CHALMERS
UNIVERSITY OF TECHNOLOGY

Modelling of a rear axle Torque Vectoring Dual Clutch driveline for battery electric vehicles with verification in a virtual environment

Master's thesis in Automotive engineering

CHIRAG VASANTH JAIN

DEPARTMENT OF MECHANICS AND
MARITIME SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

MASTER'S THESIS IN AUTOMOTIVE ENGINEERING

Modelling of a rear axle Torque Vectoring Dual Clutch driveline for battery electric vehicles with verification in a virtual environment

CHIRAG VASANTH JAIN

Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2021

Modelling of a rear axle Torque Vectoring Dual Clutch driveline for battery electric vehicles with verification in a virtual environment
CHIRAG VASANTH JAIN

Master's thesis 2021:32
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Supervisor: Jan Andersson, Electric Driveline Dimensioning, Volvo Cars Corporation
Examiner: Bengt J H Jacobson, Vehicle Dynamics group, Division of Vehicle Engineering and Autonomous Systems, Department of Mechanics and Maritime Sciences

Cover: Cornering maneuver simulation on IPG CarMaker simulation platform
Chalmers Reproservice
Göteborg, Sweden 2021

Modelling of a rear axle Torque Vectoring Dual Clutch driveline for battery electric vehicles with verification in a virtual environment
Master's thesis in Automotive Engineering
CHIRAG VASANTH JAIN
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology

Abstract

Vehicle manufacturers depend on modelling and simulation in a virtual environment while developing new automotive systems, as it saves both time and costs. These models allow rapid changes to be made during the development of a system. As electrification gains momentum amongst vehicle manufacturers, consumers and the authorities, new transmission concepts are being explored. This report explores the method of physical and acausal modelling for building a simulatable model of one such concept namely, a rear axle Torque Vectoring Dual Clutch (TVDC) system for a battery electric vehicle. The TVDC model is also integrated with the complete vehicle model to conduct complete vehicle verification in a simulation environment. The objective is to model the system physically using the Modelica language and export it as a Functional Mock-up Unit into a virtual vehicle environment on IPG CarMaker. The model is then integrated with a control model and a full vehicle model. Modelling approaches using standard Modelica libraries and 'flat' or equation-based modelling are explored and compared. The model is then validated by running simulations on standard test tracks and observing the performance of the system. The torque vectoring function will be the main focus and the drive torques to the rear wheels from the TVDC driveline will be studied to ensure that they have the desired behaviour. The model is further validated by comparing the simulation results with measurement data from a test vehicle with the TVDC system installed, for the same test runs and ensuring that the same kind of control signals results in similar behaviour in drive torque distribution on the rear axle. Additionally, an open differential vehicle model is simulated to establish comparisons with the TVDC driveline in terms of power consumption and steering effort.

The equation-based modelling approach was found to be more flexible in terms of modelling preconceived systems in contrast with library-based modelling that though easier, is restrictive in connecting certain component blocks together. On running the necessary simulations, it was concluded that the driveline model was robust in the sense that the system did not fail in simulations involving dynamic driving scenarios and long driving cycles such as the Göteborg City Cycle. The model demonstrated its ability to be controlled, by exhibiting the behaviour desired by the devised control strategy. Comparison with track measurement data from a test vehicle fitted with the TVDC driveline showed that the same kind of control signals in the simulation and measurements resulted in the same kind of torque distribution at the rear axle and this is a successful validation of the model's performance in steady-state driving scenarios. It was also concluded that information on the control strategy used in the test vehicle's driveline would be useful in validating the model performance further in dynamic driving situations. Comparison of the TVDC driveline model with the open differential model did not show a significant difference in energy consumption or steering effort. However since the focus was on modelling and model validation, no substantial conclusions were drawn as the control strategy used in the TVDC system was basic and was probably insufficient for the sake of comparison.

It can be concluded that physical modelling in Modelica is a reliable way to develop driveline system models for simulation. This physical modelling method can be an effective early 'virtual development' tool for driveline concepts and as such is verified by the comparisons with real vehicle tests.

Keywords: Electric propulsion, Modelling, Modelica, Simulation, Driveline, Torque vectoring, Clutch capacity, Control, TVDC, Measurement, Passenger cars

Acknowledgements

I would like to extend my gratitude to my supervisor Jan Andersson from Volvo Cars for giving me this opportunity to work on this thesis and for supporting me at every step. My sincere thanks to Bengt J H Jacobson, my examiner at Chalmers University for all the help he has extended to me while working on this thesis. My supervisor's and my examiner's expertise in the modelling subject along with their ideas ensured that the thesis work progressed in the right direction and helped me surpass all hurdles. I am also thankful to the colleagues in the Electric Driveline Dimensioning team at Volvo Cars for supporting me with their technical expertise and taking time off their regular work to help me out and make me feel part of the team.

I would like to thank my family for their constant love and support that kept me going. I am grateful to my friends Aditya S Murthy and Raghavendra R G for all the 'think-tank' sessions we had while doing this thesis work. Finally, I would like to thank my classmates and faculty at Chalmers University, my friends and anyone who has had the slightest impact on this thesis work.

Chirag Vasanth Jain
Göteborg, June 2021

Nomenclature

T	Torque in Nm
w or ω	Rotational speed in rad/s
J or I	Moment of inertia in kgm^2
c	Spring constant in Nm/rad
F_z	Normal load or weight in N

Abbreviations

4WD	Four Wheel Drive
AWD	All Wheel Drive
BEV	Battery Electric Vehicle
CM	CarMaker
ICE	Internal Combustion Engine
ESC	Electronic Stability Control
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
GCC	Göteborg City Cycle
GUI	Graphical User Interface
RL	Rear Left
RR	Rear Right
RWD	Rear Wheel Drive
TC	Traction Control
TVDC	Torque Vectoring Dual (or Double) Clutch
UAQ	User Accessible Quantity
VCC	Volvo Cars Corporation
VVE	Virtual Vehicle Environment

Contents

Abstract i

Acknowledgements iii

Nomenclature v

Abbreviations v

Contents vii

1 Introduction 1
1.1 Background . . . 1
1.2 Problem motivating the project . . . 1
1.3 Envisioned solution . . . 1
1.4 Objectives . . . 2
1.5 Deliverables . . . 2
1.6 Limitations . . . 2
1.7 Method . . . 2

2 Theory 4
2.1 Literature study . . . 4
2.2 TVDC system overview . . . 4
2.3 Choice of modelling tool . . . 5
2.3.1 Signal-based modelling . . . 5
2.3.2 Physical modelling . . . 5
2.4 Tools used in the thesis work . . . 5
2.4.1 Modelica/Dymola . . . 5
2.4.2 Functional Mock-up Interface . . . 6
2.4.3 IPG CarMaker . . . 6
2.5 Torque and speed causality . . . 7
2.6 Brief description of clutches . . . 8
2.7 Torque Vectoring . . . 8

3 Approach and implementation 9
3.1 TVDC driveline interface . . . 9
3.2 Modelica library-based model . . . 10
3.2.1 Component Description . . . 11
3.2.2 Acausal connectors . . . 13
3.2.3 Interfaces . . . 14
3.2.4 Sources . . . 14
3.2.5 Sensors . . . 14
3.2.6 TVDC model using libraries . . . 15
3.2.7 Model Performance . . . 16
3.2.8 Model Update . . . 16
3.3 Modelica equation-based model . . . 17
3.3.1 Exporting model as an FMU . . . 19
3.4 Driveline model integration on CarMaker . . . 20
3.4.1 Integration on CM 'generic' model . . . 22
3.4.2 Integration on VCC CarMaker model . . . 23
3.5 TVDC controller . . . 25
3.5.1 Controller A . . . 25
3.5.2 Controller B . . . 25
3.5.3 Controller C . . . 27
3.6 Simulation . . . 28

3.6.1 TVDC Driveline test simulation . . . 28
3.6.2 TVDC model validation by comparison with vehicle measurement data . . . 28
3.6.3 Comparison of TVDC and open differential performance in simulation environment . . . 28

4 Results **29**
4.1 Results from the TVDC Driveline test simulation . . . 29
4.2 TVDC model validation by comparison with vehicle measurement data . . . 31
4.2.1 Figure of eight track . . . 31
4.2.2 Handling track 1 . . . 37
4.2.3 Handling track 2 . . . 40
4.3 Results from the comparison of TVDC system and open differential in simulation environment . . . 41
4.3.1 Figure of eight track . . . 41
4.3.2 Handling Track 1 . . . 42
4.4 Update on library model . . . 43

5 Conclusions and future work **45**
5.1 Conclusions . . . 45
5.2 Future work . . . 45

References **47**

A Appendix **48**

1 Introduction

The automotive industry is welcoming a whole new era of electrification in vehicles. Passenger cars are a major focus of electrification as major cities around the world are investing heavily in building the necessary infrastructure that can sustain the needs of electrification within mobility. Volvo Cars Corporation is also a part of this massive transformation and is already pushing forward sustainable mobility by introducing its own Battery Electric Vehicles (BEVs) into the market. Electrification is a completely new concept and hence it brings in major changes in how vehicle subsystems are built around it. Drivelines from conventional ICE cars cannot be directly implemented in BEVs, they are subject to redesign and modifications. New concepts in electric drivelines need to be tested for their performance, initially in a simulation environment, where they can undergo rapid modelling changes in the initial stages of development. Numerous tools will be used for model development and simulations, hence the tools need to allow easy exchange of dynamic simulation models. The models have to be accurate in replicating the real system in terms of performance and must encapsulate the physics of the various components and the system as a whole. This brings in the need for physical modelling that will be crucial for developing new concepts in electric drivelines.

1.1 Background

To meet future regulations on CO_2 emissions and energy consumption, Volvo Cars Corporation (VCC) has taken the decision to focus on the technology change going from ICE (Internal Combustion Engine) driven vehicles to BEVs (Battery Electric Vehicles). Due to cost and time efficiency, the dependency on physical testing needs to be reduced. Hence modelling and simulation in a virtual environment is preferred since it offers greater flexibility in tuning the system for the required performance. To achieve reliable results from computer models, accurate models are needed. Existing models of the drivelines cannot be modified to be used for the new driveline concepts, so new models need to be developed and validated. Control systems need to be designed for the new driveline systems, as they might consist of components that are not used in conventional drivelines.

The first wave of electric vehicles has used conventional drivelines in their powertrains. All Wheel Drive (AWD) BEVs like the Volvo XC40 Recharge use open differentials on the rear axle to split the torques going to the wheels. Some BEVs have simplified drivelines, due to electric motors housed in the wheel hub, with a constant ratio gearbox, where the motors can be controlled individually to deliver the required power. For the next generation of Volvo BEVs, the electric propulsion system (MEP2) is planned to be equipped with a new torque vectoring system on the rear axle drive unit. This torque vectoring system is called the TVDC (Torque Vectoring Dual Clutch) driveline. The new system needs to be modelled in order to create reliable simulation results.

1.2 Problem motivating the project

The existing open differentials in the BEVs can only distribute the torques between the two wheels on the rear axle equally. There are driving scenarios, for instance, cornering or driving on surfaces with varying coefficient of friction, where it is beneficial to be able to distribute the torques to the wheels in any required ratio. An open differential is not sufficient in such cases; a TVDC system provides the necessary functionality. This thesis work aims at developing an electric driveline model for the TVDC system. The system will be controlled by two clutches, creating the torque vectoring function on the rear axle. The new system needs to be modelled and controlled in order to ensure that the system is reliable. This work is essential for creating requirements while designing components and ensuring that the right components are selected and dimensioned for the right usage. Information about the system from the supplier needs to be taken into the model. This is essential for creating functional, durability and efficiency requirements for the components in the driveline. These models need to be evaluated in pure simulation environments and later on in real vehicles and/or test rigs.

1.3 Envisioned solution

This thesis work intends to physically model an electric driveline system with the right interfaces so that it can be implemented in a simulation environment in a form that is easy to exchange between the modelling

and simulation environments. In order to develop an electric driveline model for the TVDC system, different modelling approaches are to be considered, and the advantages and drawbacks of the modelling approaches are to be studied. The physical modelling will be carried out on Dymola, which uses the Modelica language. On successful validation of the model, it needs to be integrated into a simulation environment with a complete vehicle model. IPG CarMaker consists of a complete vehicle model, in which a user-defined driveline model can be imported. FMI (Functional Mock-up Interface) provides a standardized format that enables smooth model exchange between tools, and since both Dymola and CarMaker support FMI models, it is chosen as the format for the physical model. A control model is to be developed for driving scenarios on CarMaker, which gives plausible results. The next course of action would be to work on comparing the results obtained from the model with real vehicle tests, for the purpose of model validation.

1.4 Objectives

- Develop a Modelica model of the TVDC (Torque Vectoring Dual Clutch) system, including the control interface for the clutch actuator.
- Export the TVDC model as an FMU (Functional Mock-up Unit) to IPG CarMaker, add clutch control and vehicle level control.
- Test Run in CarMaker with complete vehicle model and the TVDC electric driveline.
- Validate the model with vehicle measurement data from the track.

1.5 Deliverables

- Modelica model of the TVDC driveline and an FMU of the model, with interfaces to the driveline control and vehicle model on CarMaker.
- Vehicle model and driveline control with interfaces to the driveline FMU.
- Complete vehicle model with the above two deliverables integrated.
- Validation of the model behaviour with vehicle measurements on track.
- Comparison of performance between open differential and TVDC drivelines, in terms of energy consumption.

1.6 Limitations

The thesis work makes the following assumptions in modelling the TVDC system and in its simulation-

- Vehicle velocity estimation is assumed to be ideal.
- Steering wheel angle is assumed to be considered taking into account the system compliances.
- Load on the rear axle and respective wheels is assumed to be measured or estimated.
- Clutch controller is developed to get the same kind of behaviour in the clutches as the vehicle measurements, but there is limited knowledge of the clutch control logic used in the actual vehicle.

1.7 Method

The thesis work involved modelling, simulation, control system design, model validation and verification of the results derived out of the developed models. Modelling and control system design was carried out on different tools like Dymola and IPG Carmaker and the simulation results obtained were verified by comparison with the actual vehicle log data. The approach for carrying out this thesis work is enlisted below-

- A detailed literature study was carried out to understand how modelling is performed in the Modelica tool (Dymola), FMU generation and integration with CarMaker standard models. The literature review also focused on understanding basic torque vectoring control algorithms and the physics of clutches.
- Models were built in Dymola using equations and standard libraries (blocks) to understand how robust, simulatable models could be set up. This step aimed at getting familiar with the Dymola modelling environment and working with debugging model errors that might emerge as the model complexity increases.
- Models were built in Dymola for the TVDC system using standard libraries and equations, and the two modelling approaches were compared for feasibility, ease of simulation and compatibility on CarMaker. The models were exported as FMUs to CarMaker.
- A vehicle model was configured for simulation on CarMaker using the TVDC Driveline FMU from Dymola and to test the model in different driving scenarios. A control system was designed for actuating the clutches based on functional requirements for the manoeuvres. The system was tested to ensure it was robust and gave desirable results.
- The tested driveline model was integrated including its control system with the Volvo Cars CarMaker environment for simulations on standard test tracks.
- The simulation results were validated with the actual vehicle measurement data.
- The TVDC driveline model was compared with the open differential model for power consumption and steering effort (steering angle required), for the same driving scenarios in CM simulations.

2 Theory

2.1 Literature study

The primary objective of the literature study was to understand modelling using standard Modelica libraries in Dymola. For the electric driveline model, components from the 'Rotational' library were predominantly used. There are several simulatable examples in the 'User's Guide' section of the 'Rotational' libraries [1]. These examples provide an overview of how the components can be connected to each other, the use of interface signals and how they differ from physical quantities. The characteristics of rotational components such as clutches, inertias, stiffness elements and sensors used to measure physical quantities like torques and speeds, were studied using the Modelica examples and library documentation. The modelling of a clutch in Modelica including its slip is explained in Modelica by Example [2] and also in the Xogeny Blog post on Modelica [3]. Additionally, the modelling approach using equations on Modelica was also explored in order to have versatility in the modelling process.

Hanjie Xu's work [4] on electric steering models in Dymola proved to be insightful in understanding the different kinds of Modelica interfaces that can be used, in order to have interchangeable models using an FMU on IPG CarMaker. Bhat and Malghan's master's thesis work [5] on propulsion system modelling was invaluable as it involved modelling of a propulsion system and integration in the CarMaker environment using Matlab/Simulink. As the driveline model requires a basic level of control in order to exhibit some desired behaviour, a study of different torque vectoring algorithms was carried out. Anton Stoop's thesis [6] demonstrated a torque vectoring algorithm for an RWD vehicle. Aldo Schaaf's master's thesis [7] also explained concepts that could be used to devise a torque allocation logic for a 4WD electric vehicle. Grahovic and Rosicki's thesis [8] involved controlling the torque vectoring function of a TVDC system already installed in a Formula Student car, and this was useful in understanding the TVDC system.

2.2 TVDC system overview

Torque vectoring is a system by means of which the torque supplied to each of the driven wheels in a vehicle can be controlled individually, i.e. it is a torque distribution system. A torque vectoring with dual clutch (TVDC) system is a driveline for the rear wheels in a BEV, that can be used to achieve a torque vectoring function by independently controlling the engagement of the two clutches. Each of these clutches is connected to a rear wheel and the level of engagement dictates the power supplied to the wheel. The schematic of a rear axle TVDC driveline unit is shown in figure 2.1.

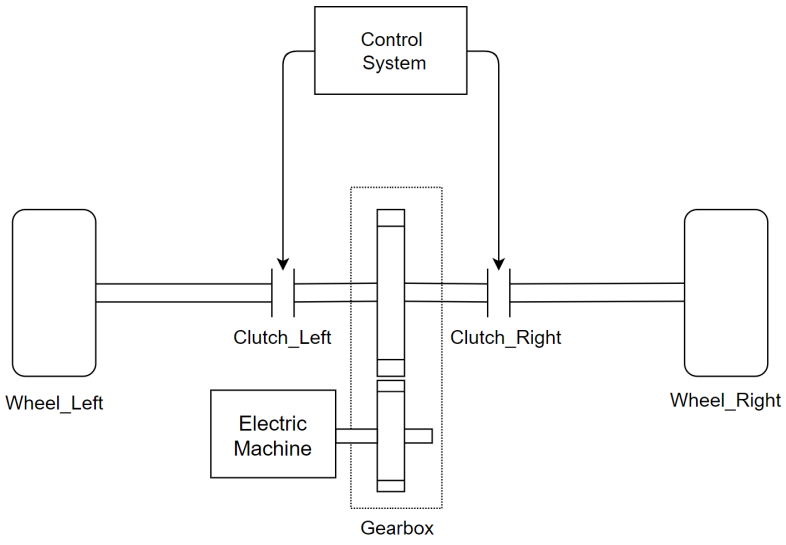

Figure 2.1: Schematic of a rear axle TVDC system

The TVDC driveline is part of the rear powertrain which consists of a single electric machine/motor as the power source. This system consists of components between the electric motor and the wheels. The output shaft of the rear electric motor is connected to a single-ratio gearbox which in turn has two output shafts connected to the two rear wheels through a clutch for each wheel. A control system is used to control each of the clutches independently in order to achieve the torque vectoring function.

2.3 Choice of modelling tool

Modelling a system involves choosing the right platform which is convenient for the modelling approach chosen by the user. Modelling approach classifications that are comparable against each other are signal-based modelling and physical modelling.

2.3.1 Signal-based modelling

Signal-based modelling is a popular modelling approach employed by tools like Simulink. In this approach, the modelling is carried out in a sequence of blocks, where the sequence is predefined and information is exchanged between the blocks in the form of signals. Such a modelling approach is convenient for modelling control systems, where the information flow direction of the control signals follows a particular sequence. This approach also uses explicit or causal modelling where the equations are arranged in the form of assignment statements for the unknowns [9]. The order of solving these assignment statements is fixed. The causality of each of the interface variables is fixed to either input or output [10].

2.3.2 Physical modelling

Physical modelling of a system is a modelling approach similar to the structure of the actual system. In this modelling approach, the different components that the system is comprised of are 'assembled' together to build the model using component blocks with connectors that do not have a causality (direction) or by listing the equations defining the physics of the system. Acausal modelling is employed in this approach, where the equations governing the relation between different components in the system are listed out, such that they are balanced (number of equations is equal to the number of unknowns/variables) and these equations solely describe the relation between the unknowns and do not indicate how these unknowns are evaluated. Hence there is no causality assigned to the interface variables or any sequence defined for solving them.

Acausal modelling makes it easier to reuse component models, which is necessary for the physical plant modelling of an engineering system [1]. On the other hand, causal/explicit models need to be redesigned when a small change is made to the system structure. An explicit model would need to reassign the statements in order to solve the system of equations for a different set of outputs or in the case of blocks, the sequence has to be redefined. Modelica supports physical and acausal modelling and hence was chosen as the modelling platform. Modelica also supports modelling for both continuous and discrete dynamics. Discrete dynamics is important to model for a dry friction element, as in the case of the TVDC system studied in the present thesis. The acausal modelling approach makes it easier to build a plant model for the TVDC driveline on Modelica, as the user does not need to bother about how the set of equations are solved, rather focus on building a consistent set of equations defining the relationship between the various components and the Modelica solver takes care of solving for the system outputs.

2.4 Tools used in the thesis work

2.4.1 Modelica/Dymola

Modelica is an object-oriented programming language that consists of libraries for various engineering domains such as mechanical and electrical, and this component-based modelling makes it easy for the user to build physical models. The modelling process can be carried out using standard Modelica libraries/blocks or using equations that define the physics of the system being modelled.

Dymola is a modelling and simulation tool that uses the Modelica language. Dymola provides an interface to build the models and simulate them by providing the required initial conditions. It also enables the user

to visualize the results of the simulations as plots. Dymola 2021x was used for this thesis work in order to physically model a TVDC driveline system with the required interfaces.

2.4.2 Functional Mock-up Interface

Vehicle manufacturers need to work with virtual development (modelling and simulation) in the early phases of a project. Therefore, suppliers have to deliver virtual prototypes of their subsystems, i.e. models for simulation. Functional Mock-up Interface [11] is a standard to exchange models in a tool independent way.

There are numerous modelling tools that are convenient in their methodology for developing different systems. Each of these tools has benefits and drawbacks in modelling a particular system. In such a situation, it would be beneficial to have the flexibility to build models on different platforms based on the ease of modelling and integrate those systems together to run simulations. Functional Mock-up Interface (FMI) is one such standard that enables interfacing between models built on different tools and provides a standardized interface for the models interacting together in a simulation environment. FMI Standard is available in commonly used modelling tools while exporting the models to a different environment. This helps in ensuring compatibility between the different models and efficient interfacing. The Modelica model was exported to the virtual vehicle model environment using FMI and is called a Functional Mock-up Unit (FMU).

The modularity of subsystem models is not perfect, because still, the causality of all interface variables needs to be fixed, so a model in FMU format does not fit in all complete vehicle models. There are proposals for how to solve this, using a standard [12] to encrypt Modelica models so that Modelica models can be exchanged instead of FMUs. This would increase the modularity significantly.

2.4.2.1 Model Exchange and Co-Simulation

FMI supports two formats for exporting FMUs, namely Model Exchange and Co-Simulation. For a Model Exchange FMU, the numerical solver used for evaluating variables in the FMU is not embedded in the model and is utilized from the importing tool. The importing tool decides the time-step for evaluation of the FMUs. In the case of Co-Simulation, the solver is embedded in the FMU by the exporting tool, and the time-step for the solver is pre-determined. The choice of exporting FMU is also dependent on the importing tool, and what works with it. Since Dymola, which was the exporting tool, supported a format with both Model Exchange and Co-Simulation, this was used for the FMU export.

2.4.3 IPG CarMaker

IPG CarMaker is a virtual vehicle testing environment in which the driveline FMU was simulated and tested. IPG CarMaker is a simulation software where models of passenger cars can be tested for different vehicle functions. IPG CarMaker allows the user to configure a vehicle model on its standard vehicle model platform, where the characteristics of subsystems can be parameterized. Certain subsystems can be deactivated in the CarMaker standard model and a user-defined model can be integrated with the rest of the CarMaker model. This allows the user to test new systems on CarMaker. This vehicle model can then be integrated with a driver model which can be modified by the user on CarMaker to define different driving styles. A road or track model can be imported using GPS data for running the simulation or one of CarMaker's standard test tracks can be used. A driving manoeuvre can be defined for driving on the track dynamically to define different driving conditions for different parts of the track. A complete vehicle model, with a road model, driver model and manoeuvre defined is simulatable on CarMaker. CarMaker has several visualization tools such as IPG Control to visualize the results of the simulations. The user-defined quantities in CarMaker provide an extensive set of variables that can be tracked in a vehicle simulation. IPG Movie also enables the user to visualize the vehicle simulation in the virtual environment in real-time (cover image is taken from IPG Movie).

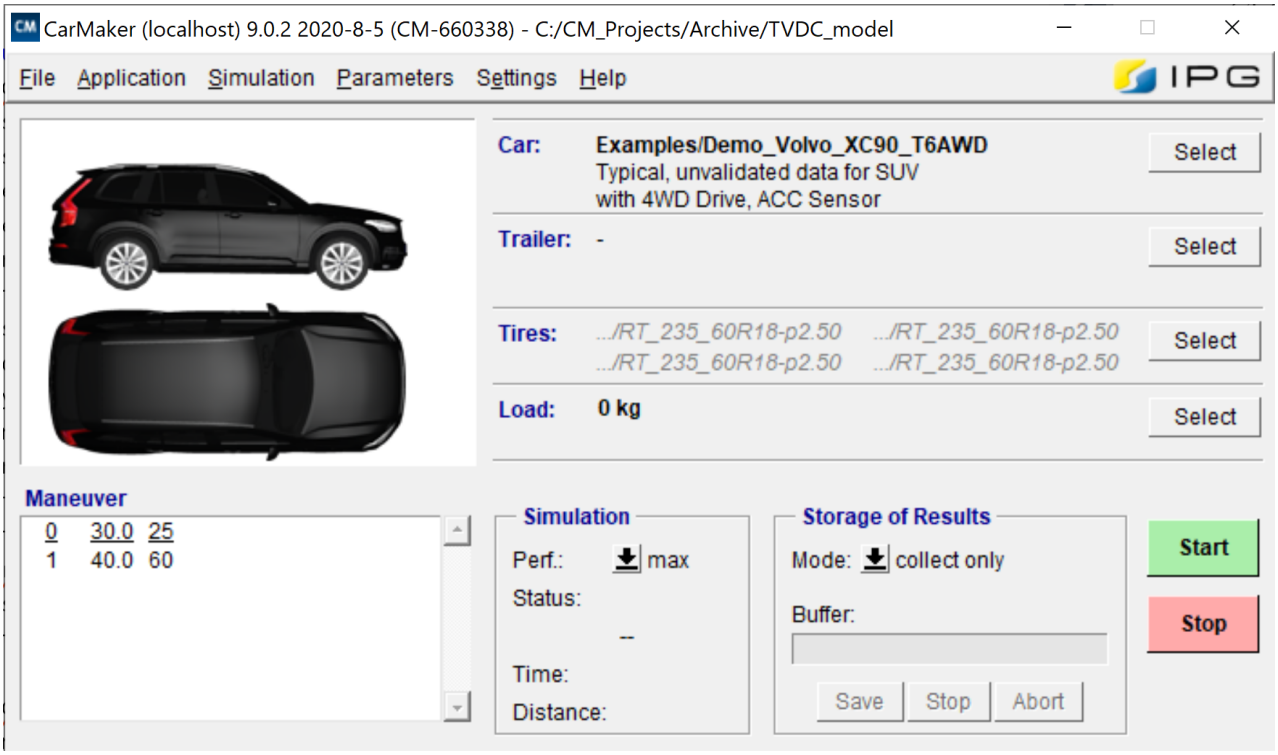

Figure 2.2: CarMaker User interface

IPG CarMaker has an interface with Simulink called 'CarMaker for Simulink' or 'cm4sl' that allows the user to integrate a vehicle subsystem with the right interfaces like the TVDC model to the CarMaker vehicle model through Simulink. The CarMaker user-defined quantities also called 'Dictionary variables' can be modified in the user-defined model using standard CarMaker 'Read CM Dict' and 'Write CM Dict' blocks in Simulink. Additional control modules can be built on Simulink to control the user-defined model, and this gives a lot of flexibility in the modelling process.

2.5 Torque and speed causality

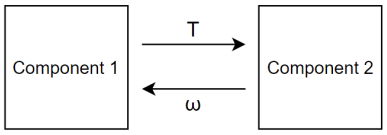

Figure 2.3: Torque and rotational speed causality, showing that if one of them is an input variable to a component, the other is an output variable to the same component.

Torque (T) and rotational speed (ω) have a cause-effect relationship, which implies that one causes the other to manifest. This is an important aspect of modelling a physical driveline model, which will have torques and speeds as interfaces. Torque and rotational speed are results of a factorization of power and represent energy flow. It is important to remember that if torque is sent as an input signal to a system, the rotational speed is expected to be a resultant of the torque and vice-versa. This concept is important in assigning the direction of the torque and speed signals in a driveline model.

2.6 Brief description of clutches

A clutch is a mechanical device that engages and disengages power transmission, especially from a driving shaft to a driven shaft [13]. In clutches the power is transmitted by being completely engaged or closed, a condition referred to as 'sticking' or by being partially engaged, a condition referred to as 'slipping', where some of the power is lost in friction. Also, the clutch can be disengaged, where the clutch is inactive and does not transmit any power between the driving and driven shafts. The maximum torque that the clutch can transmit (in any direction) is referred to as the torque capacity of the clutch. For a disengaged condition, the torque capacity is zero. When the torque supplied to the clutch exceeds its torque capacity, the clutch starts to slip. There are references to 'clutch torque capacity' or 'clutch capacity' throughout this report and they refer to the torque capacity of a clutch.

2.7 Torque Vectoring

For a vehicle in a cornering manoeuvre, the outer wheels (the wheels away from the centre of curvature of the corner) need to travel a greater distance than the inner wheel (the wheels closer to the centre of curvature of the corner), as the length of the arc increases with increase in radius of the arc. The outer wheels in a corner/turn navigate a path with a greater radius (measured from the centre of curvature) when compared to the inner wheels. This also means that the outer wheels need to rotate faster than the inner wheels. This function is mechanically achieved in conventional automobiles using an open differential, where the torques are split equally on an axle, but the outer wheels rotate at a higher speed than the inner.

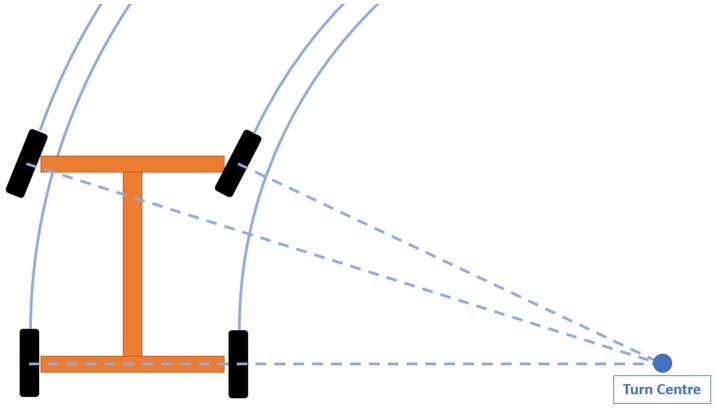

Figure 2.4: Vehicle in a corner shown with inner wheels (right) and outer wheels (left) [14]

In addition to cornering kinematics, the handling characteristics of the vehicle can be improved if the torques can be supplied to the wheels on an axle in unequal distributions as well. The torque vectoring function allows such a variable torque distribution to be realized in an axle. Torque vectoring, not only can improve cornering performance but also can help in ensuring that the wheel with greater traction (grip) receives greater torque, so that 'wheel spin' is avoided. This could be beneficial while driving on surfaces where coefficient of friction varies or is low such as on roads covered in water and snow.

3 Approach and implementation

The approach for modelling a TVDC driveline was to build a physical model that was acausal in nature, which would make the modelling of the system easier. Since the driveline model would be built in Dymola and exported to CarMaker through Simulink, it was important to have the right input and output interfaces in the model in order to achieve efficient model exchange between the two platforms. Modelling was first carried out using standard Modelica libraries, to observe the behaviour of different component blocks working together. This provided an idea about how the blocks could be assembled to build the TVDC system. The next method considered was to model the system using equations. Both these methodologies were evaluated for their functionality and robustness. The final TVDC Modelica model was then exported to Simulink as an FMU where it was integrated with the CarMaker full vehicle model. The driveline model would need a controller to control the TVDC system in different driving scenarios. This control model was built in Simulink and the model was then simulatable. Simulations were run in CarMaker to validate the model and its behaviour.

3.1 TVDC driveline interface

A crucial step in defining the TVDC driveline model was to decide what interfaces the model should have. Interfaces refer to the input and output signals by means of which information exchange happens between two models. Since the Modelica model of the TVDC system would be imported on Simulink for integrating it with the CarMaker vehicle model, it was obvious that CarMaker would define the interfaces for the TVDC model.

CarMaker has several standard powertrain model interfaces that can be used to import user-defined powertrain models into the CarMaker full vehicle model, often referred to as the CarMaker Virtual Vehicle Environment (VVE). There are a wide variety of standard model structures for conventional ICE, BEV and hybrid powertrains, which can be parameterized by the user. For user-defined powertrain models, the OpenXWD model is of interest. The OpenXWD model can be used to define a powertrain or a driveline. The OpenXWD model gives the user the possibility to decide the distribution of drive torques to the wheels, and the wheel speed calculation is done at the CarMaker end [15]. There are three configurations for the OpenXWD Powertrain model: with engine, with motor and stand-alone. The stand-alone model allows the user to define the powertrain power source. For the current case, since the TVDC driveline is used in a BEV, the OpenXWD configuration 'with motor' was used. The interfaces for the OpenXWD driveline are shown in figure 3.1.

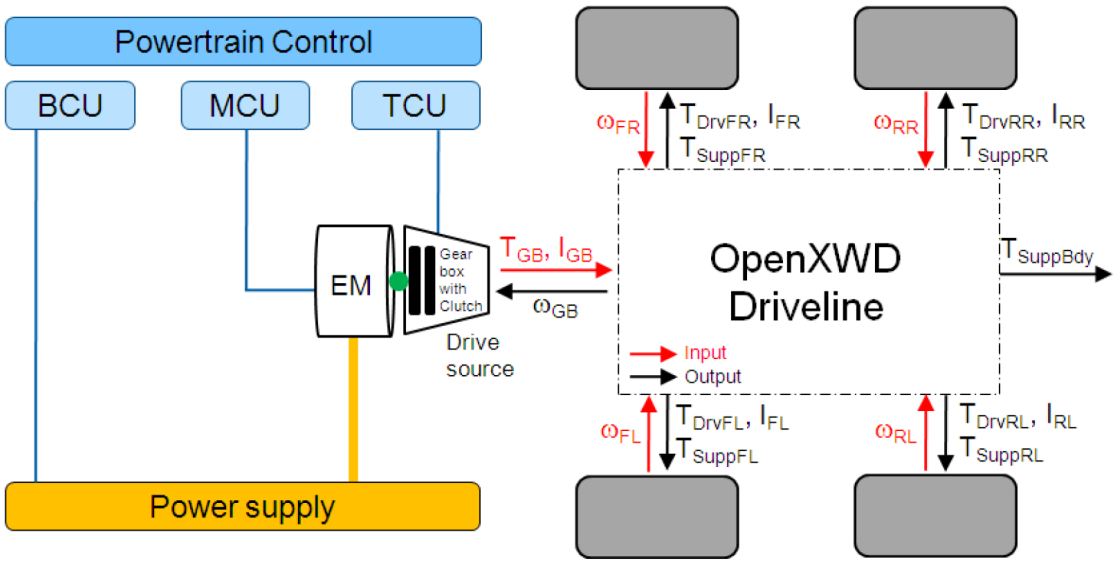

Figure 3.1: Interfaces of CarMaker powertrain model 'OpenXWD Driveline' with electric motor as drive source [15]

The moments of inertia (I_{GB},I_{RL},I_{RR},..) are actually parameters that are to be input in the CarMaker User Interface. The actual implementation of this OpenXWD driveline for the TVDC system is shown in the figure 3.2.

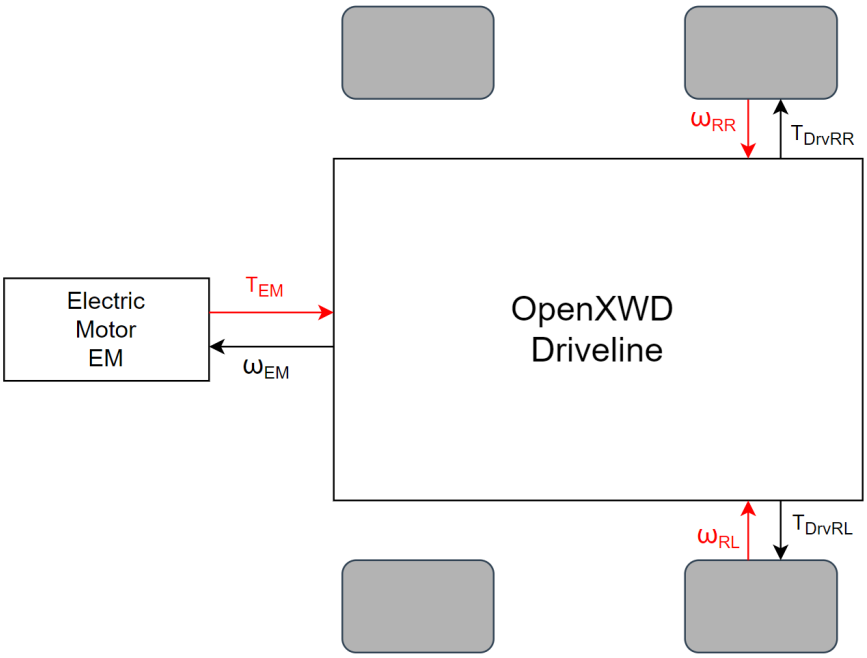

Figure 3.2: OpenXWD driveline as implemented for the TVDC system

The CarMaker VVE with the OpenXWD configuration requires the driveline model to provide the drive torques (T_{Drv}) to the wheels. CarMaker then returns the wheel speeds (ω) to the driveline model. It is used to calculate the motor speed (ω_{EM}) inside the driveline model, which is fed back to the CarMaker model, completing the loop. The motor torque (T_{EM}), which is the input torque to the gearbox in the TVDC driveline model, is given by CarMaker based on driver demand.

The TVDC driveline additionally needed interfaces to control the clutches by providing a torque capacity from a control model. In addition, it was useful to track the state of slip in the clutches to check if the clutches were functioning as expected. The clutches also had a time constant defined for the torque capacity applied, hence the actual clutch torque capacity was also an output that was monitored in the system.

3.2 Modelica library-based model

Modelica consists of predefined standard libraries for various systems that simplify the modelling process and save time. These libraries provide a very convenient way to model a physical system by using pre-defined blocks for components of engineering systems. These blocks can be connected by acausal connectors to build a system. The mechanical components are usually found in the 'Mechanics' library of the Modelica standard libraries. From the TVDC system overview in figure 2.1, the system would need to model the motor shaft, the single-ratio gearbox, the two clutches and the two output shafts as part of the model. The system was designed to have these components - a rotational inertia element representing the motor shaft input to the gearbox, a simple gear system, two clutches and two stiffness elements representing the output shafts driving the wheels. The outline of the TVDC model with elements to be modelled is shown in figure 3.3.

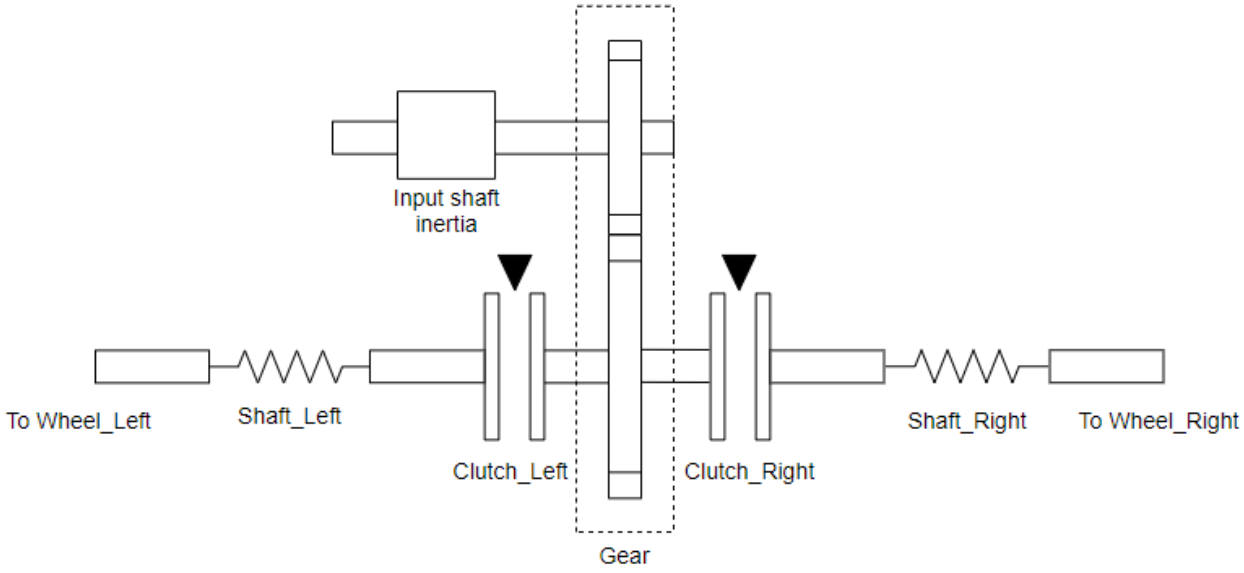

Figure 3.3: TVDC schematic with components to be modelled

Modelica library has all these components in the form of blocks defined in the library path 'Mechanics > Rotational > Components'. Information about these blocks can be found in the Modelica User's guide [16], under Libraries.

3.2.1 Component Description

The components in the Rotational library are described in this section. All these elements have two flanges each, 'flange_a' and 'flange_b', which basically serve as connection points for the acausal connectors. These flanges are the endpoints present in each component and for a rotational flange, torque and rotation angle can be measured at each flange after simulation. Below is a brief description of these components along with the Modelica code for the equations defining them. The torques are denoted by the variable 'tau', the rotation angles by 'phi' and the rotation speeds by 'w'.

Inertia

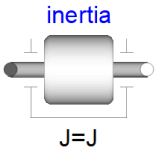

Figure 3.4: Modelica library rotational inertia element

The inertia element signifies the rotational inertia of a body. It contains a parameter field for the moment of inertia 'J' in kgm^2. The flanges can be seen on the inertia element. The grey circular element at the left end of the inertia element represents 'flange_a' and the white circular element represents 'flange_b'. The angle, angular speed and angular acceleration can be initialized for the element. The Modelica code for this element is shown below.

```
\\ Inertia model
    phi = flange_a.phi;
    phi = flange_b.phi;
    w = der(phi);
    J*der(w) = flange_a.tau + flange_b.tau;
```

Ideal Gear

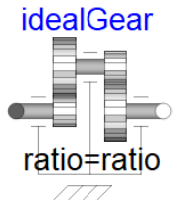

Figure 3.5: Modelica library gear element

As the name signifies, this element is an ideal gear box in which a gear ratio can be specified as a parameter. The Modelica code for this element is shown below.

```
\\ Ideal Gear model
    flange_a.w = ratio*flange_b.w;
    0 = ratio*flange_a.tau + flange_b.tau;
```

Spring

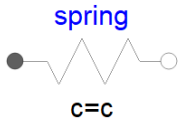

Figure 3.6: Modelica library spring/stiffness element

This element models the stiffness of a shaft, where the spring constant is a parameter in Nm/rad. The Modelica code for this is

```
\\Spring model
    flange_a.tau + flange_b.tau = 0;
    tau = flange_a.tau;
    tau = c*(flange_b.phi - flange_a.phi);
```

Clutch

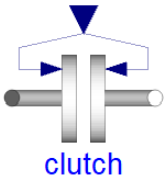

Figure 3.7: Modelica library clutch element

The Modelica clutch element uses a maximum normal force (fn_max) that can be applied to the clutch as a parameter. A geometry constant can be set to convert this into a maximum torque capacity. This geometry constant is usually set to one so that the numerical value of fn_max will represent the torque capacity of the clutch. The normal force to engage the clutch is provided as a normalized force $f\_normalized$ ($0 \leq f\_normalized \leq 1$) of fn_max i.e. the force applied is $f\_n = f\_normalized * fn\_max$. This normalized force is the input to control the torque capacity of the clutch. The clutch model consists of states for forward slip, backward slip, stuck and disengaged condition. There is more information about the modelling of friction and slip in the User's guide in Dymola [16].

3.2.2 Acausal connectors

The library-based modelling in Modelica involves blocks that are connected to each other via connectors. Since it is beneficial to use acausal modelling for plant models in Modelica, these connectors are also acausal in nature. Connectors involve in the exchange of physical quantities between the components, and their acausal nature makes the direction of this exchange irrelevant, i.e. the user just needs to connect the two components and the solver will solve for the required outputs.

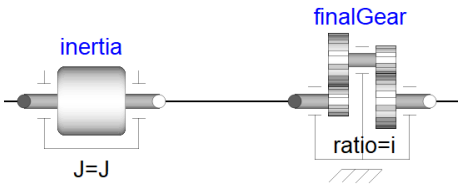

Figure 3.8: Acausal connector between an inertia and gear element built using standard Modelica libraries

Figure 3.8 shows an acausal connector that connects the flanges (ends) of an inertia element and a gear. Systems can be built by using such acausal connectors between library blocks, instead of writing equations and this simplifies the plant modelling process.

Connector variables are of two types- 'flow' variables and 'effort or potential' variables. If a connector variable is not declared as 'flow' variable, it is an 'effort' variable. Effort variables are simply set equal when connected. If a connector variable is declared as 'flow' variable, the sum of them is set equal to zero. These definitions of flow and effort variables can handle the case when more than two connectors are connected to one component block. The equations for the connector are

```
\\Connector equation
    inertia.flange_b.phi = finalGear.flange_a.phi; // effort variable
    inertia.flange_a.tau + finalGear.flange_b.tau = 0; // flow variable
```

3.2.3 Interfaces

Figure 3.9: Modelica library input and output interfaces

Interfaces are used to exchange information between models. The Modelica library consists of real input and output interfaces, found under 'Blocks > Interfaces > RealInput/RealOutput'. These interfaces transmit signals which have no physical units and are just magnitudes, in and out of the models. They differ from physical quantities like torque or speed which have units. The input interface is usually connected to the source of a physical signal and the output interface is usually connected to a sensor. These interfaces can also be used in equation-based models where there are no blocks and they can be assigned to exchange the values of predefined input and output variables.

3.2.4 Sources

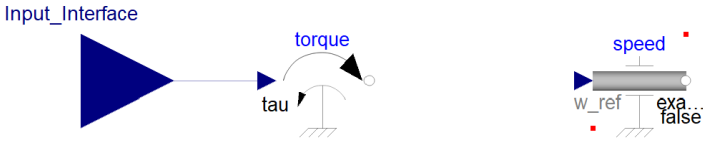

Figure 3.10: Modelica library torque source with input interface and speed source

Sources convert a unitless signal from an interface or any other signal source into a physical quantity like torque or speed that can be applied to the flange of a physical component. This source can be connected to an input interface to get magnitudes of unitless signals as input that serve as magnitudes of the physical quantity. Figure 3.10 shows commonly used sources in the 'Rotational' library, namely torque and speed source.

3.2.5 Sensors

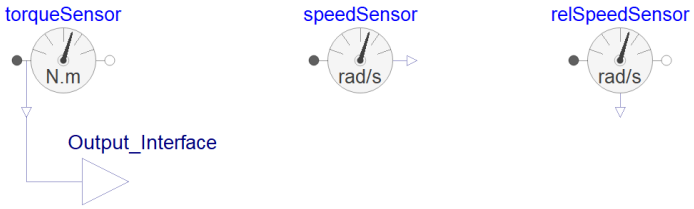

Figure 3.11: Modelica library torque sensor with output interface, speed sensor and relative speed sensor

Sensors are used to measure the values of quantities at a component flange during its simulation and these values can be plotted post simulation. The sensor can be connected to an output interface to convert the physical quantity being measured into a signal that goes out of the model. Figure 3.10 shows commonly used sensors in the 'Rotational' library, namely torque, speed and relative speed sensors. The torque and speed sensors are to be connected between the flanges of two components. The relative speed sensor is used to measure the relative speed across two flanges of a component.

3.2.6 TVDC model using libraries

Using the schematic shown in figure 3.3, the model of the TVDC driveline was built using Modelica libraries as shown in figure 3.12.

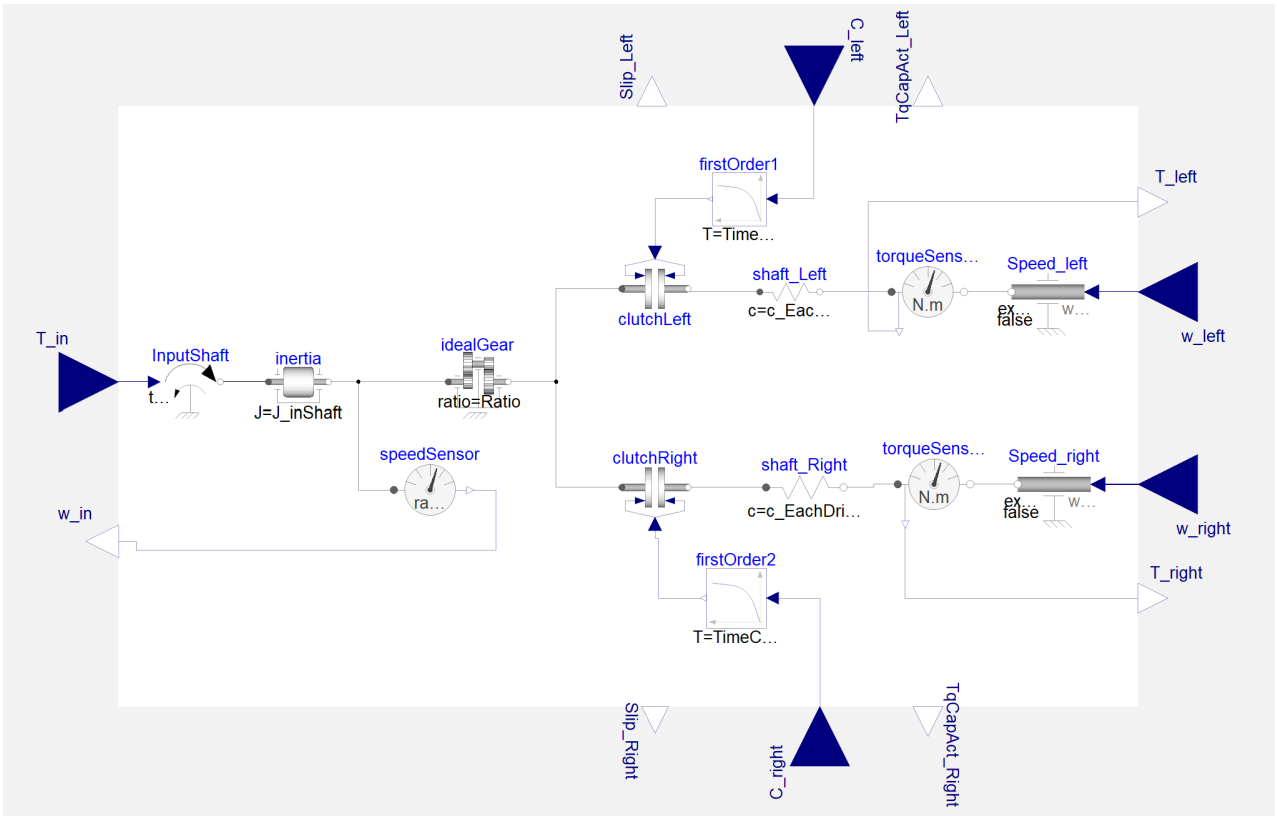

Figure 3.12: TVDC model built using Modelica libraries. Fails to simulate as the solver fails to converge to a solution

The TVDC model consists of an input interface for the torque signal from the motor model in CarMaker which is connected to a torque source 'InputShaft' which converts it into a physical quantity. This is then connected to an inertia element and an ideal gear block. At this point, the torque is split between the two output shafts. Each of these shafts has a clutch and a stiffness element representing the output shaft. Torque sensors on each of the output shafts are used to measure the torque at the output shaft and this is connected to an output interface. A speed source receives the wheel speed signal returned by the CarMaker model through the input interface and converts it into a physical rotational speed for each of the output shafts. The motor speed from the TVDC model is returned to the CarMaker model from an output interface using a speed sensor connected to the inertia element representing the motor shaft. The state of slip in the clutch is an output interface, which takes its value from the slip variable 'mode' from the library clutch block. The clutch blocks are controlled by a normalized force input, through the input interfaces 'C_Left' and 'C_Right' and a first order transfer function block that has a time constant for the clutch control signal. In addition to this, two output interfaces for measuring the actual torque capacity of the clutches are also present in the model.

3.2.7 Model Performance

The model was built according to the planned schematic, in order to have the same components as the schematic. However, the model failed to simulate and the Modelica solver fails to converge to a solution. This is because the library models have some constraints regarding how the component blocks can be connected. For instance, Modelica's in-built examples containing a clutch have inertia elements on either end of the clutch. The clutch slip in Modelica is modelled as discrete state events. This kind of constraint exists in library-based systems containing such discrete state systems and is mentioned in the section on 'Requirements for simulation tools' in the web article [17]. The library-based modelling approach is lacking in the flexibility of modelling since it does not guarantee the simulatability of any component from the rotational library connected to other components with the same flange connectors. Hence this modelling method cannot be used for any predefined component structure, without verifying its simulatability on Modelica. The focus was to build the model as planned in the schematic and when the approach did not work, it was decided to move on to the equation-based modelling approach. At a later stage of the thesis work, it was found that the model becomes simulatable on adding inertias on the output ends of the clutch and a brief account of this finding is given in the following section.

3.2.8 Model Update

The Modelica library model from section 3.2.6 was updated by adding an inertia element on the output shaft side of the clutches before the stiffness element. This made the model simulatable, as inertia blocks on either side is the most suitable surrounding for a clutch, which is highlighted in Modelica's in-built examples for rotational libraries. The updated model is shown in figure 3.13.

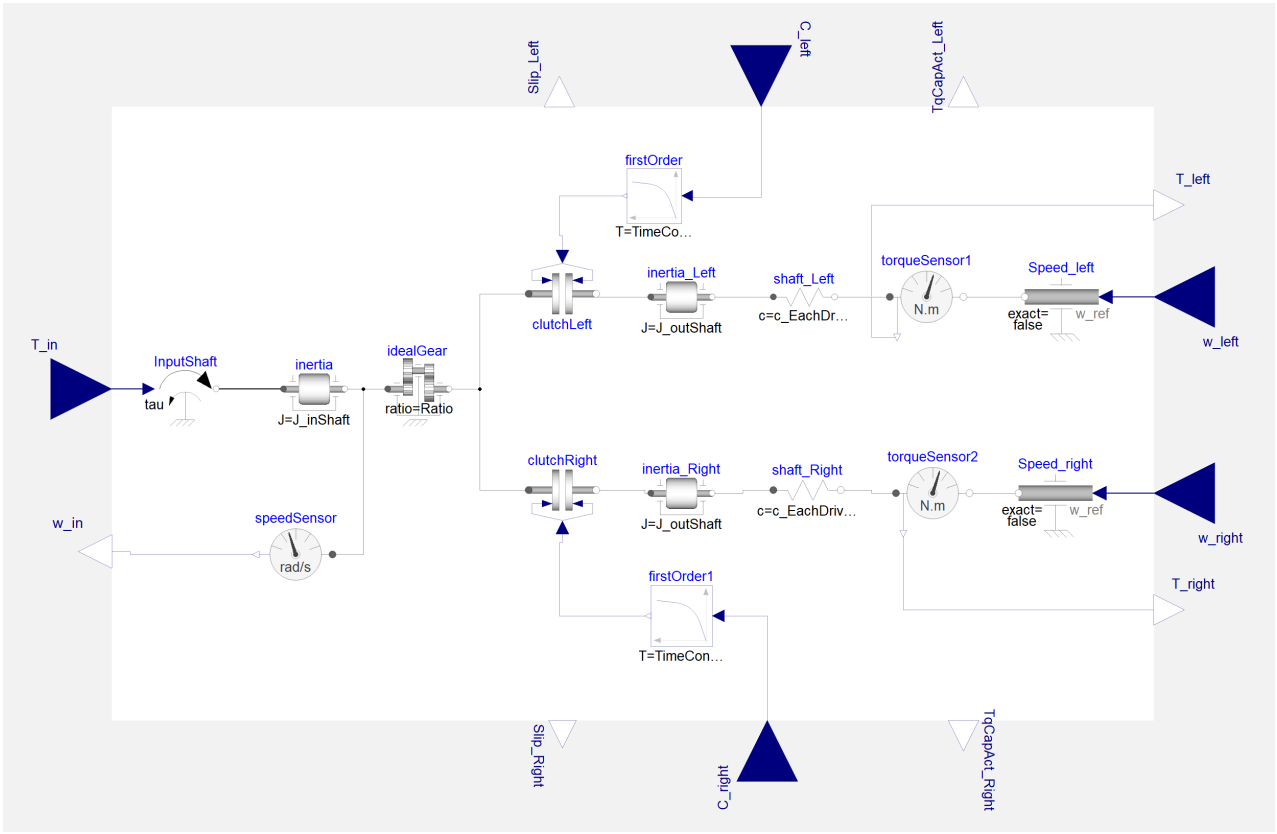

Figure 3.13: TVDC library model updated with inertias. Model is simulatable

This model, when integrated with the CarMaker VVE is simulatable and the preliminary observations of this model's performance will be discussed in the Results section.

3.3 Modelica equation-based model

The next phase was to explore the possibility of building the model using equations, a method also referred to as the 'object-oriented flat model approach' [10]. This method does not have the block/component structure that the library-based approach possesses, but it gives a lot of flexibility in building the system and connecting different components with each other through the equations. For instance, this approach does not need the clutch to be connected to inertia elements on either end to get a simulation running, as in the case of library-based models. The lack of the graphical structure makes this method harder to interpret for someone using the model, but the acausal modelling methodology makes this model reusable, once the user has understood how the model is structured. The user can just interchange the input and output interfaces as desired (while ensuring the set of equations remains consistent with the number of unknowns), without having to rewrite the equations for the new set of unknowns.

The schematic shown in figure 3.3 was the basis for the set of equations used in this approach. Figure 3.14 shows a detailed schematic for the equation-based model. The double-headed arrows represent physical quantities like torques and rotation speeds, and the single-headed arrows represent unitless signals that perform information exchange.

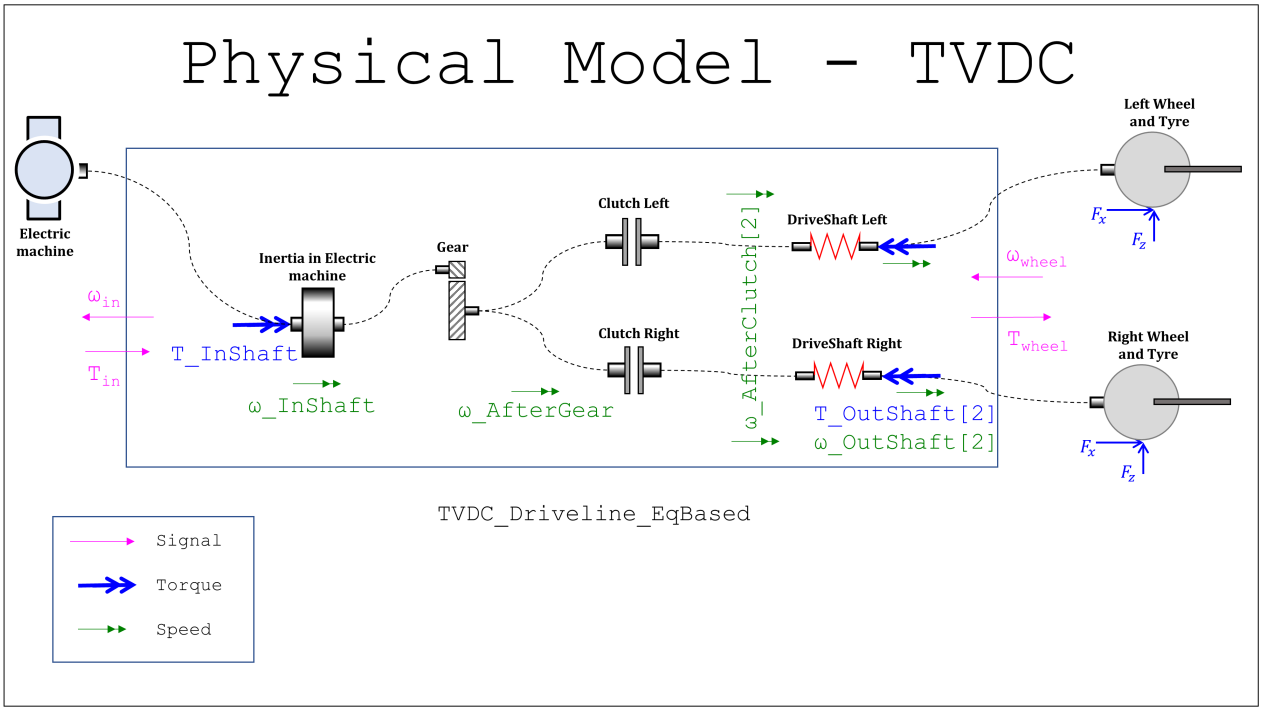

Figure 3.14: TVDC equation based model detailed schematic. Notation from Vehicle Dynamics Compendium [9]

$T\_InShaft$ and $\omega\_Inshaft$ represent the torque and rotational speed at the input shaft from the motor or the inertia element. $\omega\_AfterGear$ represents the rotational speed after the gear. $\omega\_AfterClutch[2]$ is an array of two variables representing the rotational speeds after the two clutches. Similarly, $T\_OutShaft[2]$ and $\omega\_Outshaft[2]$ are the torques and speeds at the driveshaft ends respectively. The model is described by defining the relationship between these torques and speeds across the components, in the form of equations. This set of equations has to be consistent such that the number of variables or unknowns is equal to the number of equations describing them, in order for the model to be simulatable.

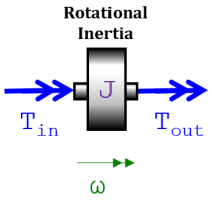

Figure 3.15: Rotational inertia element [9]

A first order relation between the torque and the speed across a rotational inertia element is given by

$$J \cdot \dot{\omega} = T_{in} - T_{out} \tag{3.1}$$

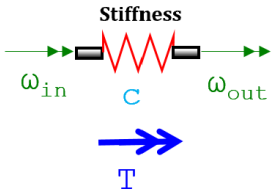

Figure 3.16: Stiffness element [9]

A first order relation between the torque and the speed across a stiffness element is given by

$$\dot{T} = c \cdot (\omega_{in} - \omega_{out}) \tag{3.2}$$

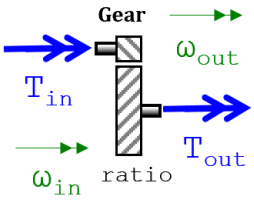

Figure 3.17: Gear element [9]

Torque and speed relations for a gear with ratio is given by

$$\frac{T_{out}}{ratio} = T_{in} \tag{3.3}$$

$$\frac{\omega_{in}}{ratio} = \omega_{out} \tag{3.4}$$

Equations 3.1 to 3.4 were used as a basis to define the relationship between the torques and speeds for all elements except the clutches in the equation-based model. For the inertia and the gear in the model, the equations are enlisted as

```
J_InShaft*der(w_InShaft) = T_InShaft - sum(T_OutShaft)/Ratio;
w_InShaft = w_AfterGear*Ratio;
```

where $sum(T\_OutShaft) = T\_OutShaft[1] + T\_OutShaft[2]$.

At the driveshafts, the relations are

```
der(T_OutShaft) = c_EachDriveShaft*(w_AfterClutch - w_OutShaft);
```

The clutch is modelled with 'discrete dynamics' by defining states for sticking and slipping, according to the modelling technique in the Vehicle Dynamics Compendium [9]. The clutch stick-slip states are defined using the discrete state variable *Slip* = -1, 0 or +1 representing negative slip, stick and positive slip states respectively. The source code listed below contains equations to calculate the torque in the driveshafts based on the clutch state and also contains the state transition conditions for the slip in the clutch. For the sake of simplicity, this code is shown for one clutch and driveshaft only and can be easily extended for two clutches and driveshafts.

```
//Clutch:
      if abs(Slip) == 1 then  // If Slip occurs
        der(T_OutShaft) = sign(Slip)*der(T_ClutchCap);
        //Using T_OutShaft = sign(Slip)*T_ClutchCap;
      else
        w_InShaft/Ratio = w_AfterClutch;
      end if;

      w_Rel = w_InShaft/Ratio - w_AfterClutch;

      when pre(Slip) == 0 and T_OutShaft > T_ClutchCap then
        Slip = +1;
        reinit(T_OutShaft, +T_ClutchCap);
      elsewhen pre(Slip) == 0 and T_OutShaft < -T_ClutchCap then
        Slip = -1;
        reinit(T_OutShaft, -T_ClutchCap);
      elsewhen pre(Slip) == +1 and w_Rel < 0 then
        Slip = 0;
      elsewhen pre(Slip) == -1 and w_Rel > 0 then
        Slip = 0;
      end when;
```

The clutch actuation is done based on the signal *T_ClutchCapReq*, which is a request or control signal that can be sent from the clutch control module. *T_ClutchCap* is the actual torque capacity of the clutch and is the result of a time constant resulting in gradual change between different clutch capacities.

The equation-based model had no issues in simulating and hence it was decided to validate it in CarMaker simulations after integrating it in the VVE.

3.3.1 Exporting model as an FMU

The driveline model with its interfaces was exported in the Functional Mock-up Interface (FMI) 2.0 standard format, as a Functional Mock-up Unit (FMU). The FMI model acts as a black box model with input and output connections (interfaces) and pre-defined parameters that can be set in the FMU block. In order to be able to change any parameter after exporting the model as an FMU, the particular value must be declared using the 'parameter' data-type or qualifier in the Modelica text editor. Parameters can be regarded as 'input data' to a model that are constant with respect to time [2]. This declaration of input data as a parameter can be done for a library-based model also, by specifying the parameter with a variable name in the component blocks and declaring that variable name with the 'parameter' qualifier. Refer to section A.1 in the Appendix for instructions on exporting the Dymola model as an FMU. The FMU was exported in the 'Model Exchange and Co-Simulation' format. The FMU can also be imported on Dymola again, to check for basic functionality.

3.4 Driveline model integration on CarMaker

The TVDC driveline FMU can be integrated with the CarMaker Virtual Vehicle Environment (VVE), using the cm4sl (CarMaker for Simulink) Simulink models or directly using CarMaker's 'FMU Plug-ins', without using Simulink. The FMU Plug-ins require the imported FMU to be connected to the corresponding CM input and output signals. It is also required to adapt the FMU to an appropriate Model Class on CarMaker which defines the system that the FMU represents. This would make the simulations run faster as it runs directly in the CarMaker environment and also would eliminate the need for a Simulink license. Figure 3.18 shows the FMU Plug-ins window that can be accessed from Application > FMU Plug-ins in the CM window.

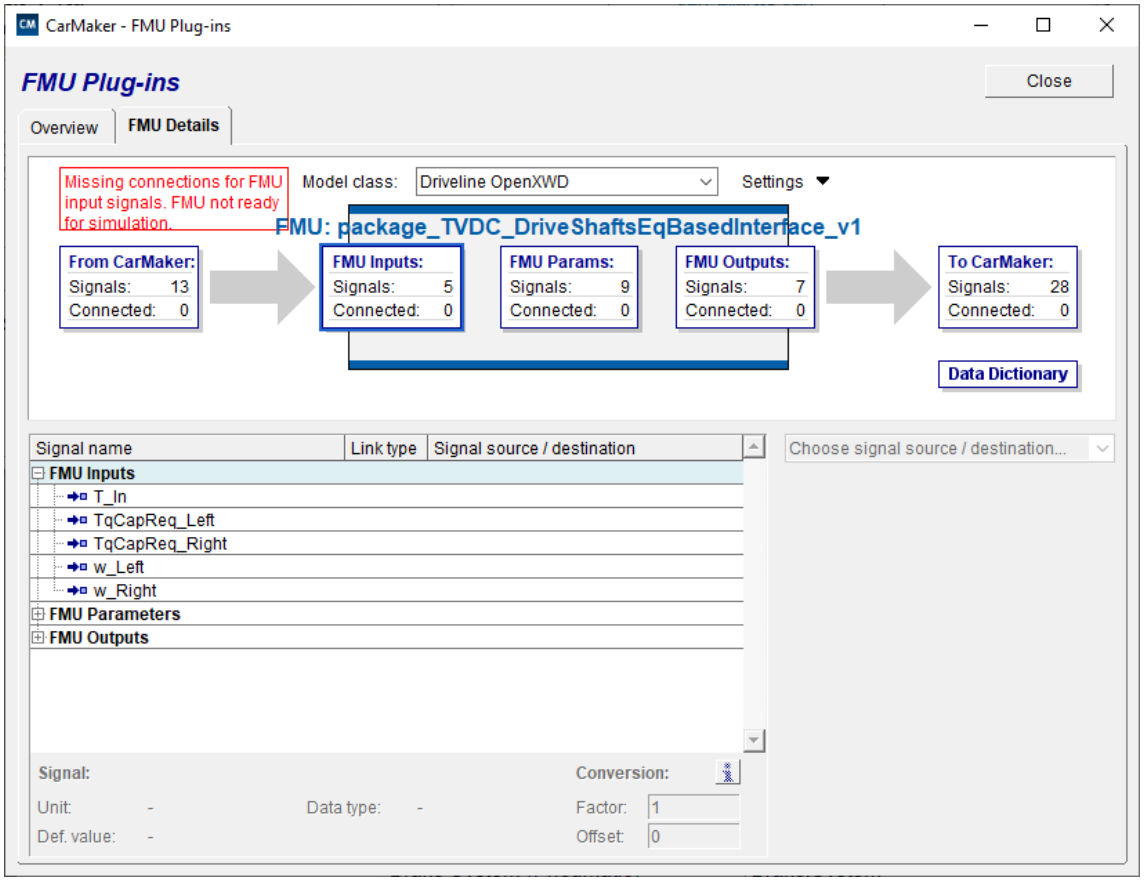

Figure 3.18: FMU Plug-ins on CarMaker

Since the TVDC driveline FMU needed a control module to control the clutches, it was easier to have the control block on Simulink as the clutch control could be tuned in between simulations quite easily, as compared to FMU Plug-ins, where the control model could not be modified and would need to be re-imported. The FMU Plug-ins would also need a control block modelled as an FMU and connected to the clutch actuation signals in the TVDC driveline FMU. This in turn would need a new CarMaker dictionary variable to be defined for the interfacing. For integration of the TVDC driveline FMU with CarMaker, cm4sl was chosen as it was easier to use Simulink for the control block and also the VCC CarMaker model already used cm4sl for the powertrain model.

The driveline model was integrated with the CarMaker Virtual Vehicle Environment (VVE) using the cm4sl (CarMaker for Simulink) Simulink models found in the CarMaker project folder. Refer to section A.2 in the Appendix for instructions on importing the FMU in Simulink. Since the CarMaker Programmer's guide [18] mentions using FMU for Co-Simulation, while importing the TVDC driveline FMU, the 'Co-simulation' option was chosen for the FMU block on Simulink. The TVDC driveline FMU on Simulink is shown in figure 3.19. The FMU had the same interfaces as specified in the equation-based Modelica model on Dymola.

Figure 3.19: TVDC driveline Functional Mock-up Unit (FMU) imported in Simulink

On double-clicking the block, the preset parameters for the model can be seen and these values can be changed if required, for varying the characteristics of the model.

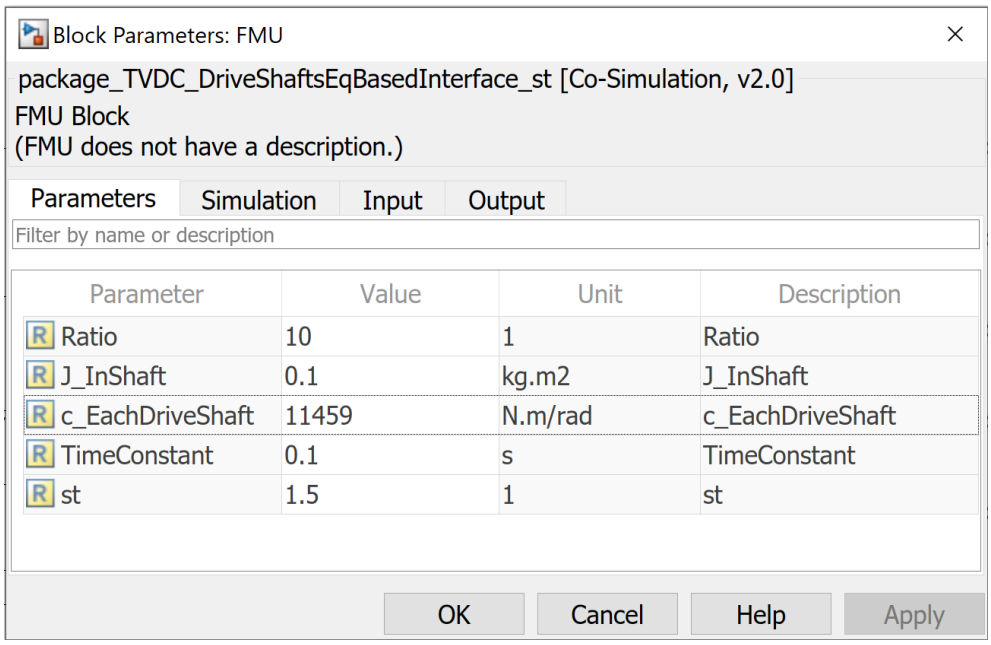

Figure 3.20: TVDC driveline FMU parameters dialog box

3.4.1 Integration on CM 'generic' model

Since the TVDC driveline model was built according to the CM OpenXWD interfaces, it was integrated first on the CM 'generic' Simulink model, found in the 'src_cm4sl' folder in every CM project folder. This is the most basic model of CarMaker for Simulink, and the user can simply insert the OpenXWD driveline model in this model. The CarMaker for Simulink interface is shown in figure 3.21. The 'Open CarMaker GUI' block opens the CarMaker environment, such that the Simulink signals modified in the 'generic' model will overwrite the CM default model signals.

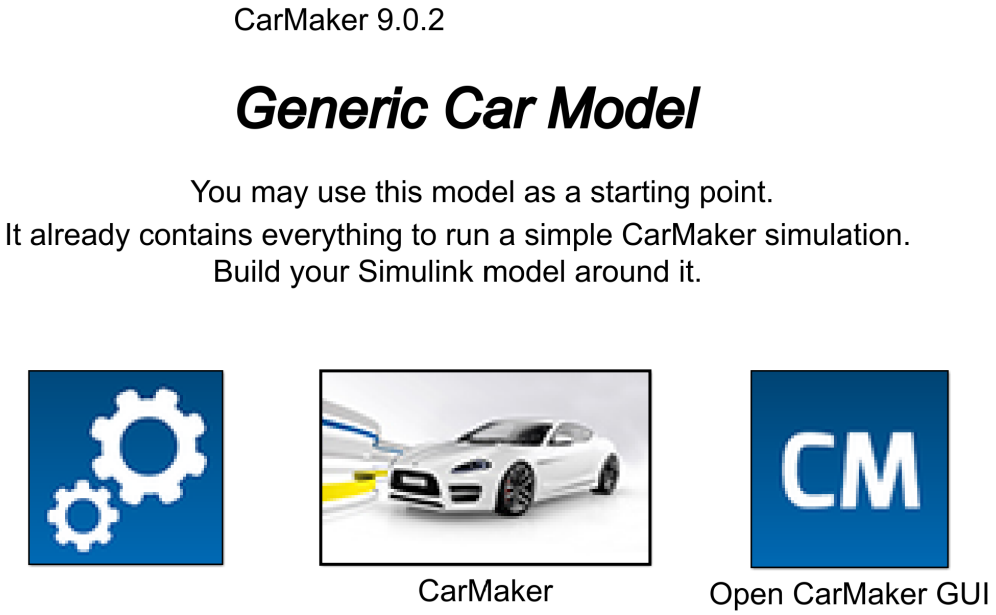

Figure 3.21: cm4sl generic model UI

Figure 3.22 shows the schematic for the integration of the TVDC model into the CarMaker generic model.

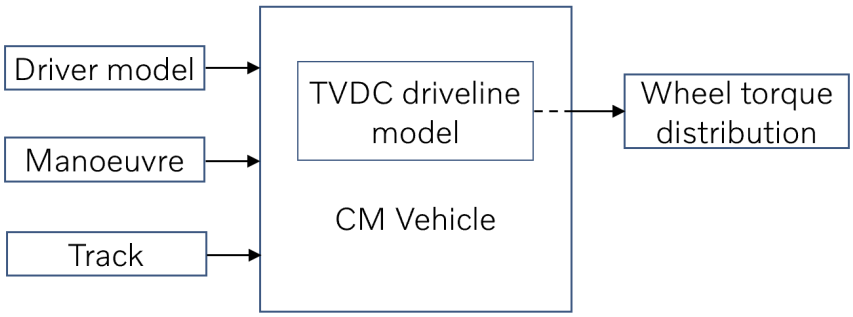

Figure 3.22: Schematic for integration on CM Generic model

The TVDC FMU was imported in the 'CarMaker > IPG Vehicle > Powertrain' path on Simulink. The FMU was imported in this level and was connected to the CM input signals to the OpenXWD model, also described as the CarMaker User Accessible Quantities (UAQs), by reading these signals using 'Read CM Dict' blocks (shown in orange in figure 3.23). These blocks were used to input the CM signals to the FMU input interfaces. Similarly, 'Write CM Dict' blocks (also in orange) were connected to the output interfaces of the TVDC FMU to write the values of variables returned by the TVDC driveline FMU to the CM signals or User Accessible

Quantities. On integration, CarMaker runs its full vehicle model with the signals from the 'Write CM Dict' blocks on Simulink as the signals driving the CM vehicle model. This generic CarMaker model was used only to check the 'robustness' of the TVDC driveline model. The main intention was to ensure that the model runs in different driving scenarios, without giving any errors, using a basic level of control. This could be referred to as a 'qualitative' study of the model performance.

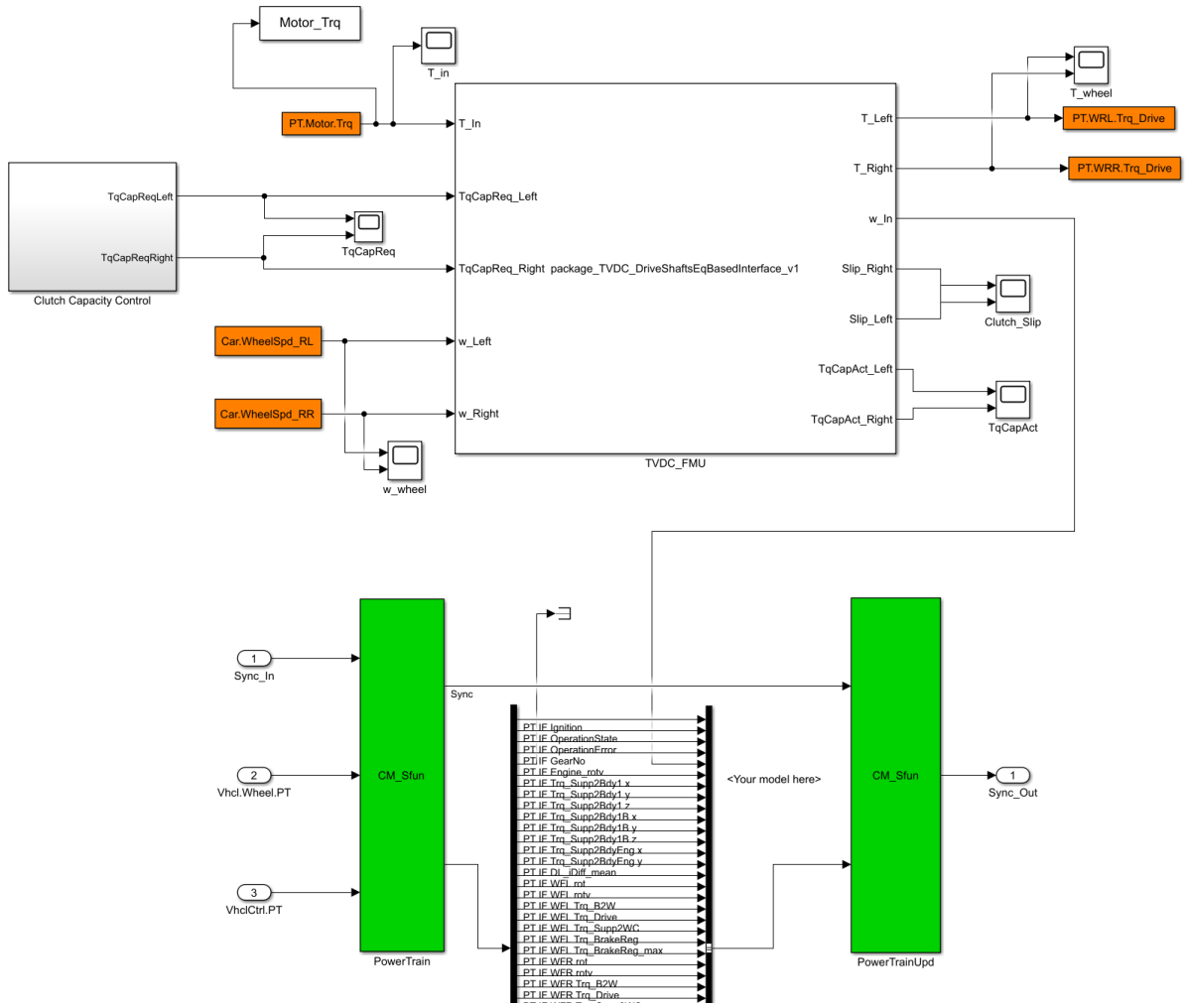

Figure 3.23: TVDC driveline FMU integrated with the generic (default) CarMaker VVE

3.4.2 Integration on VCC CarMaker model

Volvo Cars Corporation also have custom models for their vehicles on CarMaker. The Electric Driveline team at Volvo Cars uses the CarMaker for Simulink model with their own powertrain models imported from various other modelling tools, including models of different vehicle subsystems and their control. The TVDC FMU was integrated with the RWD and AWD Simulink models used at VCC. The driveline model used by VCC for its BEVs was replaced by the TVDC driveline model in the CarMaker VVE, and the model uses VCC's electric motor model. Figure 3.24 shows the schematic for the integration of the TVDC model into the VCC CarMaker model.

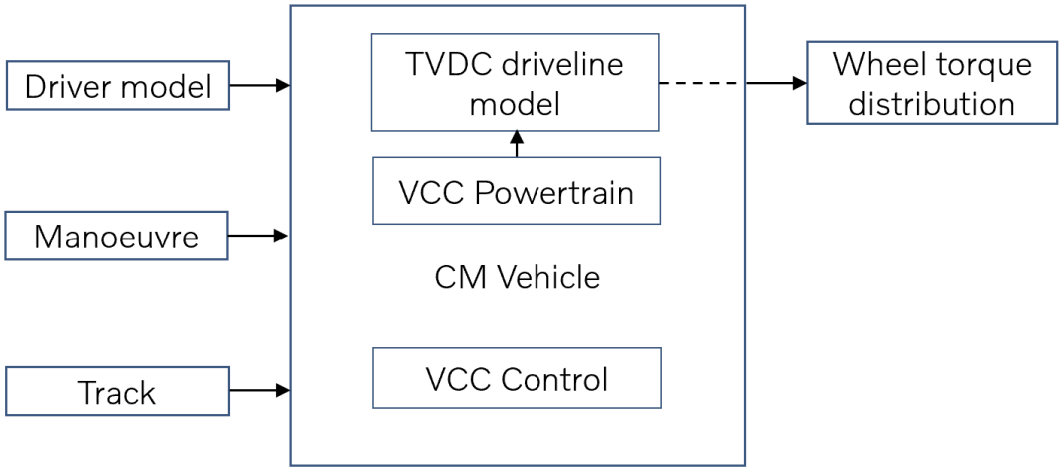

Figure 3.24: Schematic for integration on VCC CM model

Figure 3.25 shows the TVDC FMU in the VCC CarMaker for Simulink VVE with the interfaces of the FMU connected to the corresponding CM signals. The VCC CarMaker models were used to generate the comparisons shown in the 'Results' section of the report and this was a quantitative measure of the performance of the model.

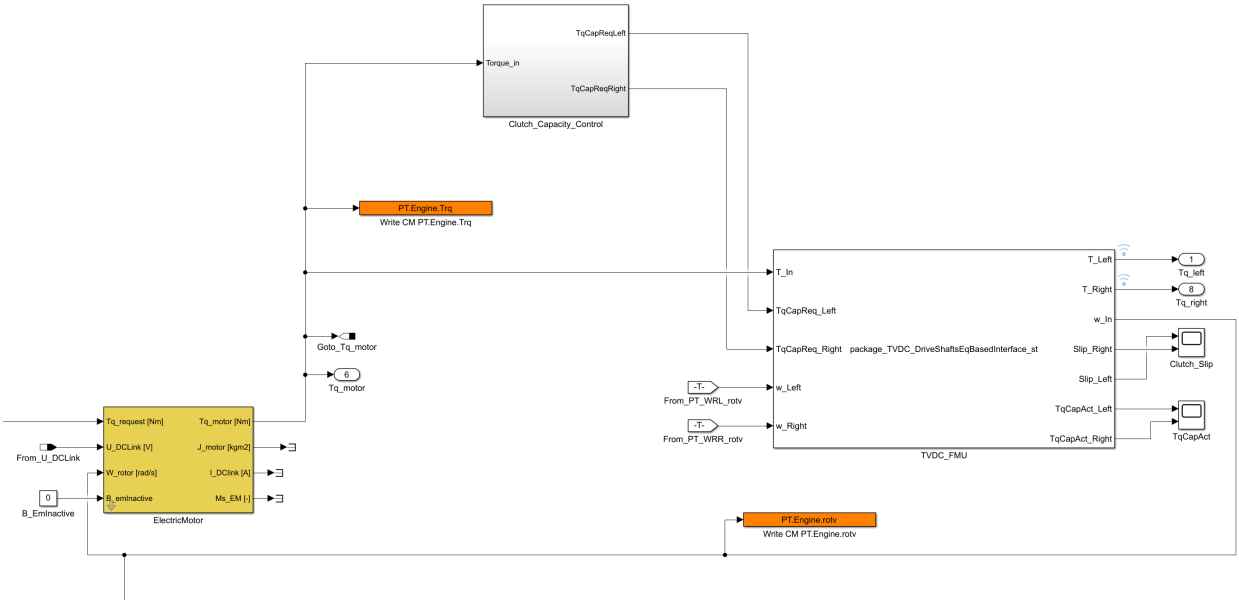

Figure 3.25: TVDC driveline FMU integrated with VCC CarMaker VVE, with the VCC electric machine model

3.5 TVDC controller

In order to obtain the desired performance from the TVDC driveline model, it had to be controlled. The control model would help in achieving the desired torque vectoring function from the TVDC driveline by controlling the clutches using a torque capacity signal calculated using the control algorithm. Both the CarMaker generic model and the VCC model have the control model block 'Clutch_Capacity_Control' controlling the clutch torque capacities, as shown in figures 3.23 and 3.25. The control logic initially developed was updated to have valid comparisons with different scenarios like measurement data and virtual environment simulations. However, this updated model was still not suitable for the data comparisons carried out, as it did not result in similar behaviour from the TVDC system compared to the measurement data. Three different controllers were developed in the process and used in different simulation scenarios.

3.5.1 Controller A

Controller A was a control model developed to have a basic level of control on the clutch torque capacities so that the CarMaker generic model could be tested for its 'robustness', as mentioned in section 3.4.1.

This control algorithm uses basic principles of torque vectoring and vehicle propulsion to induce the torque vectoring function in the TVDC driveline. In a cornering situation, the outer wheel in an axle has a higher normal load than the inner wheel due to the lateral load transfer that occurs as a result of the centrifugal force and the lateral acceleration experienced by the vehicle in a corner. This causes the inner wheel to be unloaded. The principle used for the control logic is that the wheel with a higher normal load (F_z), has greater traction or 'grip' and hence it can be supplied with a higher propulsive force compared to a wheel with a lower normal load. In the case of a vehicle cornering, the outer wheel has a higher normal load than the inner wheel, and hence it can be supplied with a higher torque. The torque distribution, in this case, was based on the normal load on each wheel in the rear axle, for which the TVDC driveline was used. The torque delivered to each wheel was directly proportional to the normal load on that wheel. A pseudocode outlining the control strategy is shown below.

Algorithm 1 Controller A

```
if |delta| < steering_limit then
    Torque_capacity_inner = Clutch_capacity_max
    Torque_capacity_outer = Clutch_capacity_max
else
    Torque_capacity_inner = constant*(Fz_inner/Fz_axle)*T_axle
    Torque_capacity_outer = Clutch_capacity_max
end if
```

A steering angle limit 'steering_limit' was set for the steering wheel angle (delta) so that the control condition does not activate for small inputs at the steering wheel. For steering wheel angles within this steering limit, the two rear wheel clutches were supplied with the same torque capacity, i.e. the maximum torque capacity for the clutches 'Clutch_capacity_max'. For steering wheel angles greater than the steering angle limit, the inner wheel clutch is supplied with a torque proportional to the ratio of the normal load on the inner wheel (Fz_inner) to the normal load on the rear axle (Fz_axle). The inner wheel clutch is supplied with this ratio multiplied by the torque delivered to the axle (T_axle) and a constant ($0<\text{constant}\leq 1$) used to limit the torque further. The outer clutch was provided with maximum torque capacity.

3.5.2 Controller B

The clutch control signals from the initial track measurement data that was obtained were plotted as shown in figure 3.26.

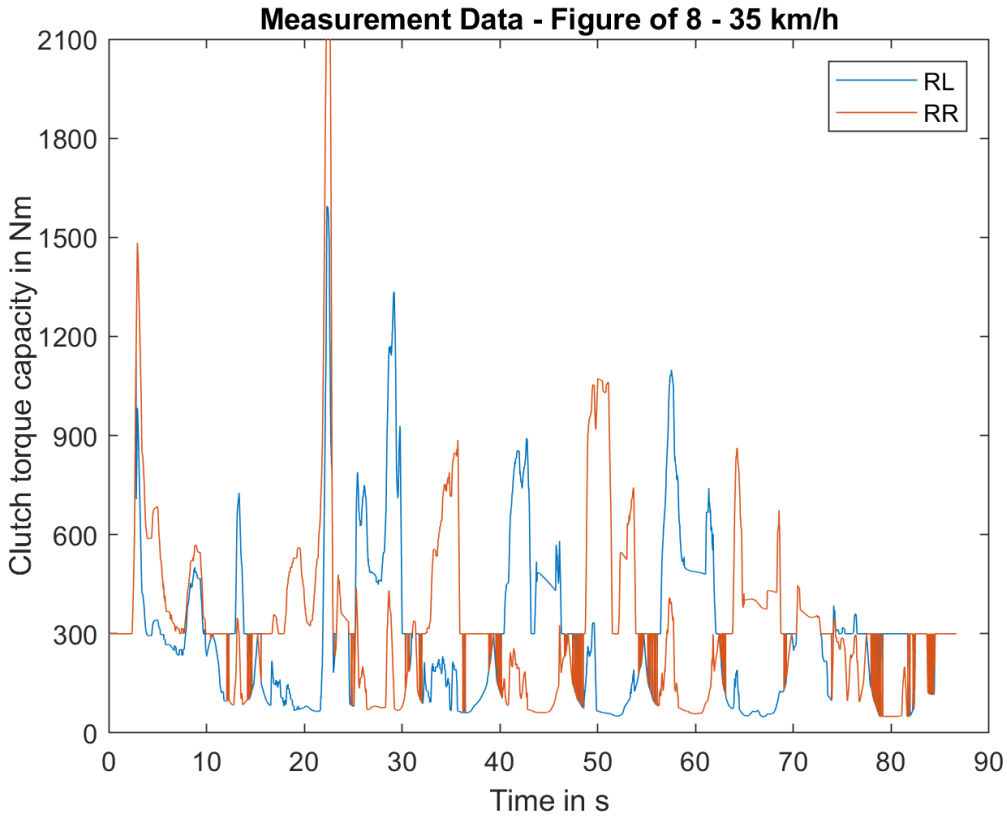

Figure 3.26: Clutch capacity signals from a measurement made on the Figure of eight track, vehicle driven at 35 km/h

This showed that the clutch control strategy was slightly different from what was devised in Controller A. The control strategy from the measurement data hinted at varying the clutch torque capacity around a 'baseline' capacity, in this case, about 300 Nm. Hence the control strategy from controller A was modified to have this 'baseline' capacity and was called controller B. Controller B used the same principle as Controller A for its control, but with a slight modification.

Algorithm 2 Controller B

```
if |delta| < steering_limit then
    Torque_capacity_inner = Clutch_capacity_max
    Torque_capacity_outer = Clutch_capacity_max
else
    Torque_capacity_inner = constant*(Fz_inner/Fz_axle)*T_axle
    Torque_capacity_outer = T_baseline + (Fz_outer/Fz_axle)*T_axle
end if
```

With this control, the simulation results on CarMaker yielded results similar to the measurement data as seen in figure 3.27. This will be discussed again in the results section.

Figure of 8 - 35 km/h

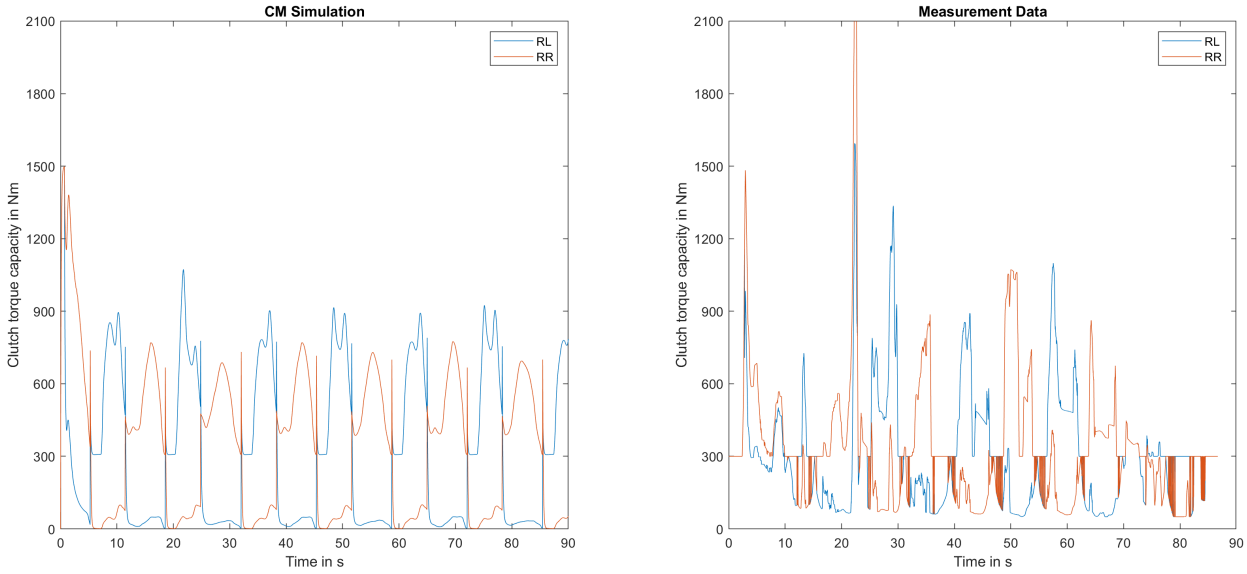

Figure 3.27: Comparison of clutch capacity signals from CM simulation and measurement data on the Figure of eight track, vehicle driven at 35 km/h

3.5.3 Controller C

Measurement data from another set of track tests were obtained and Controller B was not having the same kind of clutch control, as seen by comparison with the clutch capacity signals from the measurement data, for similar manoeuvres on the CM simulation and the test track. It was noticed that the same 'baseline' clutch capacity was present in the test vehicle's clutch control, but the inner wheel clutch was released in a manner that was proportional to the steering wheel angle magnitude, instead of the normal load on the inner wheel. This was incorporated into the control strategy and the simplified pseudocode is shown below.

Algorithm 3 Controller C

```
if a_x < acceleration_limit then
  if |delta| < steering_limit then
    Torque_capacity_inner = max(T_baseline, T_axle/2)
    Torque_capacity_outer = max(T_baseline, T_axle/2)
  else
    Torque_capacity_inner = T_baseline − factor ∗ |delta| ∗ T_axle
    Torque_capacity_outer = max(T_baseline, T_axle − Torque_capacity_inner)
  end if
else
  Torque_capacity_inner = Clutch_capacity_max
  Torque_capacity_outer = Clutch_capacity_max
end if
```

Based on the observations made from the measurement data, it was seen that the inner wheel clutch was released in a corner proportional to the steering wheel angle (delta), and 'factor' is a constant that can be set while tuning the system to get the right amount of clutch capacity release. It was ensured to have adequate torque capacity in the clutches in situations involving high longitudinal acceleration (a_x) and hence an acceleration limit was set for this condition.

3.6 Simulation

The simulation was set up on IPG CarMaker using a driver model and a test track with a manoeuvre defined for the test track for testing the TVDC driveline model integrated into the CarMaker full vehicle model. Most of the test runs had standard test conditions defined for the driver model and in cases where the driver model was not defined, it was usually a steady-state driving simulation where constant vehicle speed was involved and the driver model and manoeuvre was defined as explained in section A.3 of the Appendix. The CM Road was defined using GPS data of the track in '.kml' format and is explained in section A.4 of the Appendix.

3.6.1 TVDC Driveline test simulation

The initial simulation setup was to check the TVDC driveline for its robustness by ensuring that it would run without giving errors in dynamic driving cycles and also in long simulations, at acceptable simulation speeds. The control implementation was also checked for these simulations. The CM 'generic' Simulink model shown in figure 3.21 was used for this test on a circular track of radius 42m for checking the wheel torque distribution based on the control strategy and the Göteborg City Cycle (GCC) which is a long track with some realistic city-style driving scenarios. Controller A was used for these simulations.

3.6.2 TVDC model validation by comparison with vehicle measurement data

The behaviour of the TVDC model was validated by comparing its performance with measurement data obtained from track testing on a vehicle that was fitted with the TVDC driveline. The test vehicle was a Volvo XC90 AWD vehicle with an early version of the TVDC software from the supplier. The controller for the TVDC model in CarMaker was designed by observing the clutch capacity signals in the measurement data since there was no knowledge of the control strategy used in the test vehicle TVDC system, i.e. it was a black box model. Controller B and later Controller C were used in the simulation model of the TVDC system integrated with the VCC CarMaker Simulink model in the AWD configuration. The nature of the rear axle drive torques for the same kind of clutch capacity signals was compared for the TVDC model with the measurement data on tracks like the Figure of eight, Handling Track 1 and Handling Track 2 models from the Volvo Cars Proving Grounds at Hällered.

3.6.3 Comparison of TVDC and open differential performance in simulation environment

The TVDC driveline system was compared with a conventional open differential system which has an equal torque distribution to the rear wheels for energy consumption and steering effort. This was an attempt at understanding any benefits of the TVDC system over the open differential system. This comparison was expected to show how the TVDC model could be used for comparisons in the simulation environment. The VCC CarMaker Simulink model along with Controller A was used for this comparison. The energy consumption was first checked for a straight line driving scenario in order to ensure that the transmission losses were modelled right in the two models, which should give the same energy consumption for both models. Then a comparison was done on the Figure of eight track and Handling Track 1 which are standard testing tracks at the Volvo Cars Proving Grounds at Hällered. The CM Road models for both these tracks were available and simulations were run for the open differential and the TVDC driveline models. Both the open differential and the TVDC model were run on the VCC CarMaker vehicle model with an RWD configuration, and Controller A was used for the TVDC model.

4 Results

Section 4.1 outlines results from the initial TVDC driveline test simulations. The CarMaker full vehicle model (or VVE) with the TVDC FMU integrated was used to run simulations in CarMaker in order to validate the model. Section 4.2 serves as a form of model validation by comparing the results obtained from the CarMaker simulation with the real vehicle measurement data, thus evaluating the model performance. Additionally, a comparison of performance is carried out in a virtual or simulation environment between the TVDC driveline and the open differential driveline, with respect to energy consumption and steering effort in section 4.3. Lastly, the performance of the updated library model from section 3.2.8 is compared against the equation-based model in section 4.4.

Convention and terminology: It must be noted that in a cornering manoeuvre, a positive steering wheel angle represents a left turn and a negative angle represents a right turn. For a left turn, the inner wheel refers to the left wheel and the outer wheel refers to the right wheel. The same can be extended to a right turn as well. This terminology and convention will be used commonly while explaining the result plots. Wheel torque also refers to the drive torque at the wheels delivered by the TVDC driveline. RL refers to the Rear Left wheel/clutch and RR refers to the Rear Right wheel/clutch.

4.1 Results from the TVDC Driveline test simulation

The TVDC driveline with Controller A was simulated on a circular track of radius 42 m as shown in figure 4.1.

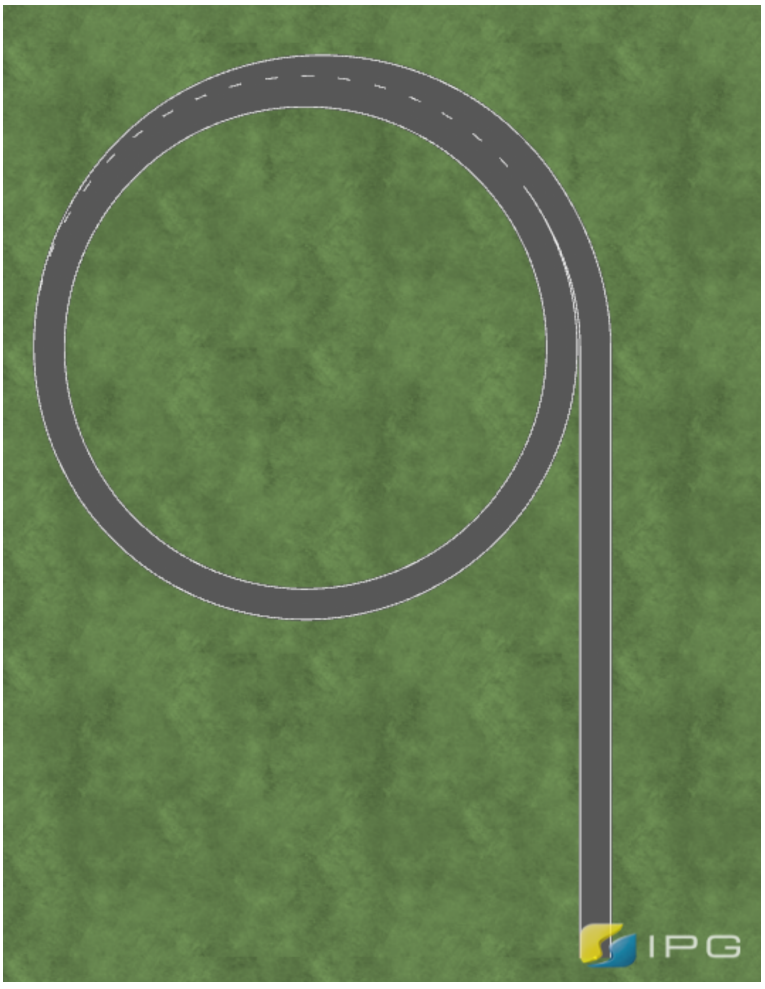

Figure 4.1: 42 m radius circular track

The simulation of the test vehicle on track is shown in figure 4.2.

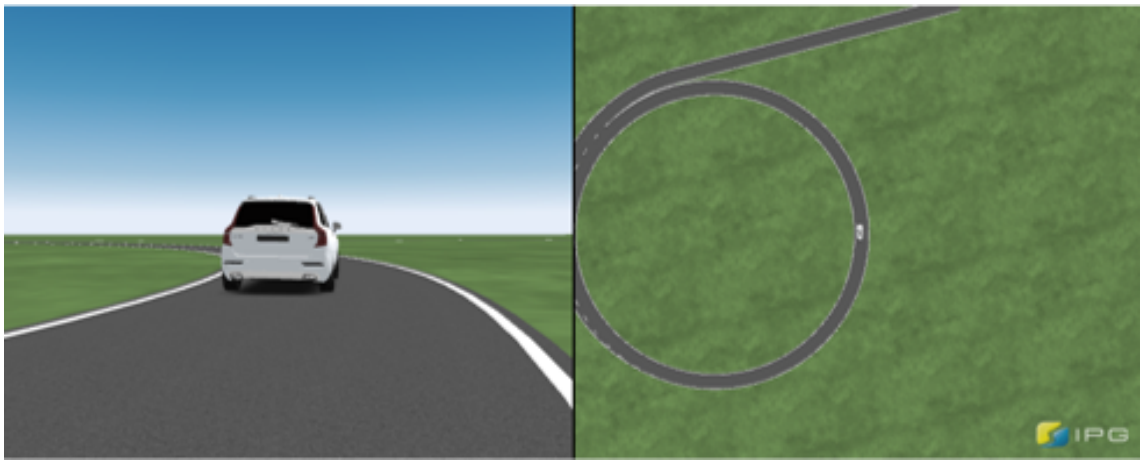

Figure 4.2: CM simulation on 42 m radius circular track

The wheel torques from the simulation are shown in figure 4.3. It is seen that for a positive steering wheel angle (left turn) the outer wheel (right) torque is higher than the inner wheel torque which is as desired from the control strategy in Controller A.

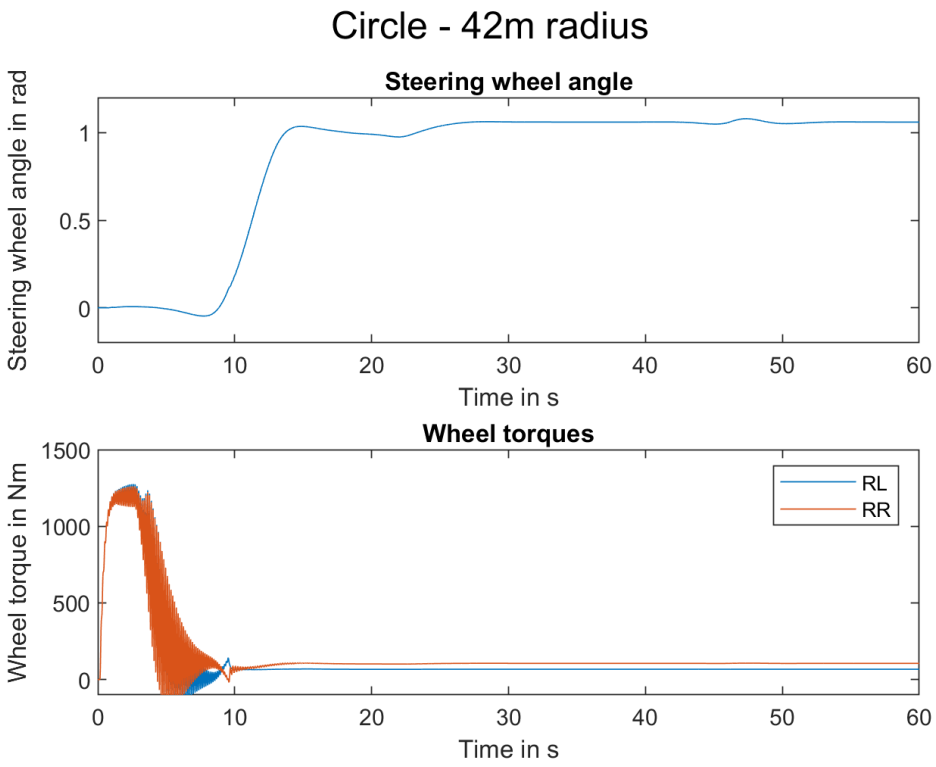

Figure 4.3: Steering wheel angle and wheel torques from the simulation on the 42 m radius circular track using TVDC driveline

The TVDC driveline model with the CarMaker 'generic' vehicle model was used to run a simulation on the Göteborg City Cycle and the model ran for the complete cycle without any errors, thus proving its robustness.

4.2 TVDC model validation by comparison with vehicle measurement data

The CarMaker simulation data was compared with the measurement data from the test vehicle on track and the results are presented below.

4.2.1 Figure of eight track

The figure of eight track is a track at the Hällered Proving Ground of Volvo Cars. On this track, the cornering manoeuvre is carried out symmetrically and hence the TVDC model can be tested and the results of the simulations are comprehended easily. This is a preliminary test to understand whether the intended wheel torque behaviour is obtained according to the control applied to the clutches in the system. The track, along with its dimensions is shown in figure 4.4.

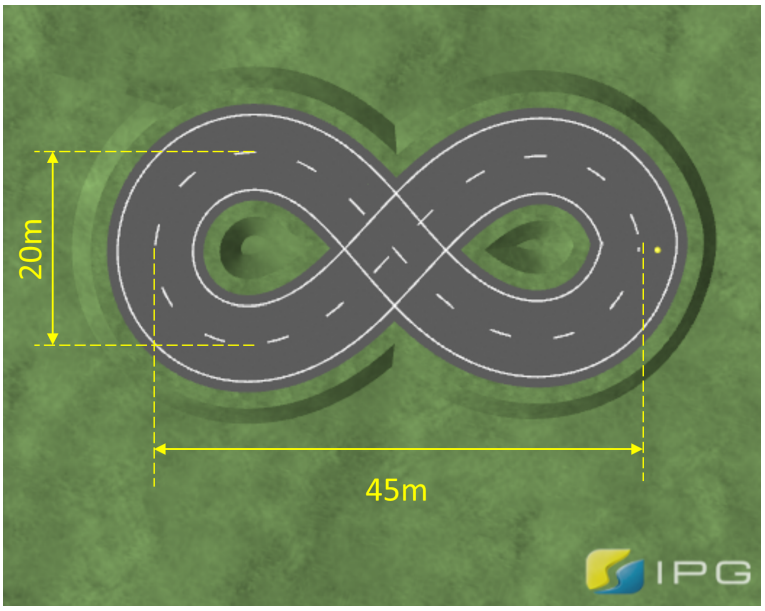

Figure 4.4: Figure of eight track with dimensions

4.2.1.1 35 km/h manoeuvre - Controller B

The first measurement track data that was received, was data measured for a constant speed manoeuvre at 35 km/h on the figure of eight track. Controller B was implemented for the control of the clutches for this test in the CarMaker simulation. A plot of the speed profiles of the CarMaker (CM) simulation and measurement data from the track is shown in figure 4.5.

Note: Comparing the two speed profile plots, it can be observed that though they are in the same range of speeds, they are not exactly the same and the CM speed profile is quite idealistic, which cannot be realized in a real driving scenario, which can also involve inaccuracies in the speed estimation. Hence it is meaningful to only have a qualitative comparison between the two scenarios, where only the nature of the wheel torques is compared, and the magnitudes are not. This kind of analysis will be carried out throughout this section, where only the nature of the torques is compared.

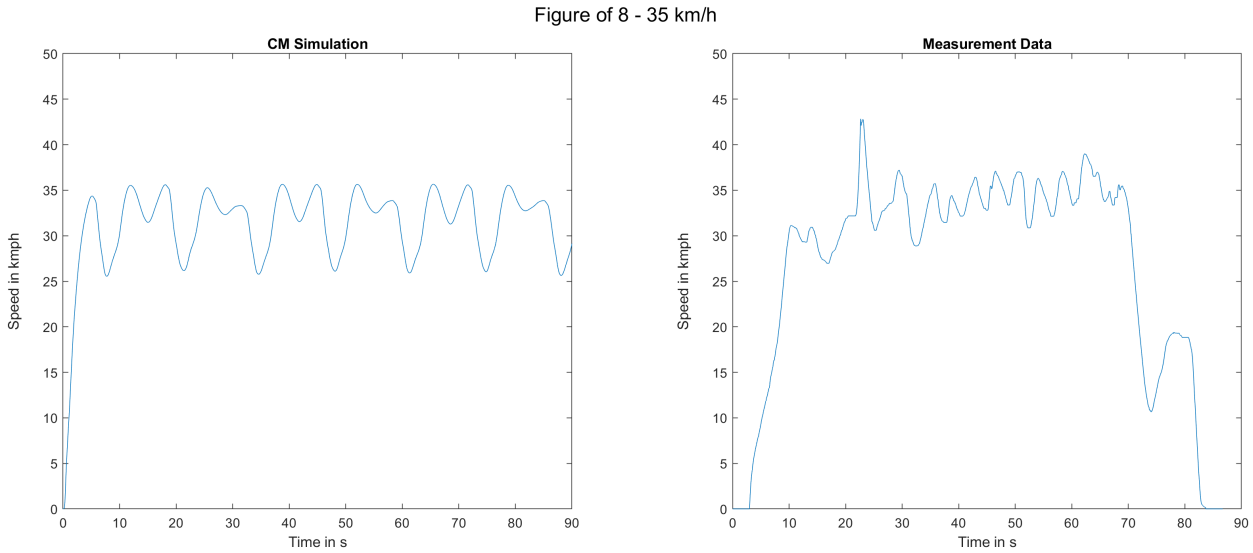

Figure 4.5: Speed profiles from CM simulation using Controller B and measurement data for the 35 km/h test carried out on the figure of eight track

The steering wheel angles, wheel torques and clutch capacities are compared for CM simulation v/s measurement data in the plot shown in figure 4.6.

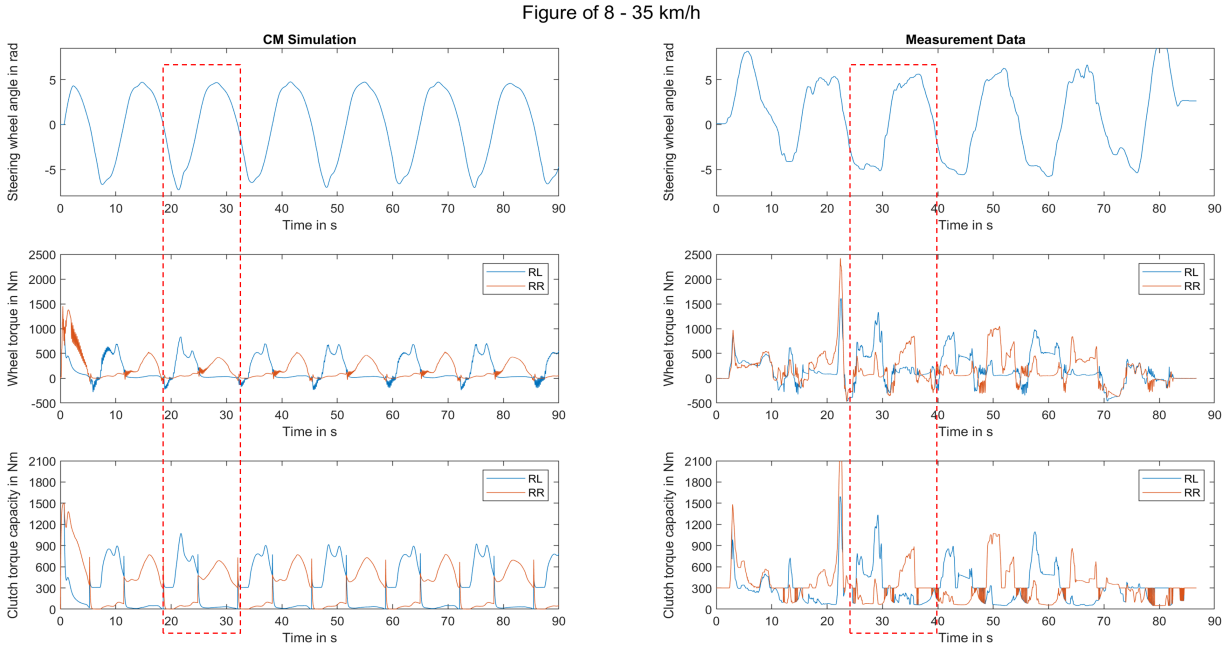

Figure 4.6: Steering wheel angles, wheel torques and clutch capacities from CM simulation using Controller B and measurement data for the 35 km/h test carried out on the figure of eight track. The areas of interest for the comparison are enclosed in red dashed-line boxes

For the sake of comparison, the data enclosed by the red-dashed line boxes are the areas of interest. These regions have similar steering wheel angles and speed profiles and hence provide meaningful comparison. In the CM simulation plots, when time $\approx$ 20s, the steering wheel angle is negative, implying the steering wheel is

turned to the right. It can be observed that the inner wheel clutch i.e. right clutch is released and the outer wheel clutch has a higher torque capacity compared to the inner wheel clutch, which is the expected behaviour based on the control logic for the clutches. Subsequently, it is seen that the outer wheel (left wheel) has a greater torque than the inner wheel (right wheel) which is the required nature of torque distribution for a right turn. On comparing with the measurement data plots, similar behaviour is seen at time $\approx$ 27s, where the steering wheel angle is similar to the CM simulation angle, i.e. it is a right turn, and the outer wheel torque and the corresponding outer wheel clutch capacity are higher when compared to that of the inner wheel. Similarly, on comparing the data at time $\approx$ 30s in the CM simulation plot and at time $\approx$ 37s in the measurement data plots, it is seen that the steering wheel angles are positive, resulting in a higher clutch capacity on the right wheel clutch and a higher wheel torque on the outer (right) wheel. Thus it is seen that the CM data and the measurement data for the figure of eight track at 35 km/h are of the same nature, for similar driving conditions.

4.2.1.2 25 km/h manoeuvre - Controller B

For the next set of measurement track data that was received, data was measured for a constant speed manoeuvre at 25 km/h on the figure of eight track. Controller B was implemented for the control of the clutches for this test in the CarMaker simulation. A plot of the speed profiles of the CarMaker (CM) simulation and measurement data from the track is shown in figure 4.7.

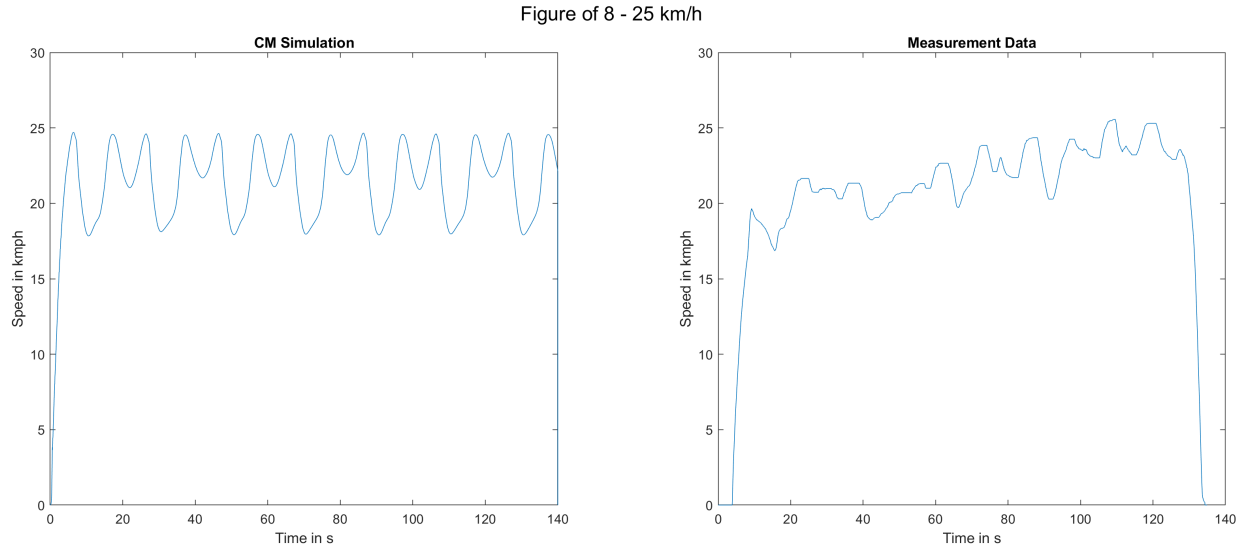

Figure 4.7: Speed profiles from CM simulation using Controller B and measurement data for the 25 km/h test carried out on the figure of eight track

The steering wheel angles, wheel torques and clutch capacities are compared for CM simulation v/s measurement data in the plot shown in figure 4.8.

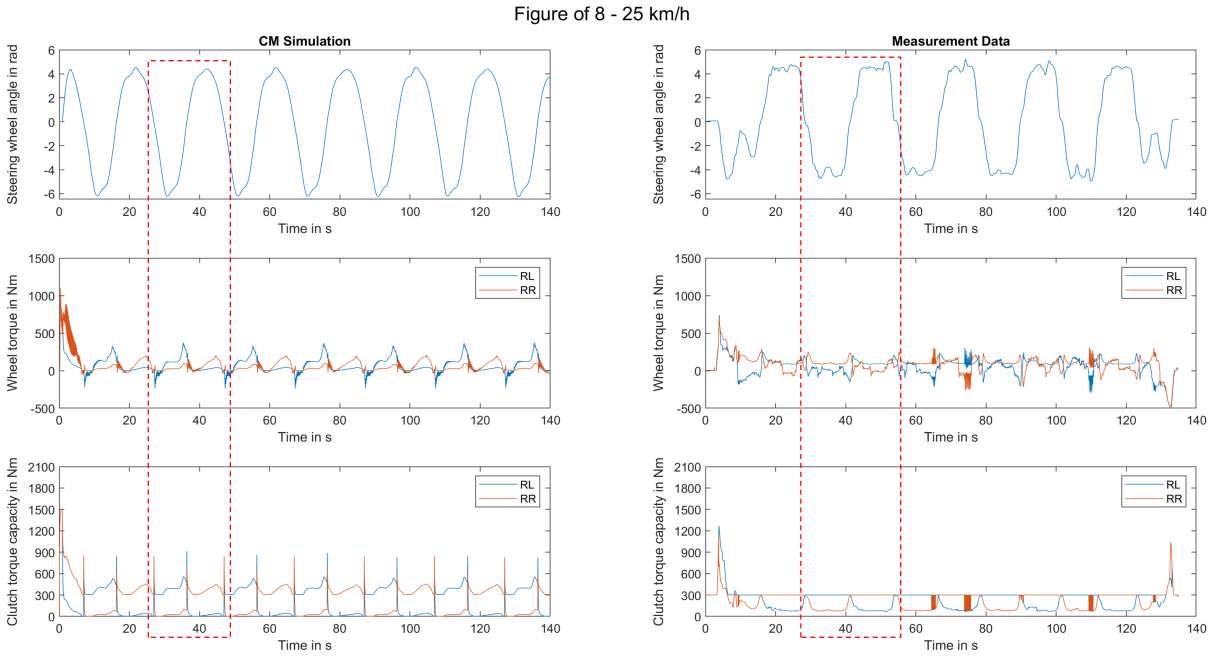

Figure 4.8: Steering wheel angles, wheel torques and clutch capacities from CM simulation using Controller B and measurement data for the 25 km/h test carried out on the figure of eight track. The areas of interest for the comparison are enclosed in red dashed-line boxes

On studying the area of interest enclosed by the red-dashed line box for the CM simulation, it is seen that the outer wheel torque is higher than the inner wheel torque for both the right turn (time $\approx$ 30s) and the left turn (time $\approx$ 45s), which follows that the outer wheel clutch has a higher torque capacity in both the cases, based on the control logic.

On studying the corresponding data on the measurement data plot, for the right turn (time $\approx$ 35s), it is observed that, though the torque capacity of the outer wheel clutch is higher than that of the inner wheel clutch, the resulting outer (left) wheel torque is lower than the inner (right) wheel torque and the outer wheel torque also takes negative values. A similar trend is seen for the left turn data at time $\approx$ 50s. In comparison, the nature of the wheel torques is not the same between the CM simulation and measurement data for the same nature of steering wheel angle or turn, even though the nature of clutch torque capacities is the same in both cases. To investigate this further, for the measurement data, a plot of the input torque delivered to the rear axle in time was compared against the clutch torque capacities. The steering wheel angle was also plotted in a separate plot for ease of comparison.

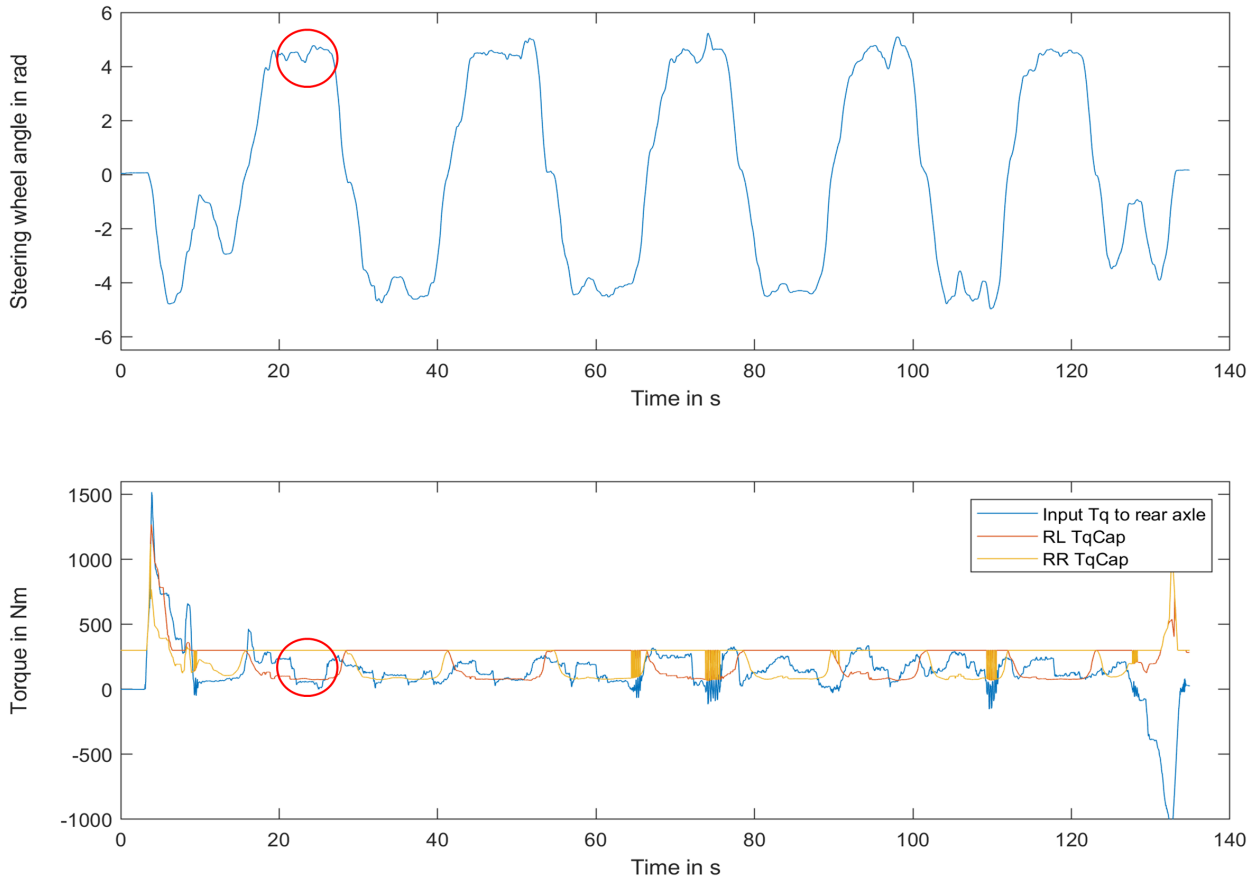

Figure 4.9: Steering wheel angle, input torque to rear axle and clutch torque capacities from measurement data for the 25 km/h test carried out on the figure of eight track. The areas of interest are encircled in red

From the measurement data plot, it is seen that at time $\approx$ 25s, for a positive steering wheel angle (left turn), though the inner wheel clutch is released, the input torque to the rear axle is lower than the torque capacity of the inner wheel clutch, and hence the inner wheel torque becomes equal to the inner clutch capacity, resulting in a negative torque at the outer wheel in order to ensure torque balance at the axle ($T_{axle} = T_{left} + T_{right}$). This kind of torque distribution at the wheels makes the vehicle 'odd' to drive in the opinion of the test driver, as there is a negative torque at one of the wheels which in a way 'brakes' it. However, this is a physically probable situation and hence the model needs to give a similar response as that seen in the track measurement data. In the above plot, even in instances where the net rear axle torque is higher than the inner clutch capacity, the difference between them is quite low and hence, most of the axle torque is transmitted to the inner wheel, while the remainder is transmitted to the outer wheel, resulting in greater torque on the inner wheel.

This distribution though undesirable, was a result of the inner wheel clutch not being released 'enough', though the intention was to distribute a greater torque to the outer wheel. This could have been resolved by modifying the control of the clutches in the test vehicle. However, since the focus of the thesis work was to model the system and validate it, it was desirable to extract a similar behaviour from the model, by modifying the control.

4.2.1.3 Modification in clutch control logic

In order to modify the control, the clutch control signals were observed. From figure 4.8, it was seen that the clutch torque capacities do not match, though they are similar in nature. If the same kind of clutch capacities input caused the same kind of wheel torques, in both CM simulation and measurement data, the model could be effectively validated for its performance. Using this idea, the clutch capacities of the CM simulation were made similar to that of the vehicle measurement data by modifying the clutch control logic in the CM model

to the strategy in 'Controller C'. This is the control logic that will be used for the comparisons between CM simulation data and the vehicle measurement data, henceforth.

4.2.1.4 25 km/h manoeuvre - Controller C

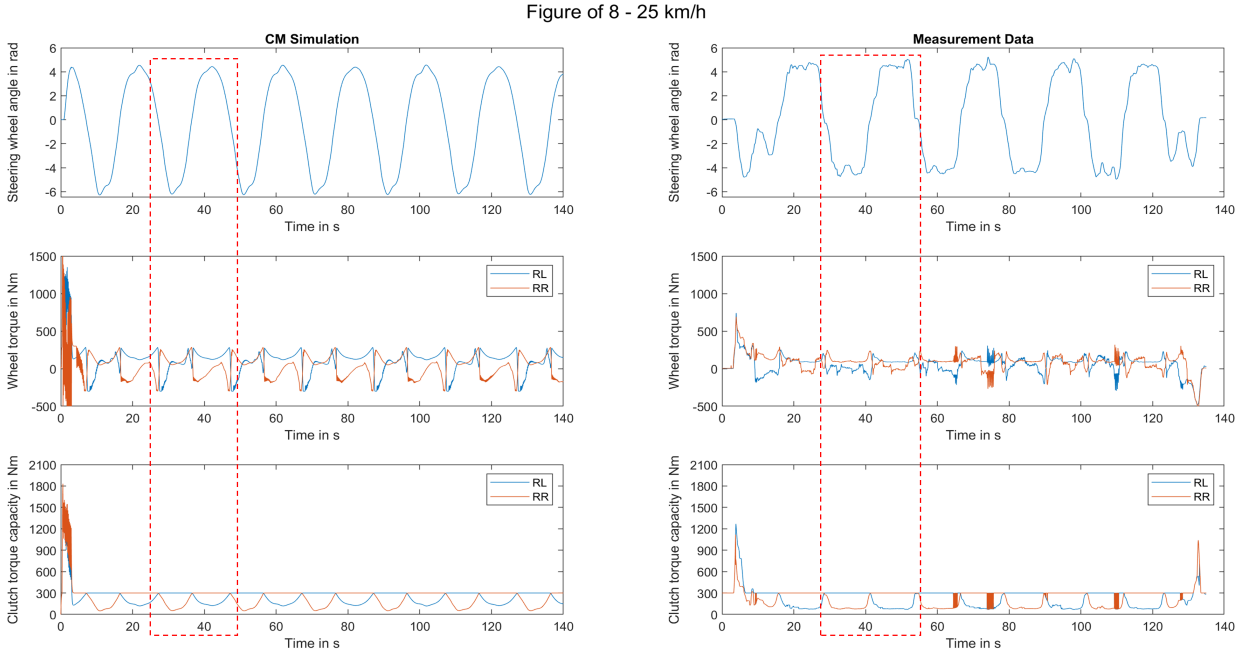

Figure 4.10: Steering wheel angles, wheel torques and clutch capacities from CM simulation using Controller C and measurement data for the 25 km/h test carried out on the figure of eight track. The areas of interest for the comparison are enclosed in red dashed-line boxes

Figure 4.10 shows the steering wheel angles, wheel torques and clutch capacities compared between the CM simulation and measurement data with the modified control strategy. On comparing the data at time $\approx$ 30s on the CM simulation plots and at time $\approx$ 35s on the measurement data plots, it can be seen that for a negative steering wheel angle (right turn), the inner wheel clutch is released and the drive torques are of the same nature in both plots. The inner wheel torque is slightly higher than the outer wheel torque at most points during the manoeuvre. It is further seen that the clutch torque capacities in the simulation plot are quite similar to the measurement data with the new controller - Controller C, when compared to the clutch capacities in the simulation plots shown in figure 4.8, which used Controller B. This similarity in the control signals results in a similar nature of wheel torques between the CM simulation and measurement data. Comparing the data at time $\approx$ 45s on the CM simulation plots and at time $\approx$ 50s on the measurement data plots, gives a similar result for the left turn manoeuvre.

4.2.1.5 35 km/h manoeuvre - Controller C

The figure of eight test was simulated again at 35 km/h in CarMaker, to check whether Controller C would give the same kind of results, though Controller B already did. But Controller C was more advanced as it ensured that the nature of the clutch capacities from the CM simulation was quite similar to the measurement data plots when compared to the clutch signals from Controller B as shown in figure 4.6.

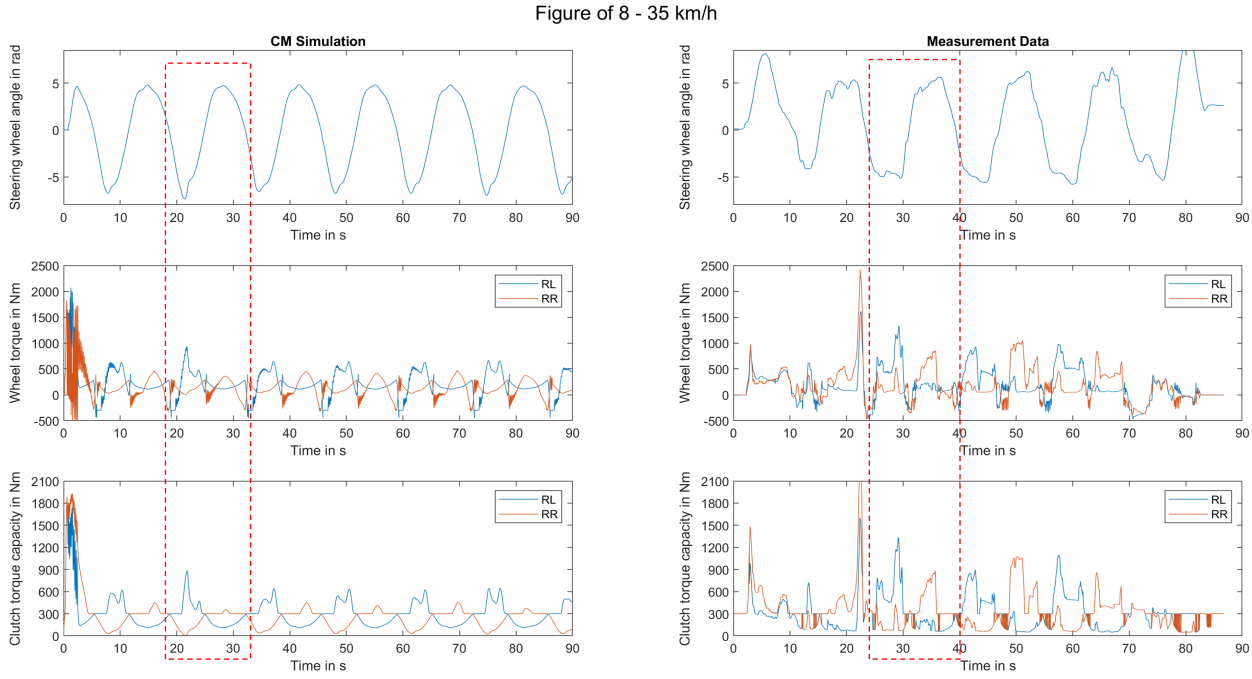

Figure 4.11: Steering wheel angles, wheel torques and clutch capacities from CM simulation using Controller C and measurement data for the 35 km/h test carried out on the figure of eight track. The areas of interest for the comparison are enclosed in red dashed-line boxes

On comparing the data at time ≈ 23s on the CM simulation plots and at time ≈ 28s on the measurement data plots, it can be seen that for a right turn, the inner wheel clutch is released and the drive torques are of the same nature in both plots. A similar result is seen for the left turn at time ≈ 28s on the CM simulation plots and at time ≈ 35s on the measurement data plots.

4.2.2 Handling track 1

Handling track 1 is a track in the Volvo Cars Proving Ground at Hällered. It has 'smooth' curves on the track, which makes it easier to achieve steady-state speeds while driving.

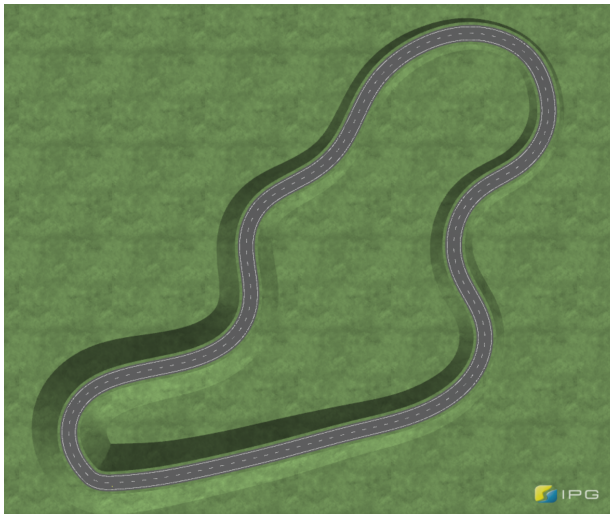

Figure 4.12: Handling track 1

4.2.2.1 45 km/h manoeuvre

The speed profiles for the simulation in CM and the measurement data are shown in figure 4.13.

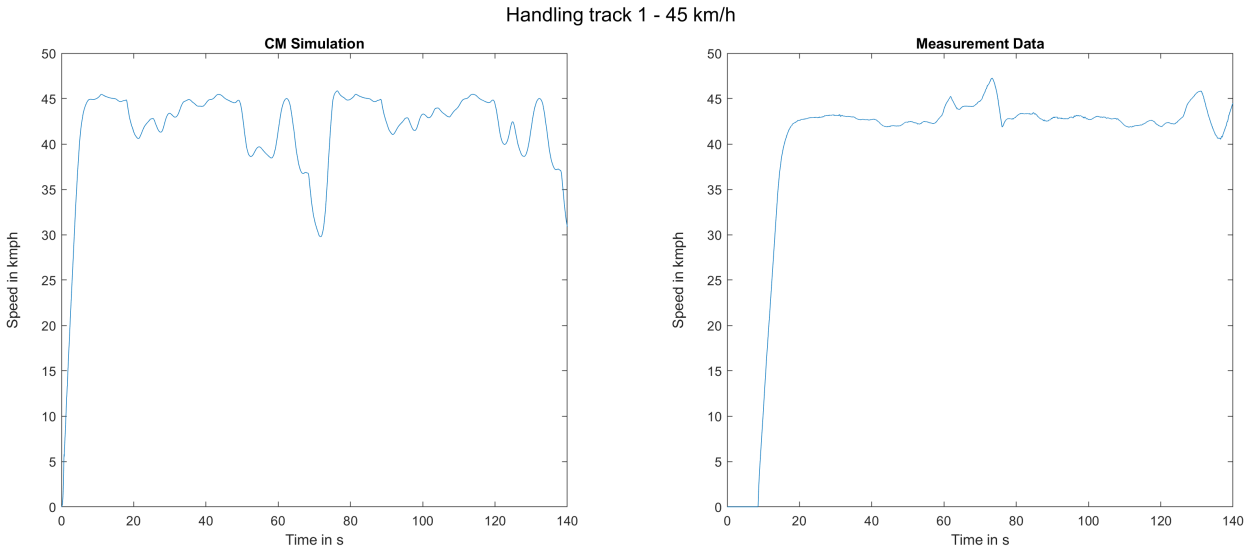

Figure 4.13: Speed profiles from CM simulation and measurement data for the 45 km/h test carried out on Handling track 1

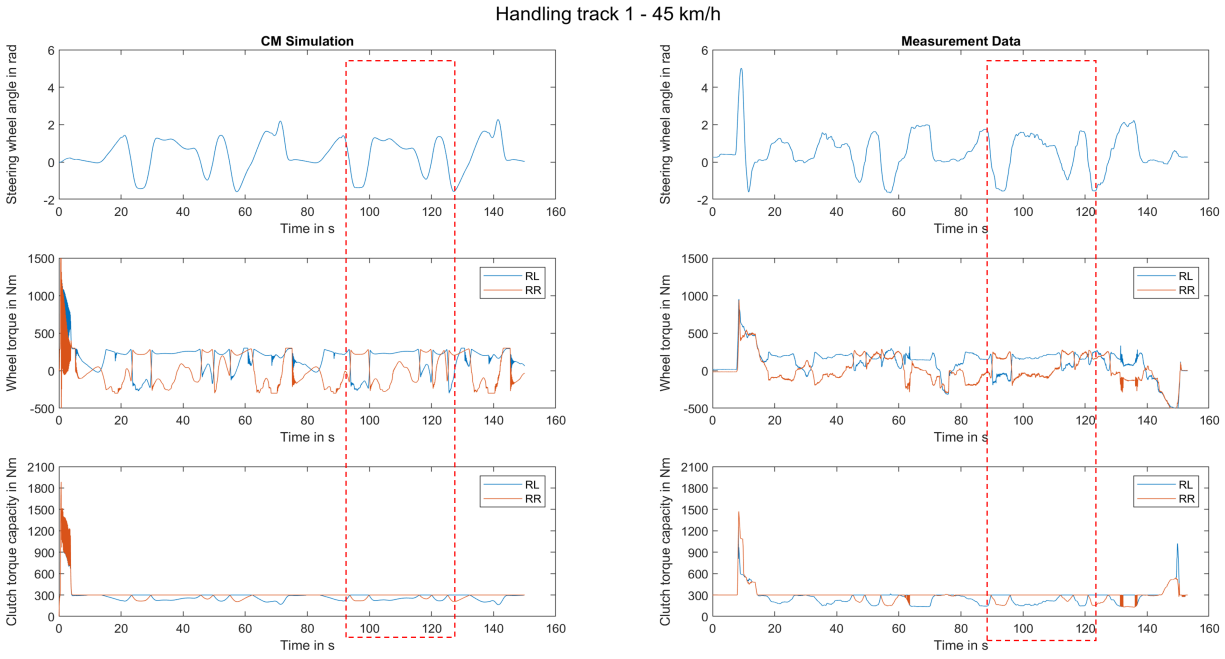

Figure 4.14: Steering wheel angles, wheel torques and clutch capacities from CM simulation and measurement data for the 45 km/h test carried out on Handling track 1. The areas of interest for the comparison are enclosed in red dashed-line boxes

Comparing the data in the area enclosed by the red dashed-line boxes, shows that for similar steering wheel angles, the clutch capacities and the wheel torques are similar for the CM simulation plots and the measurement

data plots.

4.2.2.2 50 km/h manoeuvre

The speed profiles for the simulation in CM and the measurement data are shown in figure 4.15.

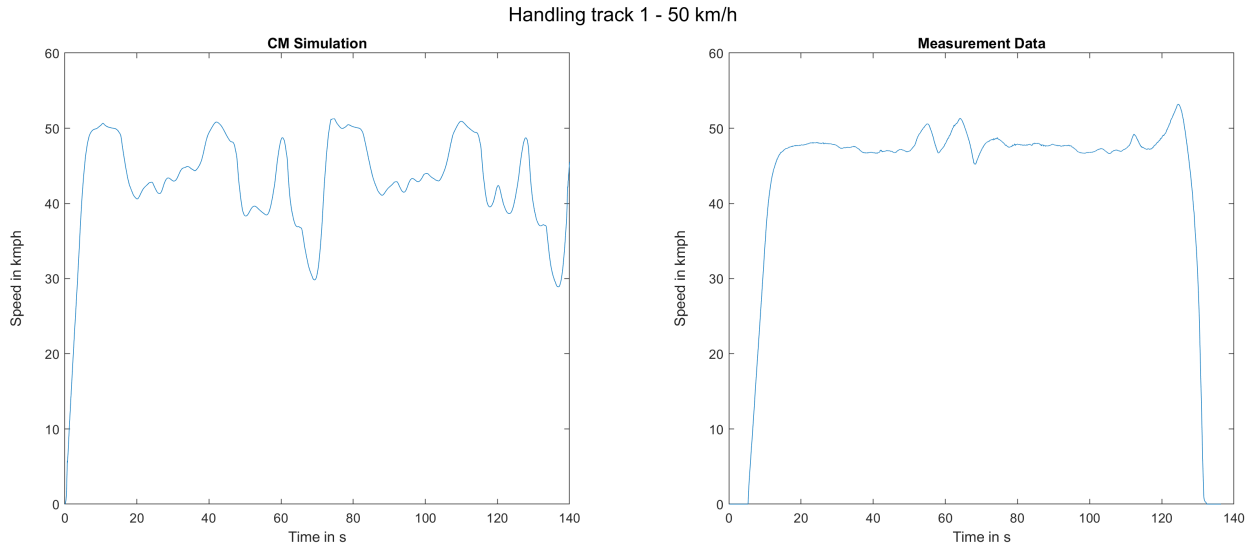

Figure 4.15: Speed profiles from CM simulation and measurement data for the 50 km/h test carried out on Handling track 1

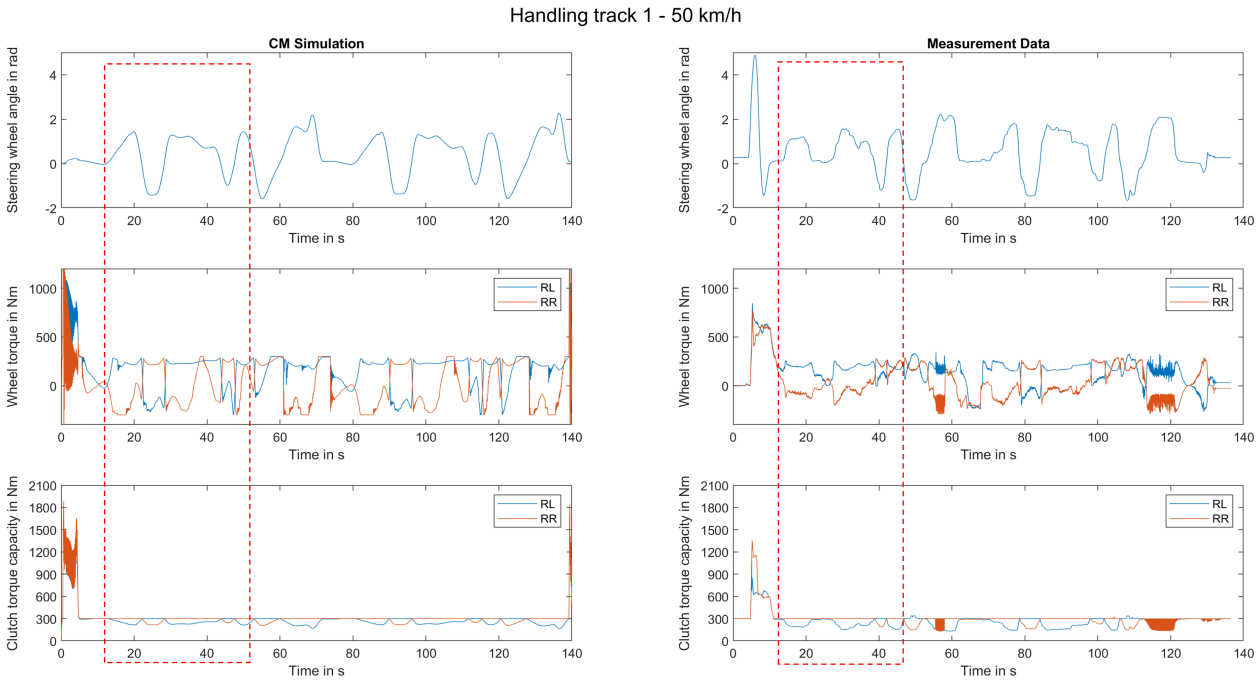

Figure 4.16: Steering wheel angles, wheel torques and clutch capacities from CM simulation and measurement data for the 50 km/h test carried out on Handling track 1. The areas of interest for the comparison are enclosed in red dashed-line boxes

Comparing the data in the area enclosed by the red dashed-line boxes shows that for similar steering wheel angles, the clutch capacities and the wheel torques are similar for the CM simulation plots and the measurement data plots.

4.2.3 Handling track 2

Handling track 2 is another prominent track used for testing cars at Hällered. This track consists of a combination of sharp turns and steady-state corners.

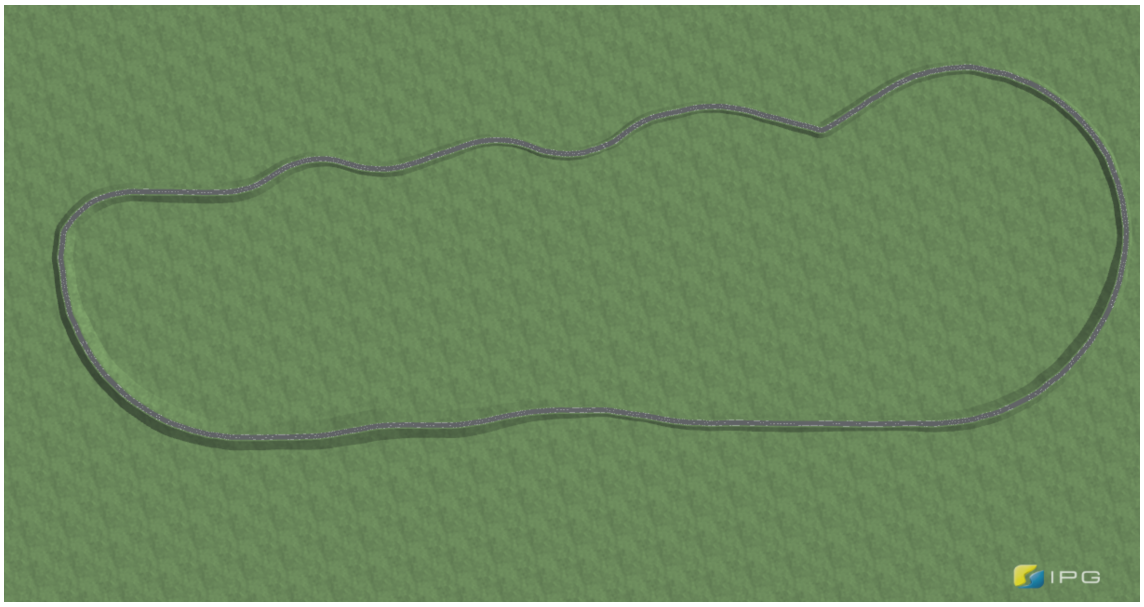

Figure 4.17: Handling track 2

4.2.3.1 70 km/h manoeuvre

The speed profiles for the CM simulation and the measurement data at 70 km/h are shown in figure 4.18.

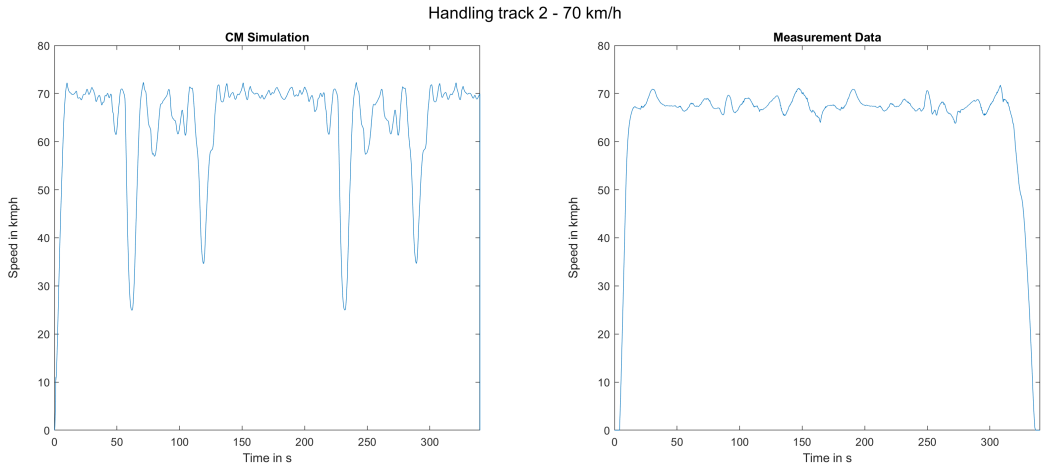

Figure 4.18: Speed profiles from CM simulation and measurement data for the 70 km/h test carried out on Handling track 2

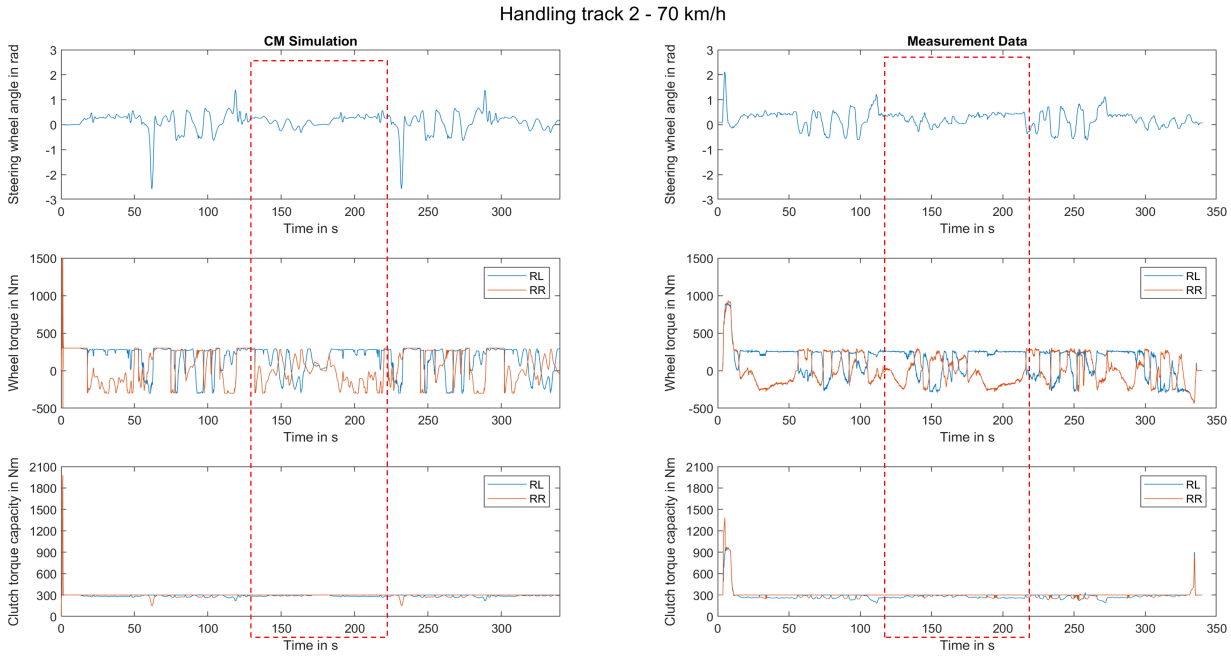

Figure 4.19: Steering wheel angles, wheel torques and clutch capacities from CM simulation and measurement data for the 70 km/h test carried out on Handling track 2. The areas of interest for the comparison are enclosed in red dashed-line boxes

Comparing the data in the area enclosed by the red dashed-line boxes shows that for similar steering wheel angles, the clutch capacities and the wheel torques are similar for the CM simulation plots and the measurement data plots.

4.3 Results from the comparison of TVDC system and open differential in simulation environment

The power consumed by the electric motor was plotted for the open differential model and the TVDC model to compare the performance. The steering effort was also compared by plotting the steering wheel angles for both the models for the same manoeuvres. The power consumed was calculated in the Simulink model as the sum of mechanical power produced ($T_{actual} \cdot \omega_{rotor}$) and the motor losses.

4.3.1 Figure of eight track

For the figure of eight track the results of the comparison are shown in figure 4.20. It can be noted that for the power consumption values, a negative sign indicates power consumed and a positive sign indicates power retrieved (say from regeneration).

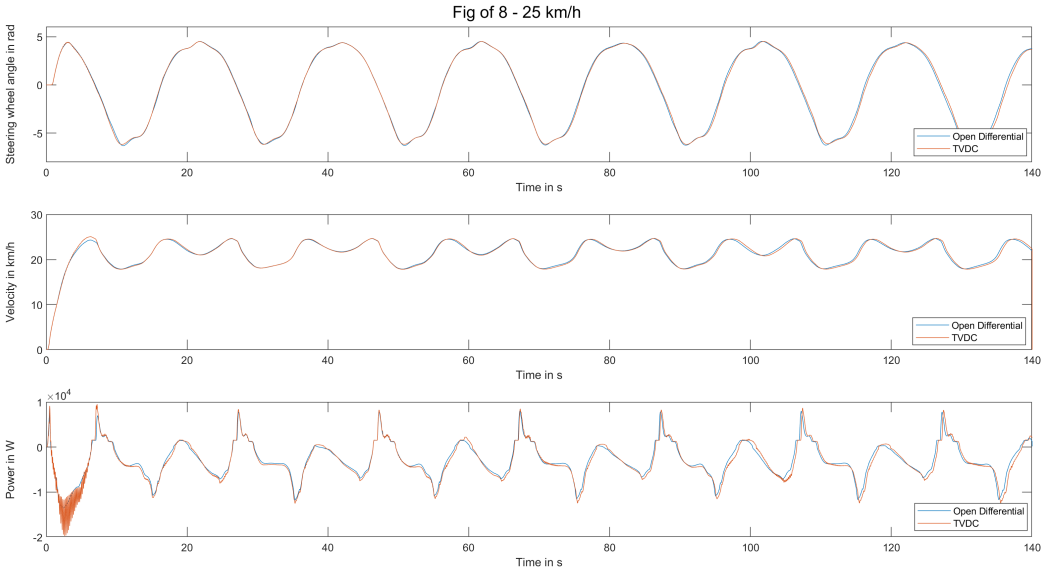

Figure 4.20: Comparison of steering wheel angle and power consumption for the open differential and TVDC systems on the Figure of eight track at 25 km/h

The energy consumption for the open differential model was 1.846×10^5J and the energy consumption for the TVDC model was 2.027×10^5J.

4.3.2 Handling Track 1

For Handling Track 1 the results of the comparison are shown in figure 4.21.

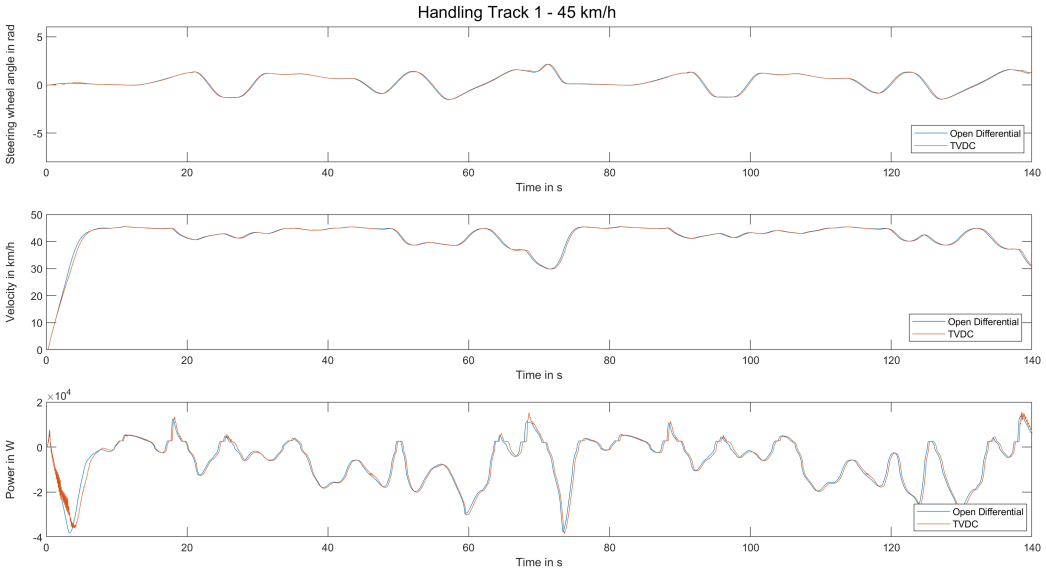

Figure 4.21: Comparison of steering wheel angle and power consumption for the open differential and TVDC systems on Handling Track 1 at 45 km/h

The energy consumption for the open differential model was 9.248×10^5J and the energy consumption for the

TVDC model was 9.605×10^5J.

From the figures 4.20 and 4.21, it can be seen that there is no significant difference in the power consumption and steering effort between the open differential model and the TVDC model. Similar tests were conducted for Handling track 2 and the Göteborg city cycle and the results were similar. The Göteborg City Cycle is a standard dynamic driving cycle used by Volvo Cars consisting of accelerations, decelerations, constant speeds and start-stop manoeuvres. A study of the effect of the inertia in the model on power consumption was done and the results are shown below.

Table 4.1: Energy consumption for open differential and TVDC for different driving cycles

Energy in Joule	Open Differential	TVDC $J = 0.1\ kgm^2$	TVDC $J = 0.08\ kgm^2$	TVDC $J = 0.03\ kgm^2$
Figure of 8 - 25kmph	1.846×10^5	2.027×10^5	2.024×10^5	1.996×10^5
Handling Track 1 - 45kmph	9.248×10^5	9.605×10^5	9.592×10^5	9.559×10^5
Handling Track 2 - 60kmph	2.059×10^6	2.053×10^6	2.051×10^6	2.043×10^6
Göteborg City Cycle	1.69×10^7	1.743×10^7	1.74×10^7	1.732×10^7

From the above table, the effect of the moment of inertia J (sum of moments of inertia of the motor's rotor and the shaft) on the energy consumption can be observed. Since these driving cycles start from rest and end at some constant speed, the torque used to accelerate the inertia might not be fully recovered in the cycle and hence a clear trend of decrease in energy consumption as the moment of inertia decreases is seen in the TVDC models.

4.4 Update on library model

The Modelica standard library-based model with its update from section 3.2.8 was used with the CM vehicle model to run a simulation on Handling Track 1 and compare its performance with that of the equation-based model for the same manoeuvre. The results are shown in figure 4.22.

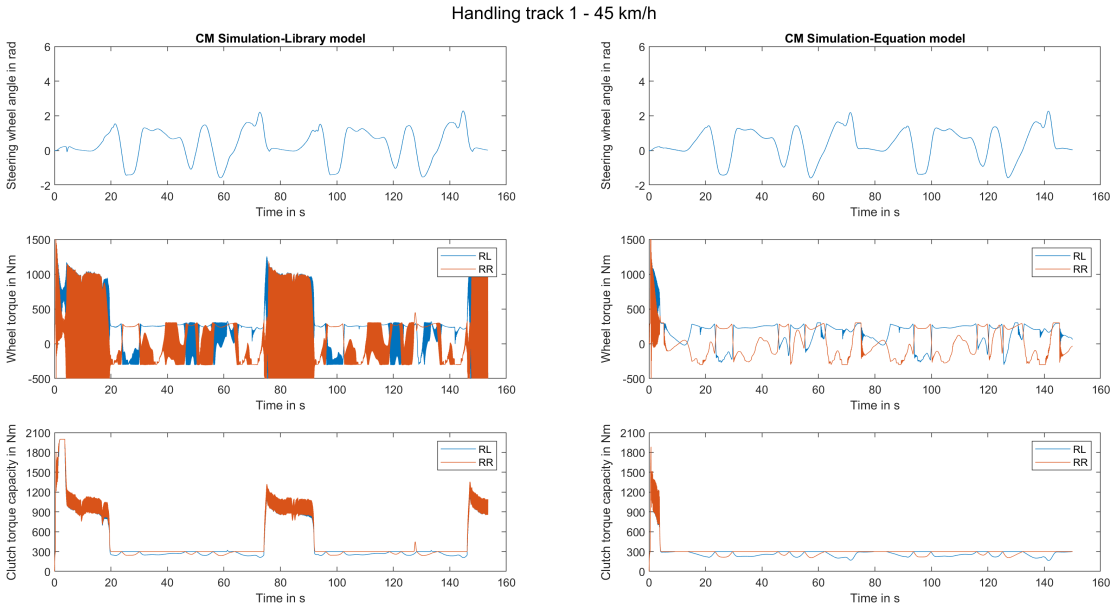

Figure 4.22: Comparison of steering wheel angles, wheel torques and clutch capacities for the updated library-based model and equation-based model of the TVDC system on Handling Track 1 at 45 km/h

From the above plots, it can be seen that though the nature of the wheel torques in both the models is quite similar, the inertia added to the updated library model results in chattering (high oscillations) in the wheel torques.

5 Conclusions and future work

5.1 Conclusions

The modelling of the TVDC system using the equation-based approach on Modelica shows greater flexibility as compared to the library-based approach, which did not work when modelled according to the component structure of the desired physical model. Modelica clutch does not seem to work without the small inertias after the clutches. On adding these small inertia elements on the driveshaft side of the clutches, the inertias give chattering (or at least high eigenfrequencies) when the clutches are in slip state and result in chattering in the output torques as well. Hence it is concluded that the equation-based model is convenient to model the systems involving discrete state events such as the TVDC driveline.

The Modelica model of the TVDC driveline is found to be robust and easily controlled (can be controlled by any desired control strategy easily). The model performance is validated by comparison with measurement data and the nature of torque distribution is in agreement with the measurements for similar clutch control signals in identical driving scenarios, under steady-state (constant speed) driving conditions. Three control strategies were devised for controlling the clutches based on the measurement data that was obtained, and changes had to be made to the initial control strategy to match the control signals in the measurement data. These control signals though not optimal were needed for model validation. It was found that the inner wheel clutches were not released enough in the test vehicle's TVDC control while cornering and this was identified by observing the nature of the torque distribution in the rear axle. This also explained why the vehicle behaved strangely during cornering according to the test driver.

There was no information about the clutch control logic used in the test vehicle and a 'guesstimate' was made by observing the clutch capacity signals from the measurement data to devise the control strategy for the TVDC model. It would be very useful to have more information about the control strategy used in the TVDC test vehicle used for measurements on the track, and this knowledge would help in running simulations using the TVDC model to further validate the model, and also simulate dynamic driving scenarios.

The comparison of the power consumption and steering effort between the open differential and the TVDC driveline in the simulation environment showed that there is no significant difference between the values for the two models. Generally, the TVDC model consumes more energy as compared to the open differential model, but there are instances in the analysis, where the TVDC model's power consumption is better. The control strategy used for the TVDC system was basic and the possibility of having an advanced control logic to see some improvements in energy comparison or steering effort (steering angle) compared to the open differential driveline cannot be ruled out. At this point, no substantial conclusions shall be drawn from the performance comparison of the TVDC driveline with the open differential, since the control strategy used for the TVDC system is quite primitive and is probably insufficient to show any benefits of using the TVDC driveline. However since the focus was on modelling the TVDC driveline and not to prove that it functioned better than conventional drivelines, it was not investigated further.

In conclusion, the physical, equation-based modelling approach using Modelica proposed in this thesis work, can be used as an early 'virtual development' tool for driveline modelling and simulation and this is verified by the comparisons with real vehicle tests. As a first attempt, the library-based modelling approach can be explored to see if it works as it is simpler, however too much time should not be spent on this as the equation-based modelling approach usually gives a working model. This 'virtual development' tool can be used for performance comparisons with conventional systems with the right control strategy.

5.2 Future work

The robustness of the TVDC system has been established and the model has been validated for steady state driving conditions. Further validation can be carried out by simulating the model for dynamic driving conditions and comparing the same with the measurement data. For this, the speed profile from the measurement data can be uploaded in the Manoeuvre section of CarMaker as explained in the section 'Input From File / Using Real World Measurements' in the CarMaker User's Guide [19]. This would however require knowledge of the clutch control strategy used in the test vehicle to make the results from the simulations plausible.

It would be highly beneficial to compare the simulation results from the TVDC CM model with measurement data obtained from a vehicle with a better-tuned control strategy. With this knowledge of an advanced control strategy for the TVDC driveline, it can be investigated whether the TVDC driveline can be used to reduce energy consumption and/or steering effort compared to conventional drivelines. Co-control of the TVDC torque vectoring function with Electronic Stability Control (ESC) and Traction Control (TC) to improve vehicle handling and performance can also be studied.

The TVDC system can be evaluated in longer cycles, such as the GCC and certification cycles, for torque vectoring function. Additional control strategies can be developed for the control of the TVDC system, in order to reduce power consumption or steering effort. A study of the power required for actuation of the clutches can be conducted to ensure that the overall energy consumption of the vehicle is not beyond the allowable bounds, for a given gain in performance - this can always be a trade-off.

References

[1] *Full User Manual*. Dymola 2021x. 2020.

[2] *Modelica by Example*. Modelica by Example. URL: https://mbe.modelica.university/ (visited on 03/22/2021).

[3] M. M. Tiller. *Clutch Modeling - Part I: Friction Clutches*. Xogeny Blog. Jan. 1, 2013. URL: http://blog.xogeny.com/blog/part-1-friction/ (visited on 03/15/2021).

[4] H. Xu. "Electric Power Steering Model Development for Virtual Validation Environment". MA thesis. 2016.

[5] A. Bhat and A. V. Malghan. "Modelling of propulsion system for complete vehicle verification through simulation". MA thesis. 2016.

[6] A. Stoop. "Design and implementation of torque vectoring for the forze racing car". PhD thesis. Delft University of Technology, 2014.

[7] A. Schaaf. "A modular Torque Vectoring System for a 4WD Electric Performance Vehicle". MA thesis. 2018.

[8] M. Grahovic and M. Rosicki. Development and Evaluation of a Torque-Vectoring Algorithm on RWD Racing Cars using a Dual Clutch (2019).

[9] B. Jacobson. *Vehicle dynamics compendium for course MMF062; edition 2016*. Tech. rep. Chalmers University of Technology, 2016.

[10] P. Fritzson. *Introduction to modeling and simulation of technical and physical systems with Modelica*. John Wiley & Sons, 2011.

[11] *Functional Mock-up Interface*. URL: https://fmi-standard.org/ (visited on 05/31/2021).

[12] *modelon-community/SEMLA*. original-date: 2019-05-29T10:42:37Z. Jan. 13, 2021. URL: https://github.com/modelon-community/SEMLA (visited on 05/31/2021).

[13] *Clutch. Wikipedia*. Page Version ID: 1024663864. May 23, 2021. URL: https://en.wikipedia.org/w/index.php?title=Clutch&oldid=1024663864 (visited on 06/01/2021).

[14] J. Vogel. *Tech Explained: Ackermann Steering Geometry*. Racecar Engineering. Section: Articles. Apr. 6, 2021. URL: https://www.racecar-engineering.com/articles/tech-explained-ackermann-steering-geometry/ (visited on 06/16/2021).

[15] *IPG Documentation, CarMaker Reference Manual*. Version 9.0.2. IPG.

[16] *Modelica*. URL: https://doc.modelica.org/Modelica%204.0.0/Resources/helpOM/Modelica.html (visited on 05/31/2021).

[17] *Modelica, Requirements for simulation tools*. URL: https://www.ida.liu.se/~adrpo33/modelica/src/binary/ModelicaXML/Modelica_1.5_XML/help/Modelica_Mechanics_Rotational.html (visited on 05/30/2021).

[18] *IPG Documentation, Programmer's Guide*. Version 9.0.2. CarMaker. IPG.

[19] *IPG Documentation, User's Guide*. Version 9.0.2. CarMaker. IPG.

A Appendix

A.1 Export FMU from Dymola

With the required model to be exported open in the Simulation tab on Dymola, choose Translate > FMU, as shown in figure A.1.

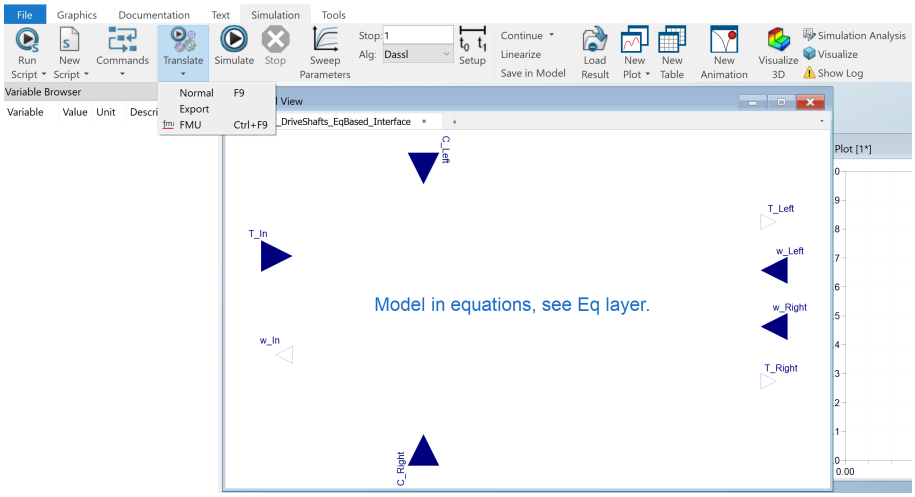

Figure A.1: Translate model as FMU in Dymola

In the next dialog box that appears, select the type of FMU for export - Model exchange or Co-simulation, or if unsure, it is good to choose the Model exchange and Co-simulation option (refer figure A.2) and choose the FMU type while importing the FMU on the other tool. It is recommended to use the latest FMI version 2.0, and also choose a name for the FMU in the model identifier and press 'OK' (the other options are usually preset and need not be changed). The Modelica translator will check the model for errors in the model and its interfaces, and translate the model into an FMU. Any errors will be shown in the 'Logs' tab of the Dymola User Interface.

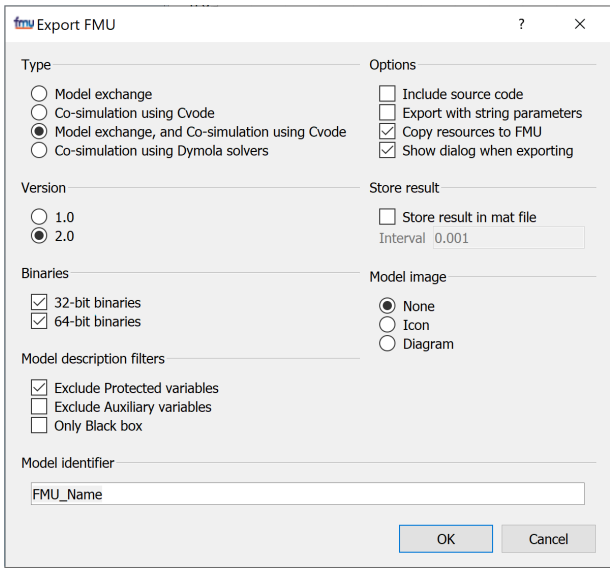

Figure A.2: Export FMU dialog box in Dymola

As a standard practice, it is recommended to use the 'Check' option under the 'Graphics' or 'Text' tabs in the Dymola toolbar, to check for basic errors in modelling, before simulation or exporting as an FMU. The FMU is stored in the same working directory as that of the current model or package.

A.2 Import FMU in Simulink

On the Simulink window, double-click on the empty space, and type 'FMU'. Choose 'FMU' in the suggestions that are shown.

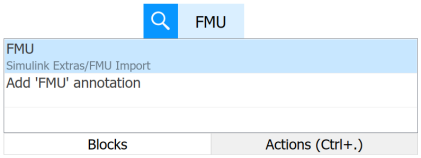

Figure A.3: Import FMU in Simulink

The block requires the path for opening the FMU file. It is recommended to have the FMU file in the same directory as the CM Simulink files. If the FMU being imported supports both Model exchange and Co-simulation, Simulink will ask the user to choose one of the options for importing.

A.3 CM Driver model

CarMaker allows the user to parameterize its inbuilt driver model. This can be done in the 'Maneuver' section of the CM window. In the Maneuver window, under 'Longitudinal Dynamics', ensure 'IPG Driver' is chosen and select the 'Driver Parameter' button, to open up the window shown in figure A.4.

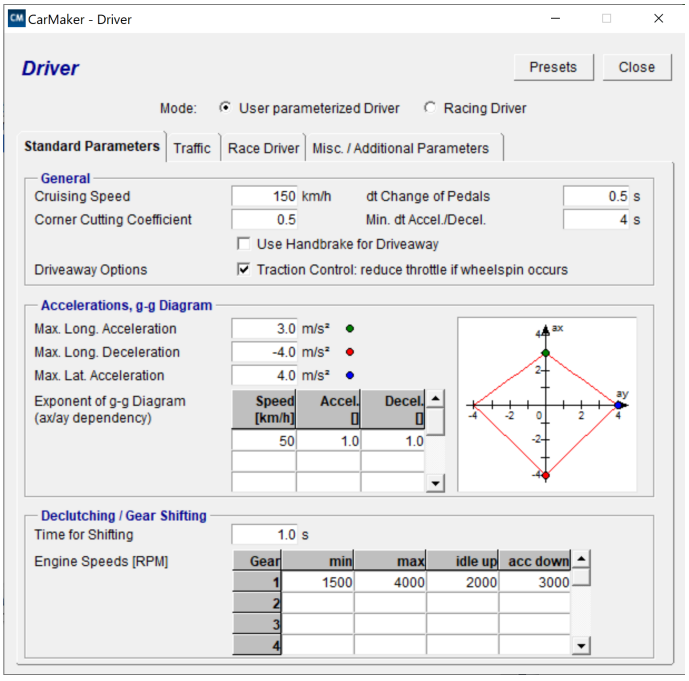

Figure A.4: CM Driver Parameter Window

The CM Driver model can be parameterized by choosing deceleration limit, lateral and longitudinal acceleration limits and other parameters. The Driver model can thus be adapted to a driving style by the user. The Maneuver dialog box allows the user to choose a speed and duration for the manoeuvre in a sequence. Refer to CM User's Guide[19] for more information.

A.4 CM Road from GPS data

CM allows the user to create tracks for testing the vehicle systems from GPS data collected on track. The GPS data from Google Maps or any other source needs to be in .kml file format. In the CM window, go to Parameters > Scenario/Road and in the Scenario Editor, under Road > Road segment, choose 'File' as shown in figure A.5.

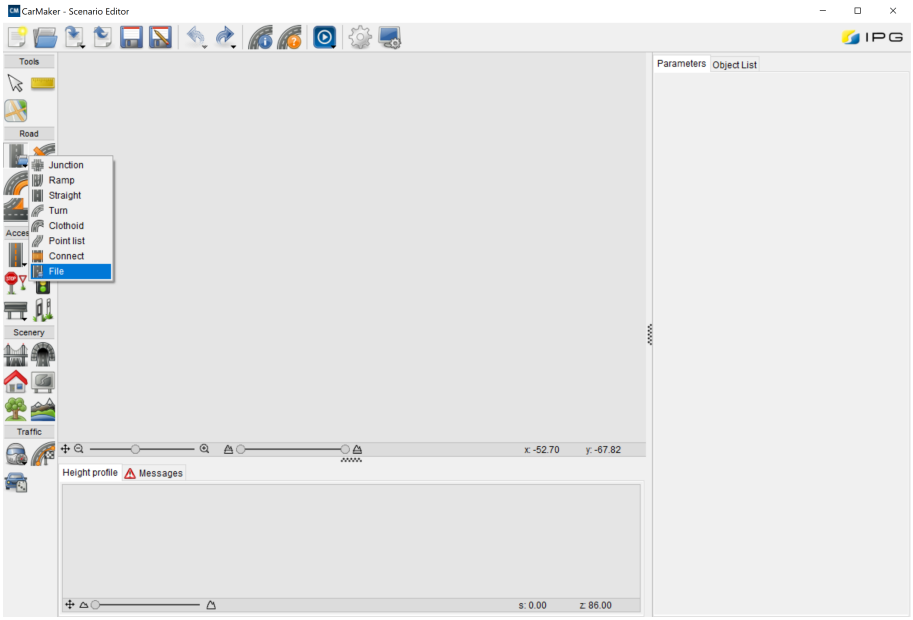

Figure A.5: CM Road Scenario Editor

Click on the grey area of the screen where the track is supposed to start, click again to open a window where the .kml file can be chosen (figure A.6).

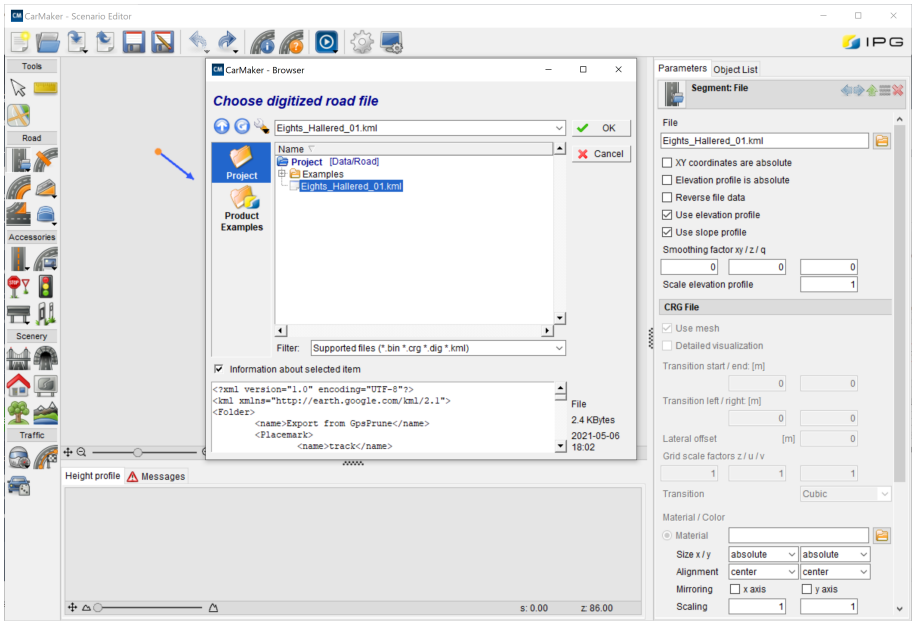

Figure A.6: CM Road Scenario Editor

Any discontinuities in the track from the .kml file can be fixed by choosing Road > Road segment > Connect and then by choosing the two open points to be connected. To define the lane in which the manoeuvres are simulated, under 'Traffic', choose 'Route' and select a lane on the track (shown in green in figure A.7). To change the starting point of the vehicle on the track, go to Scenario settings > Longitudinal offset and set it to the desired value in 'm'.

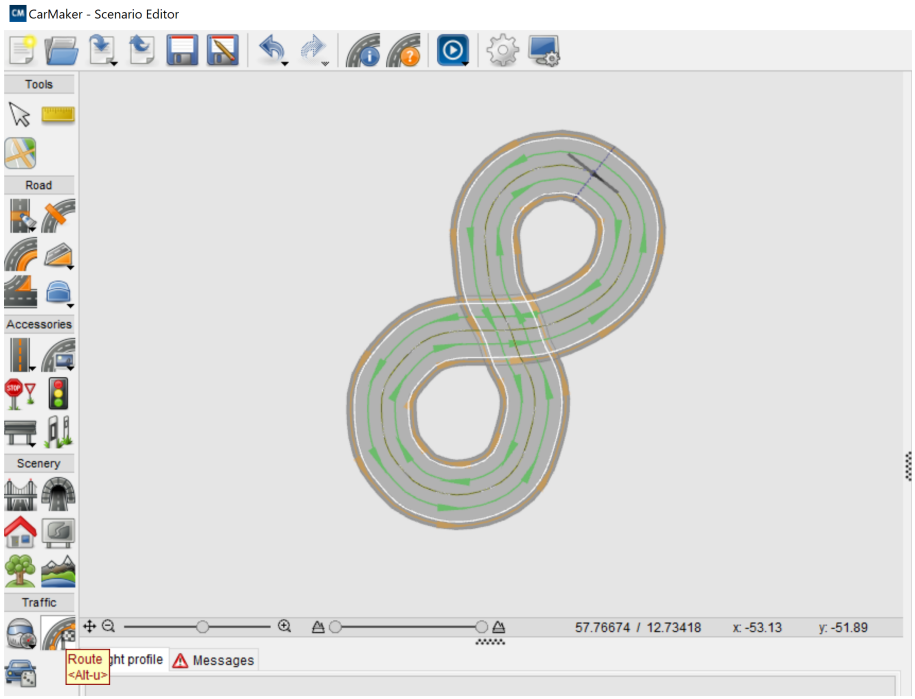

Figure A.7: CM Road Scenario Editor route selection

Use the 'Save road file as' button to save the file in '.rd5' format on CarMaker. Note that it is necessary to have the .kml file in the same directory as the .rd5 file while using the Road file for CM simulations.

DEPARTMENT OF MECHANICS AND
MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY