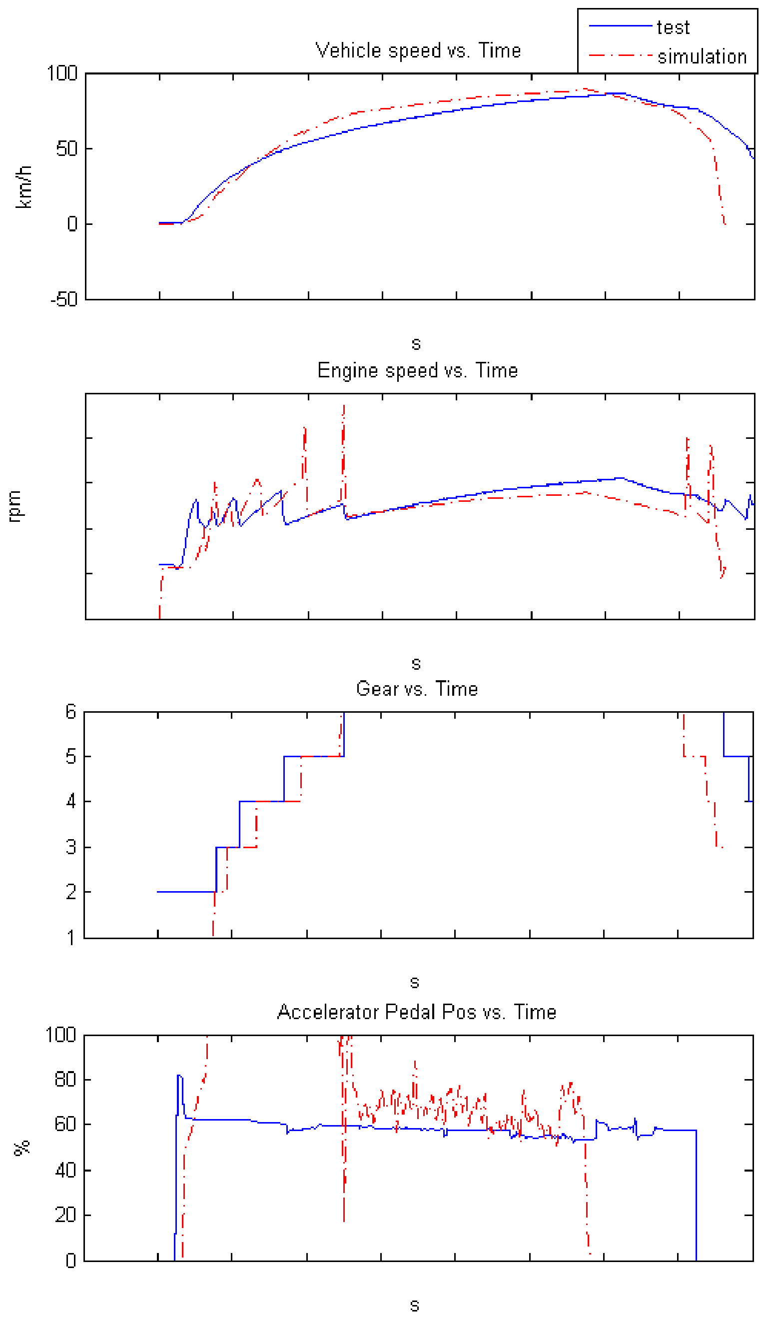

Figure 5.8: Results from test cycle 4, with 60% throttle.

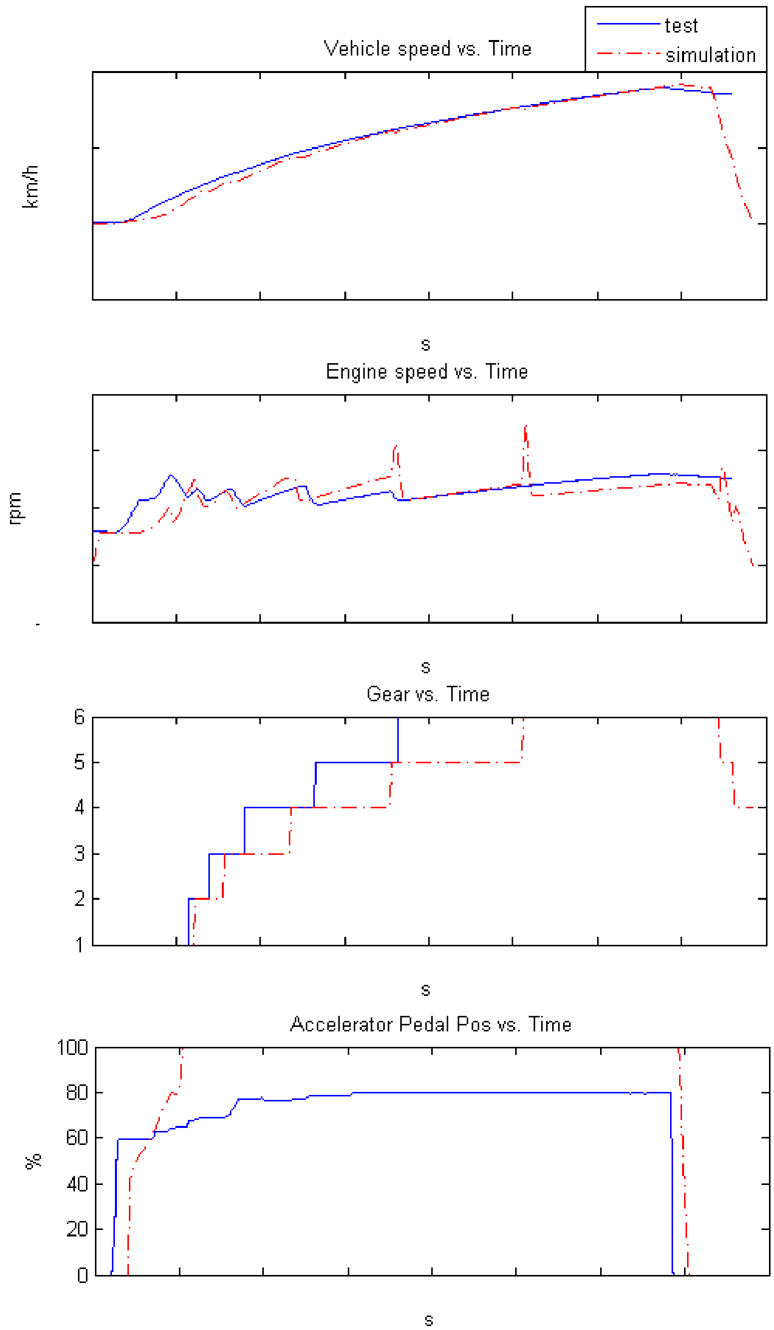

Figure 5.9: Results from test cycle 4, with 80% throttle

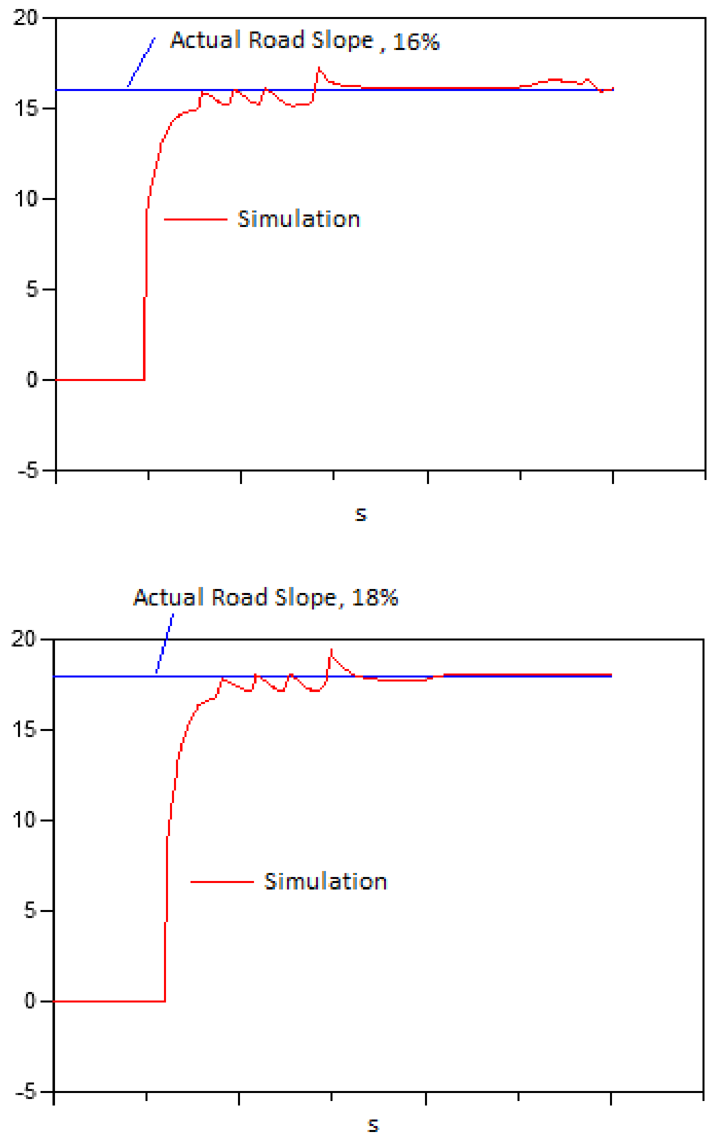

Figure 5.10: Road slope calculation results from test cycle 5.

Chapter 6

Discussion & Conclusions

The gearbox model shows a lot of promising results. This chapter contains discussion and concluding remarks about the implemented model's performance in retrospect to the thesis's objectives.

6.1 Fuel Consumption

Table 5.2 provide very little data to base any conclusions on. Generally, it seems that the simulations results in lower fuel consumption then the real vehicle. The difference between the two results are 5.3 l/100km less for SORT 1 and 5.9 l/100km less for SORT 2. Relatively the error between the simulated and measured value is $\sim 11\%$ less for SORT 1 and $\sim 14\%$ for SORT 2.

There are simplifications made in the gearbox model, which could suggest that the simulations should produce lower numbers for fuel consumption. One is that there are no *auxiliary* systems such as air conditioner or heater etc. stealing power from the engine. Together with the relative error being approximately the same for both tests. It leads to the conclusion that the model can be used to make good fuel consumption estimates, even though there is so little data, if the relative error is taken into account.

6.2 Shift Behaviour

This section is dedicated to the discussion of the overall shift behaviour of the gearbox model. The shift points in tests 1-4 are discussed in relation to vehicle speed, engine speed and accelerator pedal position.

Comparing the vehicle speed to the gear plot, in Fig. 5.1, a certain pattern of three ups and down can be seen. In these plots the closer that the vehicle speeds curves are, the closer the gear curves are as well. The engine speed curves are also close to overlapping each other, the few times that they are far apart can be explained by the different gears in that particular moment.

Since the curves for the *vehicle speed* do not overlap perfectly the test cycle is not perfectly reproduced in the simulation. This means that the exact same gear plot *should not* be reproduced either. Overall the simulation has produced a satisfying shifting behaviour that is similar to reality.

Figures 5.2 and 5.3 shows two attempt of the same test cycle, but with different simulation results. In Fig. 5.3 the fourth gear is engaged were in the other attempt it is not. The reason is a higher vehicle speed in that simulation. Whereas for test cycle 3 the opposite situation occurs, there the real vehicle skips a gear in the first attempt, see Fig. 5.4. While in the second attempt, Fig. 5.5, the real test and simulation reproduces more similar results. Especially the shift down from $5 \rightarrow 4$ that occurs at similar engine speeds in Fig. 5.5.

This shows that for just the small difference that occurs between two attempts of the same test cycle can produce some different results when recreating the attempts in simulations. Just look at the engine speed curve in Fig. 5.4.

In test cycle 4, different accelerator pedal positions were used. This resulted in overall good results from the simulations. Fig. 5.6 is pretty straight forward, again there is some anomaly in the gear plot which can be traced back to the vehicle speed. The engine speed goes really high just before shifting to third gear though, almost like a spike. Usually such a high engine speed is not wanted. But this sharp edge also occurs in Fig. 5.7, for both the real test and the simulation. Because of the real vehicle showing this behaviour it is safe to say that there is nothing wrong with the model in this aspect.

The shift $5 \rightarrow 6$ occurs simultaneous in Fig. 5.8. Looking at the engine speed curves for the same situation they overlap as well, meaning that the shift happens at the same engine speeds and this is satisfying.

Figure 5.9 shows the simulation where the vehicle speed curves overlap the closest. Also, the engine speed curves are coinciding in the later shifts in this case. The later shift points can be explained by the accelerator pedal. In the simulation the driver uses full throttle which results in higher engine speed for shifting, effecting the offset β.

6.3 Torque Converter Lock-up Clutch

The locking mechanism of the torque converter shows satisfying results in the figures. Especially in Fig. 5.1 when during one instance both the real test and the simulation unlocks for a quick second to lock again the next, the interval marked A in Fig. 5.1. Because the simulation is lagging a bit behind, in terms of vehicle speed, this motivates why the LuC in the simulation is a bit slower to both unlock and lock.

Similarly, in figures 5.2 and 5.3 the simulated LuC locks after the real test, but this time it unlocks before. The explanation for this lies in the vehicle speed, in both two figures the simulated vehicle is accelerating after and braking before the real one. Overall this leads to the conclusion that the model for locking and unlocking the torque converter works.

6.4 Road Slope Calculation

The road slope calculations in Fig. 5.10 show that the road slope can be estimated using the vehicle's equation of motion Eq. (3.17). The small notches in the calculated road slope originate from torque spikes that shoots through the gearbox sometimes. The torque spikes are on the order of 10^3 times greater than the normal torque load, and it really pulled a number on the calculations for the road slope. To combat the effect the variable for the road slope is runned through a filter that dampens it, see Appendix C. Despite this, the calculation for the road slope conforms well with the real value.

6.5 Future Work

The following list contains suggestions for improvement and continued development of the work in this thesis.

- In chapter 6.1 it is mentioned that the simulated fuel consumption has a relative error. With more data and analysis this error could possibly be better determined.
- The model was validated together with other drivetrain components. But it could be validated on its own. This could be done by controlling the load and torque on both sides and feeding the correct engine speed to the gearbox.
- Validation the torque converters amplification ratio.
- Implement the *disengaging the drivetrain* feature.
- Including the inertial acceleration resistance F_J in the force calculations.
- In the simulations the engine speed spikes for the shifts $4 \rightarrow 5$, $5 \rightarrow 6$ and back. Before the *torque reduction*, section 4.2.2, was introduced this actually happened during all shifts. Refining the torque reduction may result in removing the speed spikes.

6.6 Final Words

The biggest problem that was encountered during the thesis is that not much information available publicly on the area of *shifting strategies* can be found in articles or books. Most work dates back to the eighties, the article from Cho and Hendrick et al. [2] and the thesis from Runde et al. [10], etc. This could be because of the difficulty of the subject.

In the section about *road slope calculation*, it is mentioned that the variable is runned through a filter. This is not the only variable that is subjected to this. Almost all variables that can be viewed as a *step*-function is runned through a filter, see Appendix C for details.

It seems as if the solver in Modelica have difficulties handling step functions. Causing either the simulation to crash or giving huge spikes[1] to other variables. Could be due to Modelica being an equation based language. If one variable in a Modelica equation has a sudden increase the other variables in the same equation will be instantaneously effected. But it could also be that a step function which is a type of *stiff* function and these typically gives problems for numerical solvers.

In addition to this it may be said that the simulations are done rather quickly. Just a few minutes for the test cycles that were examined. It depends some on how much data the user chooses to save for post processing. Since this has to be save to the hard drive of the computer in order to not run out of memory.

Collectively, the thesis is a success. The model is behaving as can be expected from a real automatic gearbox, with correct shift behaviour and converter locking. As a side effect to making the model as general as possible there are variables and functions that require calibration before use.

[1] Spike is referring to an quick but temporary increase/decrease in value.

Bibliography

[1] Robert Bosch. *Automotive Handbook*. Bentley Publishers, 8th edition, May 2011.

[2] Hendrick J.K. Cho D. Automotive powertrain modeling for control. *Transactions of the ASME*, 111:568–576, December 1989.

[3] Dynasim. *Dymola Multi-Engineering modeling and Simulation*, version 5.3 edition, 2004.

[4] Peter Fritzson. *Modeling and Simulation of Technical and Physical Systems with Modelica*. Wiley, 2011.

[5] Wolfgang MEYER Hans RAT. Sort - standardized on-road test cycles. Technical report, UITP - International Association of Public Transport, 2004.

[6] Torkel Glad Lennart Ljung. *Modellbygge och Simulering*. Studentlitteratur, 2004.

[7] Felix Felgner Liu Liu and Georg Frey. Comparison of 4 numerical solvers for stiff and hybrid systems simulation. Technical report, Chair of Automation, Saarland University, September 2010.

[8] Otter M. Pelchen C., Schweiger C. Modeling and simulating the efficiency of gearboxes and of planetary gearboxes. In *2nd International Modelica Conference, Proceedings*, pages 257–266, `https://www.modelica.org/events/Conference2002/papers/p33_Pelchen.pdf`, March.

[9] Rajesh Rajamani. *Vehicle Dynamics and Control*. Springer, 1st edition, October 2005.

[10] J. Runde. Modelling and control of an automatic transmission. S.m.m.e. thesis, Department of Mechanical Engineering, M.I.T., 1986.

[11] Michael M. Tiller. *Indtroduction to Physical Modeling with Modelica*. Kluwer Academic Publishers, 2001.

[12] Tomas Zackrisson. Modeling and simulation of a driveline with an automatic gearbox. Master's thesis, Royal Institute of Technology, KTH, December 2003.

[13] ZF Friedrichshafen. *ZF-Ecomat 4, Technical Manual for City Buses, Intercity Buses and Coaches*, April 2006.

[14] ZF Friedrichshafen. *ZF-Ecolife, Technical Manual for City Buses, Intercity Buses and Coaches*, May 2011.

Appendix A

Notations

Table A.1: Notations of used quantities and their units.

Quantity		Unit
F	Force	N
T	Torque	Nm
P	Power	J/s
J	Mass moment of inertia	kg·m^2
R	Wheel radius	m
ω	Rotational Speed	rad/s
α	Rotational Acceleration	rad/s^2
r	gear ratio	-
λ	retarder ratio	-
ρ	density	kg/m^3
Z	Number of cogs	-
v	Speed	m/s
a	Acceleration	m/s^2
m	mass	kg
t	time	s
g	gravitational acceleration	m/s^2
η	Efficiency	-
σ	Performance index	-
ν	Speed ratio	-
μ	Torque ratio	-
τ	Shift time	s
θ	Gradient angle	rad
A	Frontal area	m^2
D	Diameter	m
C	Coefficient of rolling resistance	-
c	Aerodynamic Drag coefficient	-
p	Position	-
β	percent change	-

Table A.2: Subscripts and other symbols used in the report

Subscript	Full name
fluid	hydraulic fluid
eng	engine
wheel	Roadwheel
GB	Gearbox
FD	Final drive
ret	Retarder
prop	Propshaft
tot	Total
max	Maximum
min	Minimum
P	Impeller (Pump), torque converter component
T	Turbine, torque converter component
i	Gear number
fin	Stop time
idle	The lowest speed an engine can have without failing
coup	Coupling
req	Request
D	Drive
J	Inertial
off	Offset
acc	Accelerator pedal
loss	Losses
rel	Relative
$\hat{x}$	"hat" denotes an estimate, here estimator of x
$\uparrow$	Shift limit to higher gear
$\downarrow$	Shift limit to lower gear
$\dot{x}$	"dot" denotes time derivative

Appendix B

Derivations

Consider the gear shift from $i \rightarrow i'$. The relation between engine and propshaft speed is:

$$\frac{\omega_{eng}}{r_i} = \omega_{prop} \tag{B.1}$$

and like wise after the shift is done the engine speed is

$$\frac{\omega\prime_{eng}}{r_{i'}} = \omega\prime_{prop} \tag{B.2}$$

Since shifting is a quick event then $\omega_{prop} \approx \omega_{prop}\prime$, this relation can be used together with Eq. (B.1) and (B.2) to formulate an equation for the estimate $\omega_{\hat{eng}}\prime$ after the shift is done.

$$\hat{\omega}\prime_{eng} = \frac{r_{i'}}{r_i}\omega_{eng} \tag{B.3}$$

The following relation can then be stated:

$$\hat{\omega}_{eng,i'} \begin{cases} < \omega_{eng,i} & \text{if } r_{i'} < r_i \\ > \omega_{eng,i} & \text{if } r_{i'} > r_i \end{cases} \tag{B.4}$$

Appendix C

Filters

To dampen a step function a filter can be used. Let $u(t)$ be a step function, then formulate the scheme for the damped function $y(t)$, in an iteration scheme, to be:

$$y_{i+1} = y_i + \frac{u_i - y_i}{T} dt \tag{C.1}$$

Where, T is a time constant that determines how long it takes for the damped signal to reach the value of u. $T < dt$ will lead to y over shooting u. Filters can also be used for dampening other signal then step functions.

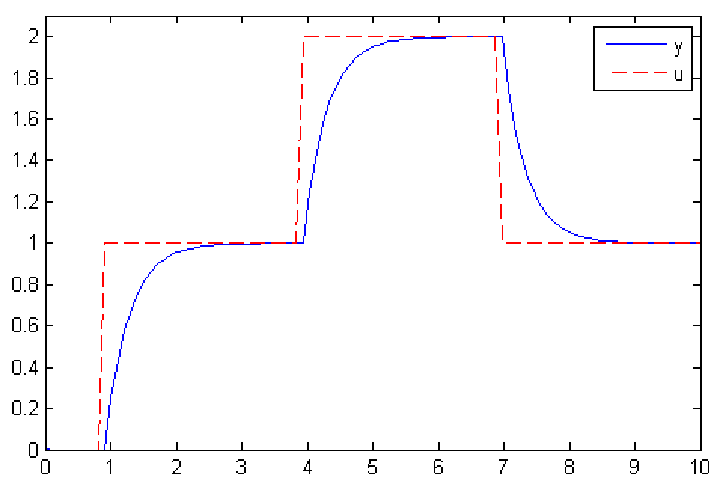

Figure C.1: Sketch of a filtered signal.