LAPPEENRANTA UNIVERSITY OF TECHNOLOGY
LUT School of Energy Systems
LUT Mechanical Engineering

Rehan Niazi

GENERALIZED MODELING APPROACH FOR REAL-TIME SIMULATION

Updated: 24.04.2018

Examiners: Professor Jussi Sopanen
D.Sc. (Tech.) Janne E. Heikkinen

Supervisor: M.Sc. (Tech.) Suraj Jaiswal

ABSTRACT

Lappeenranta University of Technology
LUT School of Energy Systems
LUT Mechanical Engineering

Rehan Niazi

Generalized modeling approach for real-time simulation

Master's thesis

2018

77 pages, 39 figures, 6 table and 5 appendices

Examiners: Professor Jussi Sopanen
D.Sc. (Tech.) Janne E. Heikkinen

Supervisor: M.Sc. (Tech.) Suraj Jaiswal

Keywords: Generic model, Vehicle model, Real-time simulation, Simulation, User-parameterization, Modeling configurator, Graphical user-interface.

The test drive of a customized vehicle by the customer was not possible prior to the existence of the simulators. Thus, leaving a customer to trust the industry and make compromises while selecting a vehicle. Although, simulators provide a virtual experience of a vehicle, but all the available vehicle models are preconfigured and hence the user can only experience those. The main idea behind this research is to enable the user to experience a self-customized vehicle and communicate the will of customer directly to the manufacturer. Therefore, the future prospect of this research is to merge both customer and manufacturer, during the innovation phase of a development project by developing a tool (modeling configurator) that can serve the purpose.

Since the simulation environment in this research is provided by Mevea Solver, which create model simulation file in XML format, hence, leading to the editing of preprogrammed XML files. This is achieved by developing a generic modeling configurator by the use of object-oriented programming in Matlab editor environment. Also to make the modeling configurator more user-friendly, App building function is utilized to create a user-interactive graphical user interface. The attributes selected for customization are: engine modeling (using polynomial equation), tire modeling (using section width and aspect ratio) and gearbox modeling (using geometric ratios). The algorithms for modeling of these attributes are based on generic mathematical model extracted from literature, whereas, the algorithm for modification of XML files is also generic in nature and can be applicable to any vehicle. In other words, to carry out a generalized approach for any class of vehicle, this research includes three different type of vehicle models: tractor, wheel loader, and fork lifter as case studies.

ACKNOWLEDGEMENTS

I start in the name of Allah, the most gracious and the most merciful, who gave me the strength to accomplish this task. I am grateful to my mentor Professor Jussi Sopanen for believing in me and providing me this opportunity to be a part of this prestigious project. I learnt a lot under his supervision. I would like to thank Janne E. Heikkinen, for his guidance and motivation all along and encouraging me in exploring different dimensions of this research. Also, I would like to thank Suraj Jaiswal for his valuable feedback in improving this research work.

I will take this opportunity to specially thank my parents, who made me reach this far in life. Their support and prayers have always made me to belief in myself. I feel blessed to have them and pray their blessing last forever upon me. Also, I would like to express my deepest gratitude to Hina Rehan, who in difficult times has always been there with me through thick and thin.

ریحان نیازی

Rehan Niazi

Lappeenranta 24.04.2018

TABLE OF CONTENTS

APPENDIX

LIST OF SYMBOLS AND ABBREVIATIONS

AR	Aspect ratio of tire
b	Distance / force arm [m]
d_{in}	Inner diameter /rim diameter [m]
d_{out}	Outer diameter of tire [m]
d, e, f	Case specific power performance coefficients
F	Force [N]
F_c	Coulomb friction level [N]
F_s	Stiction force level [N]
(F/A)	Fuel to air ratio being injected into the cylinder per cycle
$g(v)$	Function of relative velocity
h	Height or width of tire [m]
I_{xx}	Moment of inertia along *X-axis* [kgm^2]
I_{yy}	Moment of inertia along *Y-axis* [kgm^2]
I_{zz}	Moment of inertia along *Z-axis* [kgm^2]
i	Index from first to last
k_w	Gain of velocity difference
l	Final gear index
m	Mass of tire [kg]
N	Angular speed [rev/s]
n_d	Transmission ratio
n_i	Gear ratio of i^{th} gear
n_l	Gear ratio of l^{th} gear
P	Power [kW]
P_i	Engine power performance coefficients [kW/ rpm^i]
P_m	Engine rated power [kW]
P_o	Maximum engine power [kW]
p_1, p_2, p_3	Power demand coefficients
$p(w)$	Power demand function
Q_{HV}	Heating value of fuel [MJ/kg]

RD	Rim diameter [m]
R_e	Radius of wheel [m]
SH	Section height [m]
SW	Section width [m]
T	Torque [Nm]
$\dot{T}$	Motor torque derivative [Nm/s]
T_o	Motor torque magnitude [Nm]
T_m	Maximum torque [Nm]
T_p	Torque at rated power [Nm]
V_d	Displaced volume [dm^3]
v	Relative velocity of rigid surfaces [m/s]
v_e	Velocity of vehicle [m/s]
v_s	Stribeck velocity [m/s]
w	Ratio of instantaneous to maximum engine speed
z	Deflection in bristles [m]
η_f	Fuel conversion efficiency
η_v	Volumetric efficiency
$\rho_{a,i}$	Density of air injected
σ_0	Stiffness [N/m]
σ_1	Damping coefficient
σ_2	Viscous friction
τ	Time constant [s]
ω_e	Engine speed [rpm]
ω_o	Motor rotational speed [rpm]
ω_p	Engine speed at rated power [rpm]
ω_t	Engine speed at maximum torque power [rpm]
ω_{min}	Minimum economic working speed [rpm]
ω_{max}	Maximum economic working speed [rpm]
ω_{ref}	Motor rotational reference speed [rpm]

GUI	Graphical User Interface
LUT	Lappeenranta University of Technology

NURBS	Non-uniform rational B-Spline
SIM	Sustainable product through simulation
XML	Extensible Markup Language

1 INTRODUCTION

In the automotive industry, development of a project on major scale means launching a new product. The new product can be comprised of either new and old manufacturing setup or a complete new manufacturing setup. In the development of such project, product evolution process is followed and can be dependent on several levels such as design level, design content, innovation level or a number of options (Julian 2009, pp. 1-8). In order to breakthrough market and acquire the attention of customers with a new product, the manufacturer needs to have something cutting edge. In this respect the invention of simulators have assisted the manufacturers in several ways. Simulators are nowadays used for several purposes such as training, marketing, testing and entertainment in the form of gaming. All main automotive enterprises are widely using virtual reality and simulators are being widely used in the form of virtual engineering during the design phase of vehicle development (Jiang 2011, p. 173). Thus, the use of simulators in modeling and simulation, enables the manufacturers of automotive industry in reshaping the development stage at both, conceptual and innovation level. Since the device is quite expensive, it is unaffordable to have only one vehicle model available on the simulator. This situation demands the need for having generic models.

In pre-simulator era, the customers will to experience a self-customized vehicle was not possible, before it was manufactured. Therefore, a customers had rely on the industry and make compromises while purchasing a vehicle. Also in that era, the customer had an indirect influence in the innovation process. Whereas, in the presence of simulators, company can use the customer as a direct tool in the innovation process. Since modeling and simulation enable us to understand and foresee things which could not be visible beforehand otherwise and can save disaster situations (Jan 2016, p. 40). The automotive industry is adopting the use of simulator quite rapidly because of their wide scope of use. Industries can now test the virtual model of new vehicles in simulators and avoid any discrepancies which might be disastrous once the product is manufactured. The use of simulators allows a designer to visualize the product from different dimensions, which helps in coping up with design related issues. In doing so, the total development time is reduced and hence less time is consumed in concept phase (Auweraer et al. 2008, p. 398).

1.1 Motivation

The main motivation behind this research, is highlighting the role of simulators in training purposes for different kinds of vehicles. Also it's worth mentioning here, that nothing can be more valuable than human life. According to the statistics as shown in Figure 1 and Figure 2, the rate of fatal injuries in the sectors of construction and agriculture are the highest among other industries. These statistics indicate that the indulgence of simulators and research in these industries is quite valuable, thereby, reducing the rate of fatal injuries and leading to a more sustainable environment. This lead to the inclusion of different class of vehicles as part of this research.

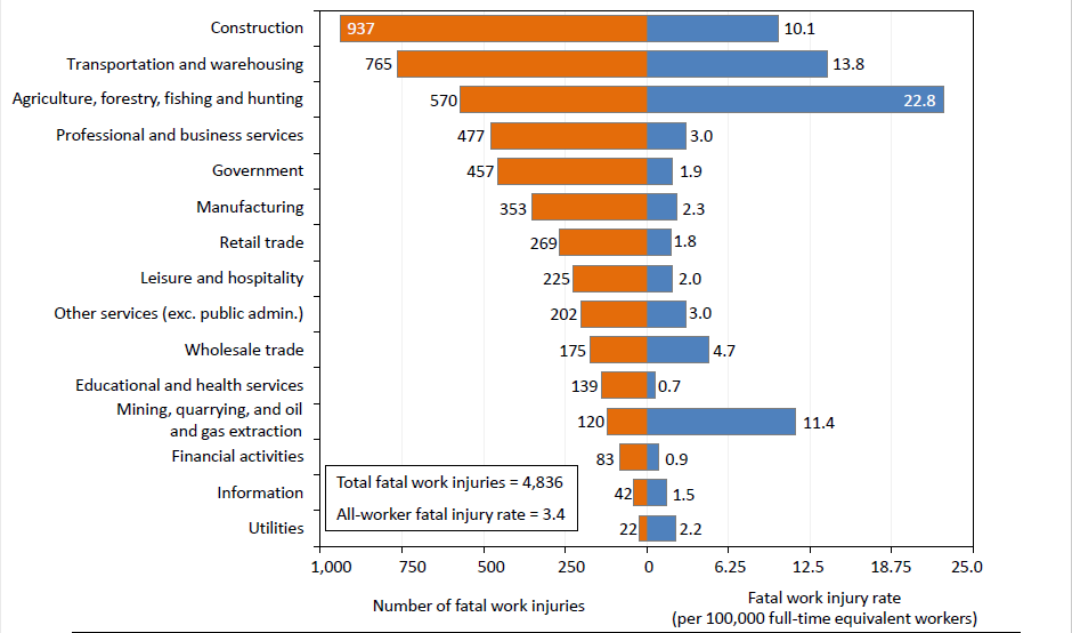


Figure 1. The rate of fatal injuries in the United States (Bureau of Labor Statistics, 2016, p. 14).

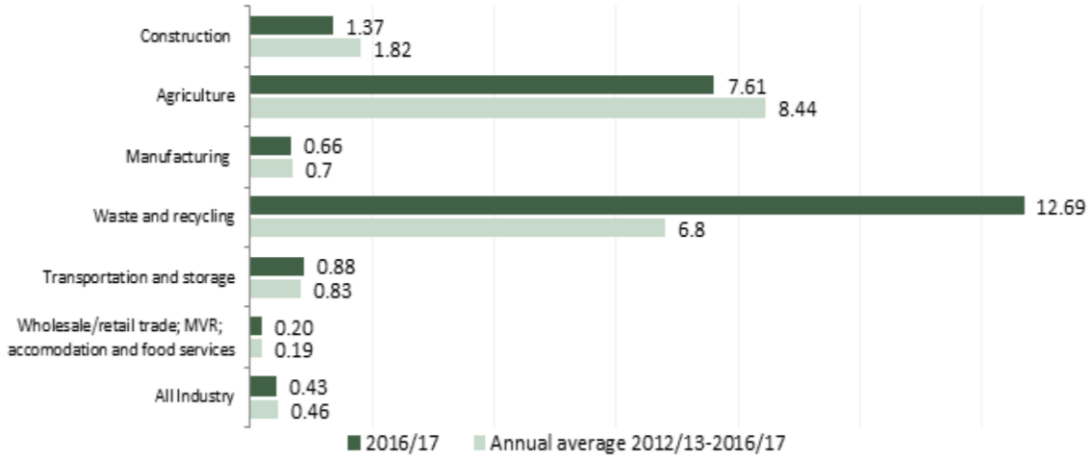


Figure 2. The rate of fatal injuries in Great Britain (Health and Safety Executive 2017, p. 5).

1.2 SIM studio

Sustainable product process through simulation (SIM) studio is a subsidiary of SIM platform at Lappeenranta University of Technology (LUT) as shown in Figure 3. The focus area of the SIM platform is to add customer value to the rapid innovation process. Thereby, using a sustainable approach to enhance simulator-driven design and manufacturing for real-time simulations. In order to address the problem SIM studio was introduced. Thus, translating customer specific demands, which are being overshadowed due to technicalities hovering the innovation stage of vehicle development. SIM studio is serving as a building block in innovation process and interconnects customer demands directly with technicalities of customization of a single or multiple vehicle models. The working flow of SIM studio, can be classified into two environments: modeling environment and simulation environment, with the latter being provided by Mevea Solver (simulation program). Since, this research is part of SIM Studio, so mainly the improvements concerning modeling approach in real-time simulations are addressed. The main working principle of modeling environment in SIM studio follows three basic steps:

1. Obtaining information from the user.
2. Integrating customer demands into model's preprogrammed simulation file.
3. Setting up the simulation environment for respective vehicle model.

Figure 3. SIM Studio

The first step in the previous version of modeling configurator as developed by Jaiswal (2017), was to take the desired values of different attributes selected by the customer and storing them in one excel file. The second step involved the integration of customer selections stored in excel file to preprogrammed simulation files. These files are in Extensible Markup Language (XML) format, as those are generated by Mevea Modeller (MeVEA Modeller [simulation program] 2017a, p.11). This step act as a translation medium between the customer and simulator. In the last step, the modified file carrying customer demands is executed by Mevea Solver, in order to provide user the experience of self-customized vehicle.

Since the simulation environment is based on Mevea Solver, therefore it was critical to understand its functionality. Mevea Solver is a simulation software developed by Mevea Oy, a recent university spinoff. Mevea Oy, offers two software programs in the field of real-time simulation, Mevea Modeller and Mevea Solver. The visualization of a model in Mevea is done by creating a 3D model of parts and converting them in .3ds format as it is recommended graphics file format (MeVEA Modeller [simulation program] 2017a, p.11). Initially the model simulation files is prepared in Mevea Modeller, which serves as a modeling tool and later for real-time simulation Mevea Solver is used. Once the vehicle simulation file is completed in Mevea Modeller three model files are generated, one .mvs file and other two in .xml format (MeVEA Modeller [simulation program] 2017a, p.7). The purpose of simulation file (.mvs file) is to combine the two XML files along with some additional features. One XML carries all the necessary information pertaining to the operability of a model, while the other carries all the information regarding environmental conditions. The simulation environment (Mevea Solver) readability is restricted to XML files as input, therefore, the urgency to modify XML file was of key importance. This lead to the modification of preprogrammed simulation file in XML format, in order to customize a vehicle model.

1.3 Research problem and research question

Earlier SIM studio was capable of simulating two vehicles that are; Tractor and Wheel loader, whereas, for academic purposes fork lifter model was also available. Whereas, the modeling environment was assisted by modeling configurator developed by Jaiswal (2017). This version of modeling configurator was capable of modifying the model, but still there

were some complications, concerning the customization of the vehicle by the customer. The available configurations for given attributes (engine, tire and gearbox) were taken from real data. Such as in case of engine performance curve, the torque maps were traced against real standard engine already available in the market. Similarly, limited options were available for the customization of tire. This approach restricted the user demands and also limited the user from experiencing desired configuration.

The process of cutomization is carried out in different steps. At each step different software is required. In the first step for user selection Microsoft Excel is used, which is an additional step and also lacks graphical representations. The addition of graphical user interface (GUI) enhances the usability of a simulation platform (He et al. 2008, p.460). The second step of executing modeling configurator was used to translate user demands from excel file into XML simulation file of model. In order to facilitate both customer and manufacturer in the innovation process, it therefore required to combine these two stages in one compact solution for a generalized approach. In this way, the observer can get an instant feedback resulting in back and forth modifications without switching to any other environment (Tian et al. 2008, pp. 469-470). Therefore, to make it more user-friendly, the presence of a GUI subtle enough to fulfill the requirements would facilitate the process. According to Weyrich & Steden (2013, p. 606), 65% of the total effort in simulation project is consumed to model the simulation. This effort creates a barrier which irrespective of cost factors, hinders the application of simulation models for companies. Therefore, the SIM studio needed to be developed more generic and the concept of reuse of models is a viable area which needs to be addressed as well. In light of the above stated problem the main research question formulated were as followed:

- How to offer a more user-friendly and interactive modeling environment?
- Why modeling configurator should be used as a separate tool to modify XML file?
- What limitations needs to be defined for customization of each attribute?
- What algorithm needs to be developed, to avoid lengthy code for each available configuration provided?
- How to create a unified algorithm for integrating user demands into preprogrammed simulation files?

1.4 Aim and objective

The main aim of this research was to make the modeling configurator more generic addressing all the research questions stated earlier. Conceptual model programming is a term used to develop an application via programming that is capable of translating the desired conceptual model (Embley, Liddle & Pastor 2011, p. 4). Thus, a more advanced graphical user interface was developed via programming that is capable of translating this conceptual model and combining the three stages prior to virtual simulation in one interface as shown in Figure 4.

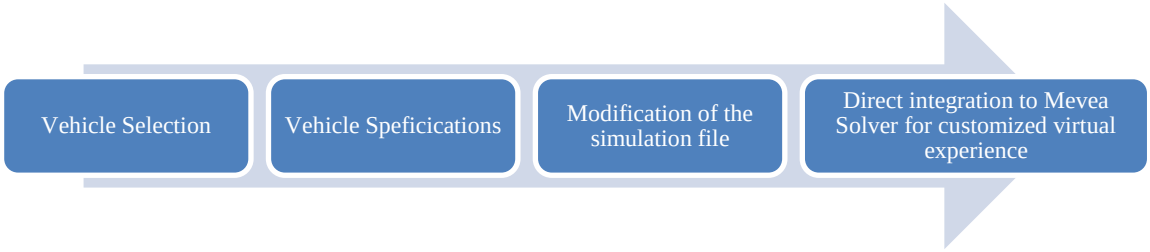


Figure 4. Process flow of generic modeling configurator.

For each attribute, generic mathematical models were to be defined under certain constraints/limitations as defined by vehicle simulation model. This gives the user a freedom to model each attribute from a range. These ranges were defined against each vehicle model to make the model more realistic. The specifications that were customizable by the user are as followed:

- Vehicle Specific rated power and speed of the engine (for engine modeling).
- Vehicle Specific number of gears forward/reverse (for gearbox modeling).
- Vehicle Specific size and quantity of tires (for tire modeling).

1.5 Research methods

The project follows a similar approach as used by Pidd (1992, p. 238), where, the simulation models of the selected vehicles are already preprogrammed. This removes user requirement of having technical background. In this aspect, a user with no programming background can also modify vehicle models with ease. Hence, the user will have the freedom to modify attributes (engine, tire and gearbox) via GUI.

Estimation of engine torque is necessary as it is needed in designing of powertrain, gearbox and thereby, also required for optimizing engine performance (Rakotomamonjy et al. 2008, p. 43). In case of engine modeling for real-time simulation, its dynamics in Mevea is defined as motor following a charateristics curve. Therefore, upon user selection modeling configurator needs to formulate a spline accordingly. In order to obtain a power and torque against speed map as desired by the customer for respective power rating as shown in Figure 5, a generic engine model was required. There are two techniques that can be used for mathematical modeling of generic engine model:

1. Using interpolation/regression analysis from manufacturer defined curves of different engines.
2. Using a generic equation to model engine performance curve.

In case of tire, they are defined as forces in Mevea, carrying a physical profile defined as spline, whereas, other properties are carried along in dummies. The customization of tire not only effects the dimensions, but also the inertial properties associated to its dummy. Therefore, the changes corresponding to all the possible standard sizes available in the market are addressed based on literature models available.

In Mevea, gearbox can be modeled as manual or automatic, which is a sub-component of power train. In terms of user customizability, the total number of forward and reverse gears can be modified. Upon modifications, the gearbox configuration for gear ratios is disturbed and are needed to be redefined. The designing of gearbox will be based on available methodologies in literature, and best suitable methodology will be adopted.

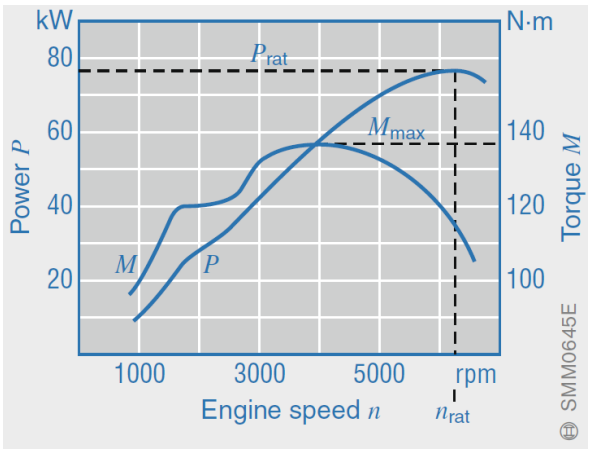


Figure 5. Torque and Power Curves vs Engine Speed (Reif 2015, p. 17).

1.6 Overview

In the following chapter, a thorough literature review is carried out in two sections that are generic modeling in the simulation and literature review for each attribute. On the basis of this review, the next chapter addresses the development of modeling configurator. This chapter is the core of this research and is categorized in three sections. The first section of this chapter addresses the generation of mathematical models for attributes to be generalized. The second section discusses tools and methodology used in the development of modeling configurator. Whereas, in third section independent case studies for each vehicle are discussed. Later, all the results are presented along with reasoning and validations of the assumptions made during the development. In the end, issues encountered during the simulation of models are discussed and a conclusion is made along with future recommendation.

2 BACKGROUND OF GENERIC MODELING IN SIMULATION

According to Santayana (1905, p. 92) “Those who cannot remember the past are condemned to repeat it”. Since, simulator have been widely used in several fields such as marketing, training, entertainment and research. Therefore, in light of such advancements, SIM studio was introduced for carrying out research activities in simulators. The foundation of modeling environment was laid down by Kaikko (2015) in his research for the development of new electric motor solution, shielding simulators as a marketing tool in the development. To support the innovative solution, a generic simulation model was also developed (Kaikko 2015, pp. 22-42). In continuation of this, Jaiswal (2017) further developed the idea used in earlier studies and tossed the idea of user modification in the preprogrammed simulation model via modeling configurator. This enabled the users to participate in the innovation process, where the user can select the custom configuration from a preconfigured list of options. As this research focused on development of a tool (modeling configurator) for simulator which is generic in nature. It required mathematical models for engine, tire and gearbox. For such purposes a systematic review was carried out for existing research and the selected keywords were used in a sequential order in different databases (see Appendix I).

2.1 Generic modeling in simulation

In the preliminary stages of design process, defining a configuration task is an important step towards efficient production. Mittal & Frayman (1989, pp. 1395-1401) investigated a generic model for solving configuration tasks. According to their research, any configuration task can be carried out in three steps: collecting all the influential component of the task, functional decomposition of the said components and defining a key component along with its relation to each function (Mittal et al. 1989, p. 1398). In term of generating a physical model, Ferryman et al. (1995, pp. 127-136) developed a generic tool enabling classification of vehicle type by defining 29 structural parameters during screening from a video footage. Whereas, Limin (2002, pp. 1055-1059) further developed the tool enhancing its capabilities in regenerating 3D shape of the corresponding vehicle.

While performing simulations, due to the complexity of models high computational power is required and making the iterations process for an analysis code problematic. To optimize

the conceptual design in virtual prototyping, Eldred et al (1996, pp. 1568-1582) developed a generic tool using object-oriented programming enabling the ease of modification without altering the analysis code. This idea of object-oriented approach and presence of user interface was followed by Neagu (1997, pp. 345-350) and mainly focused on the centralization of a conceptual model in generic prototyping.

With the familiarization of object-oriented approach in conceptual modeling, different organizations adopted to this methodology. There are different modules running inside an organization and unified approach breaks the harmony of information system in an organization. To keep this balance of information in the organization, Gargouri, Ducateau, & Boufares (1997, pp. 31–39) emphasized on the importance of an intermediate step between design and application environment.

As the approach developed, generic model were more and more implemented in the field of study. This gave rise to transformation of simulation models from experts only to user-oriented. Therefore, Caldwell & Fernandez (1998, pp. 197-225) in their research developed such a generic model. Thus, allowing the users to create the desired simulation by merging hierarchy theory with object-oriented programming. For efficient reuse of components, generic models were adopted, eliminating the need for reprogramming. Similarly, Lee & Rolland (2000, pp. 12-19) also focused on the generic modeling and devised the concept of a repository of reusable components as shown in Figure 6. This approach allowed the development of new conceptual models on the basis of already stored generic models.

The ever-increasing demands from the customers and market pressure has pushed the manufacturer to a point of producing and manufacturing a variety of products in lesser time than ever. In order to compete with market demands on producing a variety of models to meet customer needs, organizations have to develop a framework of a production system that is flexible and proficient enough to cope up with model variations. In light of such circumstances, Kirchner & März (2002, pp. 3627-3640) presented a generic platform with self-adaptive general structure, allowing the ease of reconfiguration of the system to adapt the ever increasing versatility of products. Later, Rajanna et al. (2012, pp. 549-555) also developed a generic model, allowing flexibility in the assembly line operations for typical final assembly operations in vehicle manufacturing.

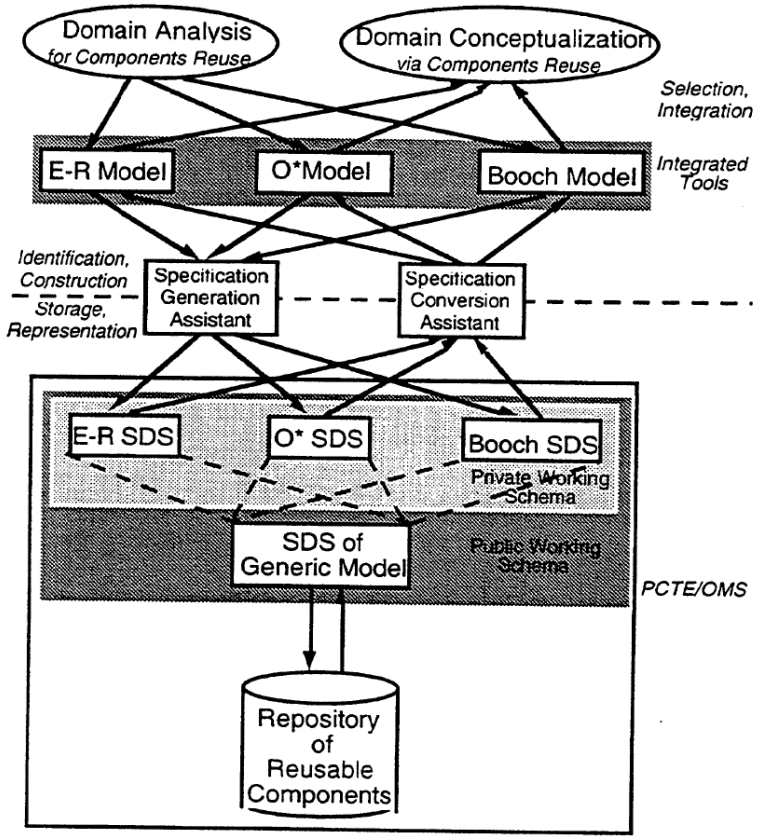


Figure 6. Block diagram for conceptual models development (Lee et al. 2000, p. 14).

From products perspective in automotive industry, generalizing a vehicle model in modeling and simulation can be complex. Therefore, Åsbogård (2004, pp. 469-474) examined a wide class and range of vehicles to evaluate dynamic models for generic modeling. The applied methodology was sustainable and efficient, as it emphasized on breaking a whole vehicle model into sub-models. Each model carry's respective information which can be simply used from the libraries and later assembled for evaluation purposes.

The implementation of object-oriented approach in modeling and simulation requires the model to be in structural hierarchical way. Gyimesi (2008, pp. 964-971) supported the use of XML in such scenarios, due to its structural representation capabilities. In the presence of a model in XML format, it's editing for simulation purposes needs some expertise level from the user. For the modification of such simulation models, a transformation module (graphical user interface or text editor) was presented as shown in Figure 7. The module receives the input data from the user. It then modified and regenerated the model, which was later delivered to the simulation engine for further processing.

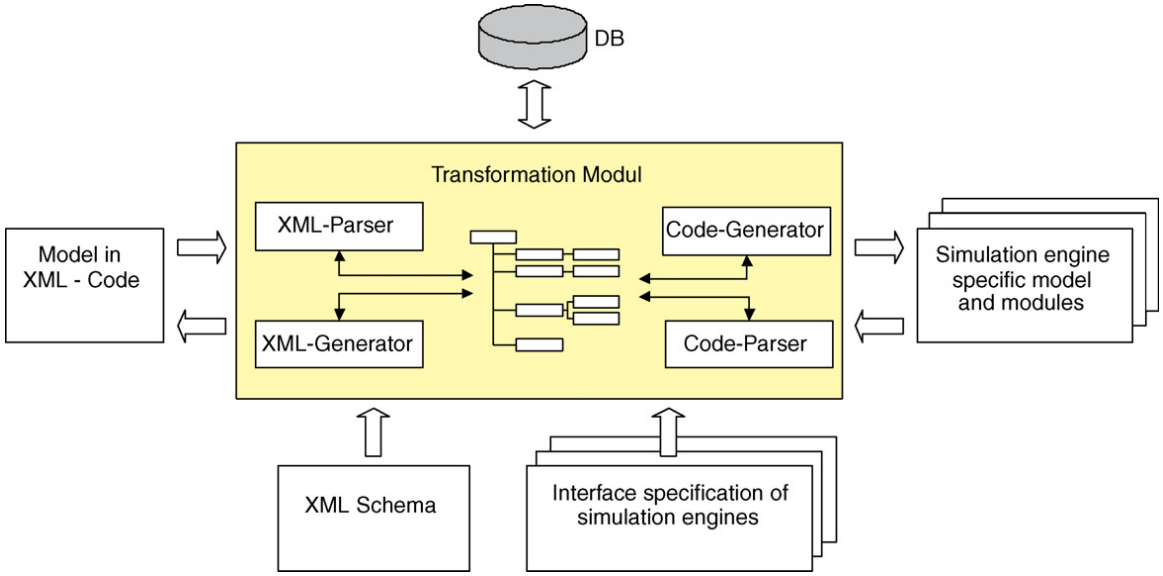


Figure 7. Breakdown of simulation process (Gyimesi 2008, p. 966).

With advancements in solving various aspects of engineering problems, MATLAB®/Simulink assisted as a tool in solving various aspects and thereby became a common practice. Therefore, Nair & Rajagopal (2010, pp. 753-757) developed a generic model for an electric vehicle utilizing the concepts of submodels and with the help of block diagram simulated a dynamic generic model for performance prediction. Wherein the drivetrain sub model for simplification purposes, followed lookup table for torque against speed maps. Furthermore, a more focused generic model for hybrid electric series powertrain was developed by Chindamo, Gadola, and Romano (2014, pp. 135–144), utilizing MATLAB®/Simulink as tool for modeling purposes. They also used a similar approach by dividing it into six subcomponents, where each block represented a mathematical model of sub-system.

2.2 Mathematical engine models

In this section, the literature review was carried out to discuss different techniques used for engine modelling. As one of the main objective was to provide user the freedom to model an engine. The literature review has been done to extract mathematical models that could be used to estimate engine performance curve. Thereby, the key point was to find engine models with least number of parameters (rated power, maximum torque, rated speed, engine efficiencies, air fuel ratio, mass flow rate, physical dimensions of an engine and more).

The performance of an engine is effected by several parameters. A slight changes in value of any parameter might result in different power rating at different rated speed. Different studies investigated the impact of these parameters on engine performance. Crossley & Cook (1991, pp. 921-925) investigated that the integration of vehicle control system requires a non-linear mathematical model. The study suggested the application of regression analysis complementing the fitting curve method extracted from dynamometer data. The developed model was comprised of Non-uniform rational B-Spline (NURB) equation for engine torque generation, as a function of air and EGR mass charge, air-fuel ratio, spark advance and engine speed. Later, Marcin et al. (2007, pp. 587-595) studied the generation of the torque directly from ongoing combusion inside cylinder and formulated an engine model. The model emperically assessed the velocity, temperature and pressure derived from the working mechanism of engine cylinders applying thermodynamics principal. Since, torque is produced due to the force produced by the engine, therefore, using pressure calculation this force can be estimated. However, Rakotomamonjy et al. (2008, pp. 43-55) defined torque model using a corelation between instantaneous load, distance covered, vehicle speed, vehicle status of operation, clutch pedal position, engine speed and gear index. Moreover, a generic function of engine torque was presented by Eriksson et al. (2010 pp. 53-71) in the form of net work done by the engine after subtracting pumping and friction losses. Farzin et al. (2012, pp. 116-124) designed a novel methodolgy including the emperical dynamic nonlinear model to calculate torque, which in turn could fix the fuel ratio in engine. A new formulation to categorize all the parameters into three subjective stages; quantifying the initial data, indentifying their functional dependencies and defining the respective relations was suggested by Kuznetsov et al (2016, pp. 213-228).

In order to complement the assumptions, limitations and predictions used in developed models, further studies were included. Chiara et al. (2011, pp. 261-277) used curve-fitting regression on experimental data, to calculate indicated mean effective pressure using a polynomial equation. Thereafter, Noor et al. (2016, pp. 1917-1930) applied multiple regression technique to obtain a mathematical model for the prediction of torque on the basis of experimental data. A similar approach was implemented by Chen (2013, pp. 2135-2138) and used fitting formula to define engine performance curve using polynomial equation and calculated coefficient from experimental data. A case study was solved by Guzzella &

Sciarretta (2007, pp. 295-302), implementing polynomial equation for power and torque against speed map by calculating equation coefficients experimentally.

The engine behavior and its mathmatical modeling can be independent of experimental data and trial runs. Ni & Henclewood (2008, pp.2695-2702) evaluated three simple mathematical models; polynomial model, parabolic model and bernoulli model, which can be used in vehcile dymanics applications. However, in situations where external charateristics and mapping charateristics are unknown, Jiang et al. (2012, pp. 316-321) designed a sofware based on emperical formula to estimate the engine performance curve. The performance curve can be generated if maximum torque of engine, engine speed corresponding to the maximum torque, torque corresponding to the maximum power and engine speed corresponding to the maximum power are known. Also, in Simscape, generic engine model can be modeled using a polynomial equation, whereas, in such case the polynomial coefficients are predefined. In driveline dynamics, Jazar (2008, pp. 165-208) also suggested the use of polynomial for mathematical modeling of an engine, where the polynomial coefficients can be calculated from power rating against rated speed.

The classifications of extracted models can be summarized as shown in Figure 8. In the models there are three types of engine models. Generic equation models are formulated independent of experimental data. These models provide high computational efficiency with low parameter dependency at significant accuracy. Therefore, these models are discussed and evaluated in next chapter.

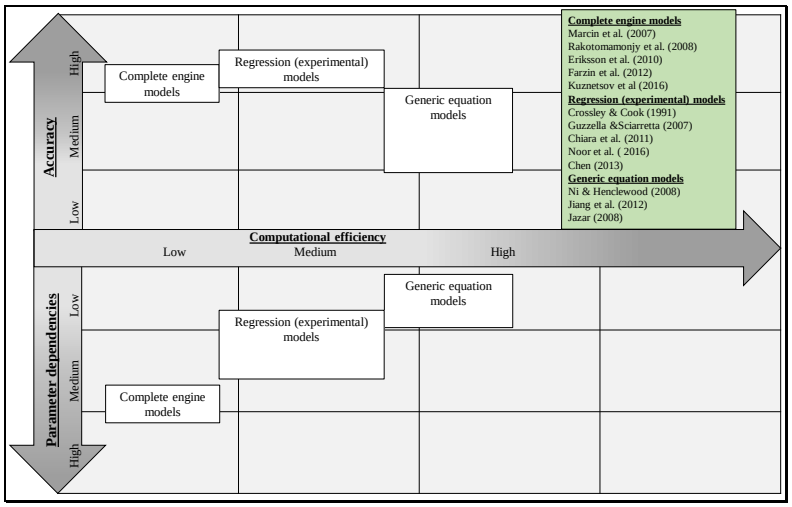


Figure 8. Classifications of engine models.

2.3 Generic tire models

There are various mathematical model developed for tire over the period of time, which can be classified on the basis of model complexity and design (see Figure 9). As, tire model plays a vital role in improving the results of vehicle model in multibody simulation. Therefore, Bakker et al. (1989) derived the magic formula tire model, which describes brake force, self-aligning torque and side force with sufficient accuracy. Due to the instabilities of this model, Sylvain, Julien & Franz (1997, pp.326-329) investigated the model at low velocities. The model was improved by applying Euler method on generic differential equation. Since, Magic formula tire model also produced complication in higher frequencies and short obstacles, Oosten and Pacejka (2000) developed a SWIFT tire model to compensate these discrepancies. Both magic formula tire model and SWIFT tire model had their own advantages, therefore, Besselink et al. (2005, pp. 245-252) combined the two model as MF-SWIFT tire model. The magic formula tire model is developed over time to overcome all the shortcomings, with the latest model called as PAC2002 which comprises of both empirical and physical modeling components (Kuiper and Oosten 2007, p.154).

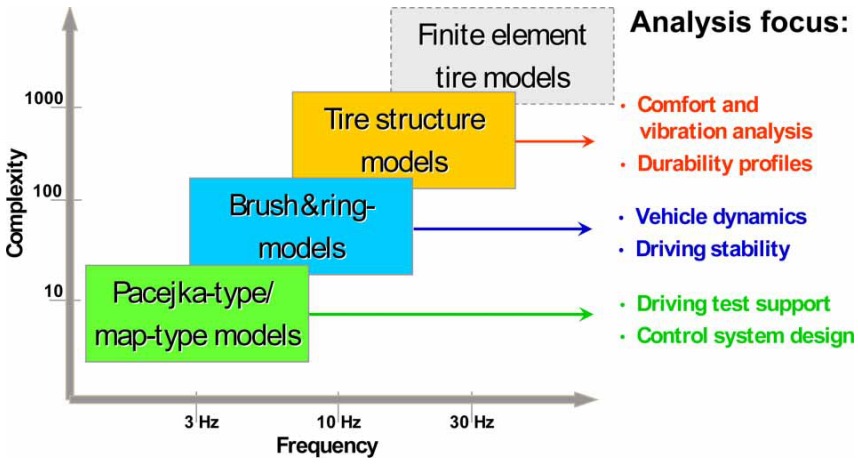


Figure 9. Classification of tire models (Ammon 2005, p. 42).

Likewise, Canudas et al. (1995, pp. 419-425) introduced LuGre model, a new dynamics model for friction including spring like charaterstics for stiction, hysteresis, stribeck effect and varying break-away force. In order to consider the suspension system while passing over obstacle, Negrus and Cocosila (1997, pp.322-325) developed another model with the assumption of vehicle being composed of rigid bodies. However, considering the representation of tire model solely in vehicle dynamics, Lee et al. (1999) proposed the use

of SEA tire model. Whereas, Liu et al. (2002) later introduced STI tire model, which was developed by using the concept of nominal slip in both lateral and longitudinal directions. While performing simulations of a system, the computational power required depends upon the complexity of the model. Therefore, Gallrein and Bäcker (2007, pp. 69-77) in their research discussed the use of CDTire, which compromises between the calculation time and scope of applicability. Also, Guo (1989) on the basis of experimental and theoretical analysis developed a pure lateral slip UniTire model, which was later evolved by taking large lateral slip, camber, turn-slip and longitudinal slip into consideration. Wherein, the scope and parameterization of UniTire model were discussed by Guo and Lu (2007, pp.79-99). Apart from these rational approaches, a different technique was used by Février and Fandard (2008, pp.26-31) in TaMeTirE model. The model used speed and tire temperature to accurately measure the lateral force for handling simulations. This model is more close to real time due to its capability to extrapolate force and moment. Since in farm machinery simulations due to high volume, tires experience nonlinearities because of high deflections, Paul (2018, pp. 3-14) addressed this issue by developing a multi spoke tire model called Hohenheim tire model. Furthermore, one other rigid ring tire model experienced limitation in dynamics responses and which was improved by Na et al. (2017, pp. 3530-3540) by introducing Bouc-Wen type force and called as Bouc-Wen tire model.

2.4 Generic gearbox

Gearbox is an integral part of vehicle transmission system and is used to adjust the required relation between torque and angular velocity depending upon the driving conditions. Over the years, vehicle transmission systems have evolved from early steam engines to 8-speed automatic transmission and even later evolved as more advanced and complex. When considering its designing on the basis of requirements, gearbox can have different configuration such as one stage or two stage etc. However, in practice while designing a gearbox, there are numerous conditions and parameters that need to be considered.

Nowadays, most commonly in vehicle transmission system three types of gear boxes are used: manual, automatic and continuously variable transmission. The designing starts from defining the largest gear ratio by considering the towing requirement and smallest gear ratio to increase either the fuel economy or maximum attainable speed by the vehicle (Guzzella & Sciarretta 2007, p. 52). Since, this research emphasis on the mathematical modeling of

gearbox, therefore, literature review was carried out accordingly. It was found that the preliminary step in designing, is to find the gear ratio for largest and smallest gear. Once they are calculated the intermediate gear ratios can be mathematically modeled using two techniques (Jazar 2008, pp. 187-190 and Harald N. et al. 2011, pp. 109-114) details of which are discussed in later chapter:

1. Geometrical gear step
2. Progressive gear step

2.5 Literature summary

The first section addressed the needs of generic simulation models. In light of the above review, some key finding were discovered. It was that the traditional approaches lack the presence of a medium that is generic for simulation purposes, eliminates the need for user expertise to modify the simulation model and user-friendly as well. This points towards the objective of this research to develop a user-friendly platform. The platform should be based on graphical user interface, developed by object-oriented programming which is capable of modifying preprogrammed simulation files. Whereas, the later sections point toward different mathematical models which can be used for modeling of each attribute.

3 DEVELOPMENT OF MODELING CONFIGURATOR

In the previous part the detailed studies of Åsbogård (2004, pp. 469-474), Nair et al. (2010, pp. 753-757) and Chindamo et al. (2014, pp. 135–144) for generation of generic simulation models of vehicle, showed that while modelling the vehicle as a whole, the convenient way is to break it down into submodels. Following a similar approach, development of modeling configurator was progressed according to the study of Mittal et al. (1989, pp. 1395-1401) for handling configuration tasks. The first task was identifying the available components. Present study integrated the components as three types of vehicles; tractor, wheel loader and fork lifter. Furthermore, the functional decomposition and identification of key components of each machine were carried out accordingly. As shown in Figure 10 the functional decomposition and key components of each machine contain all the available modifiable attributes for each vehicle model.

The key components as shown below in case of each vehicle are engine, gearbox, and tire. In case of generation of a transformation module as implemented by Gyimesi (2008, pp. 964-971), the submodels need to be generic. In generalizing these three key components the presence of gearbox seems to be problematic, as the preprogrammed submodel is automatic for the case of fork lifter. Therefore, for simplification purposes, gearbox modeling for fork lifter is forwarded by default to the simulation engine therein as a simulator in SIM studio and the need to alter the preprogrammed code is omitted. Similarly, following the approach of Lee et al. (2000, pp. 12-19) to make the module more efficient and useful, the theory of repository of reusable components is applied using a generic object-oriented approach via programming language. This approach enabled the possibility of adding more components and functions in the module for future activities.

The three key components under consideration for development of generic simulation module called as attributes are evaluated further for the generation of an algorithm for use in sub-models of respective functions. The applied framework was adapted from the study of Pidd (1992, p. 238) and Gyimesi (2008, pp. 964-971), so the user is allowed to modify attributes from realistic ranges. The module wherein called as modeling configurator, acts as a translation medium eliminating prerequisite for knowledge of programming from the

user by modifying the simulation files of preprogrammed models. In order to modify these attributes, it is critical to understand their direct or indirect effect on the other governing attributes. Figure 11 illustrates the flow of data between attributes in an XML file generated by Mevea, considering the direct impact of user modification in the process chain. It was found, that the changes in values of parameters during the modeling of an engine and gearbox does not influence other attributes.

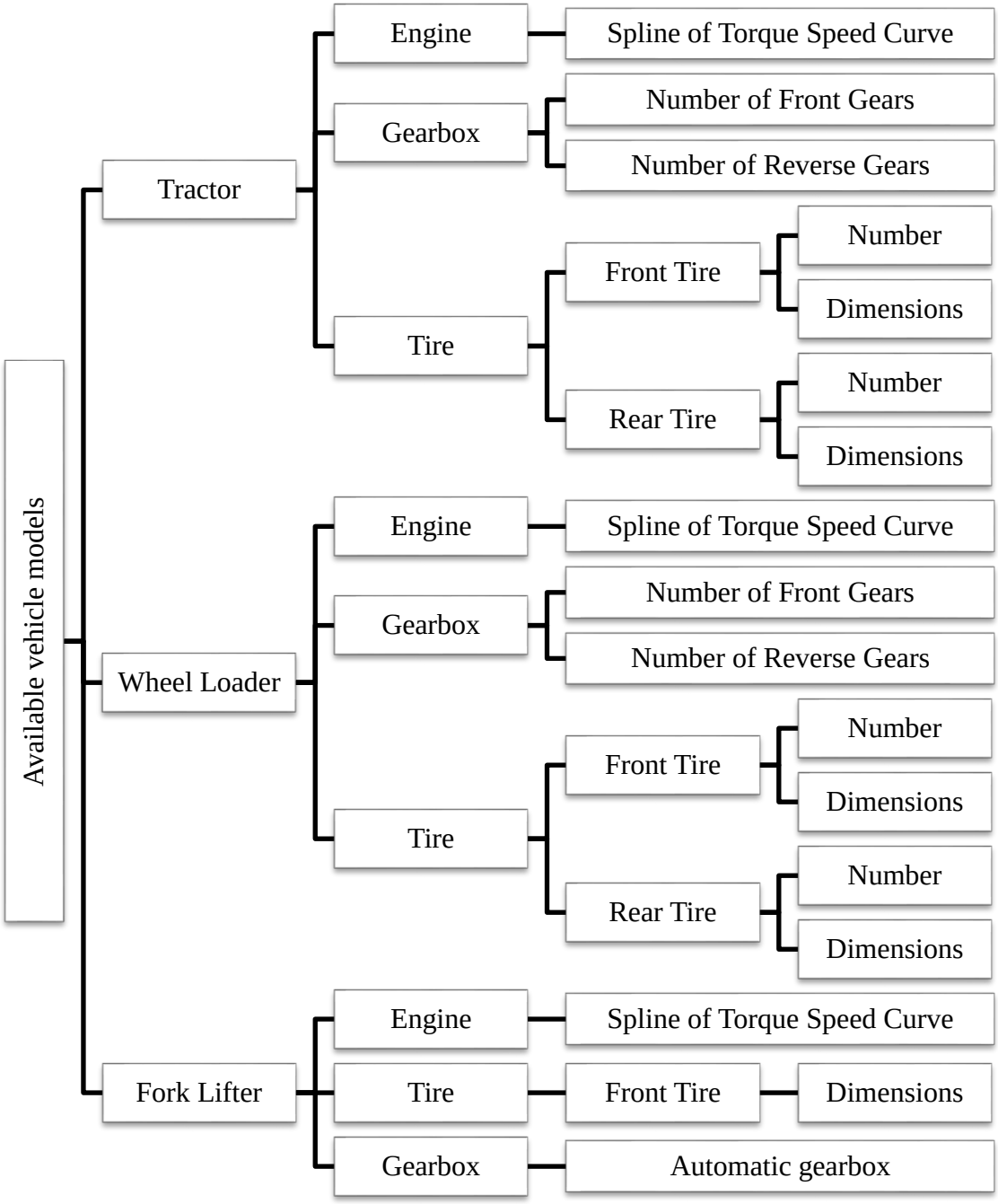


Figure 10. Functional decomposition of vehicle models.

While in case of tire modeling, there are multiple parameters emerging into dummy of a tire in Mevea, which are interlinked to each other. Once the parameters of the tire are modified, correspondingly the values of inertial and graphics properties are affected. Also as tire acts as a direct input to the brake, therefore, the addition of multiple tires suggest changes in the definition of brake inputs. Similarly, new inputs are also to be defined within brake while considering additional tires.

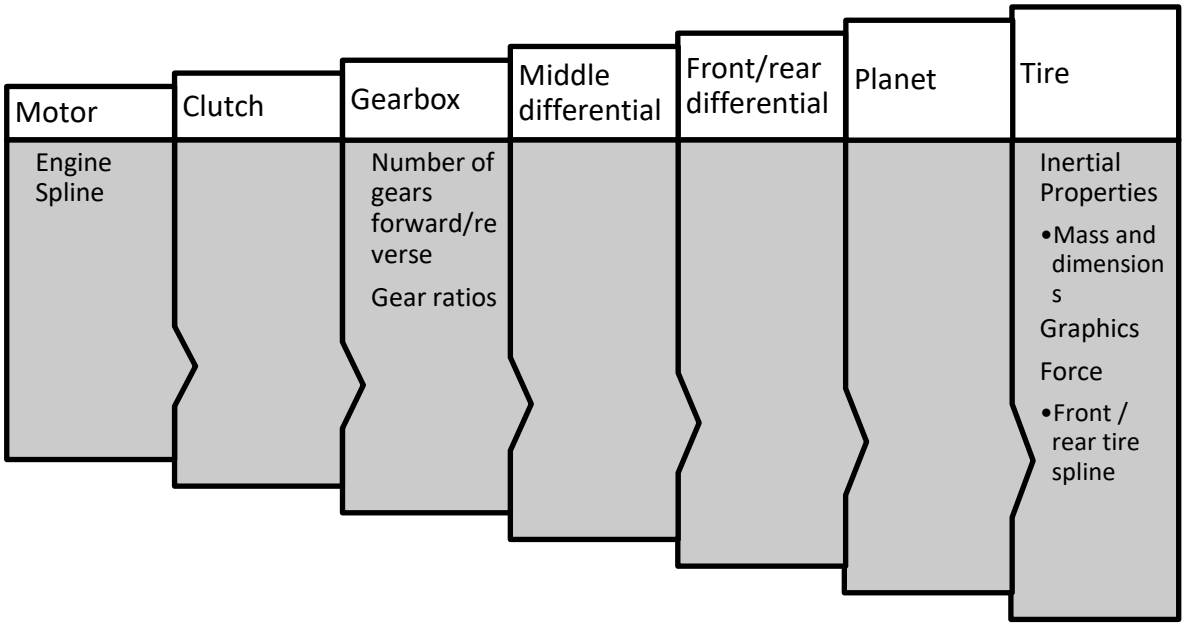


Figure 11. Flow of data within attributes in an XML file.

3.1 Development of engine model

The simulation environment used in this project is based on Mevea Solver, which is a product of Mevea Oy. The software is user-friendly and readily available at LUT for research purposes, making it the optimum choice. In order to develop an engine model, it was critical to understand the working of software. The software handles the power transmission of a working machine in a similar way as shown in Figure 12. The engine of a vehicle is dealt as motor in the power transmission system of Mevea Solver. While, in the simulation environment (Mevea Solver) torque derivative $\dot{T}$, is governed by equation (1) (MeVEA Modeller [simulation program] 2017a, p. 54):

$$\dot{T} = \frac{k_w \cdot (\omega_{ref} - \omega_o) - T_o}{\tau} \quad (1)$$

where k_w is gain of velocity difference, τ time constant, ω_{ref} motor rotational reference speed, ω_o angular velocity and T_o motor torque magnitude. Thereby, an engine model is replaced by a motor and defined in two steps. In the first step, it is required to define a torque against speed map in the shape of a spline which guides the simulation software during the simulation for the behavior of the engine at corresponding angular speeds. The next steps deal with the physical parameters of the motor which describe the correlation of motor with other components.

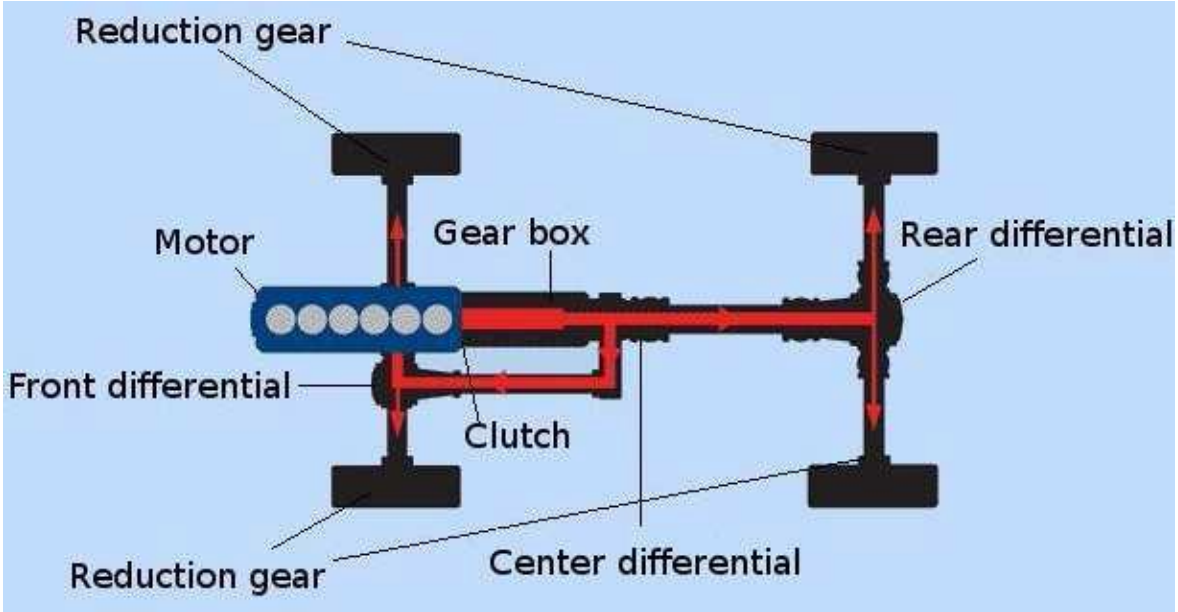


Figure 12. Power transmission system of a working machine (MeVEA Modeller [simulation program] 2017b, p. 32).

As the objective defines the user friendliness of modeling configurator, prime focus was given to reduce the number of parameters required for engine modeling without lossing significant accuracy. Therefore, when the parameters are taken as input from the user, it demanded the formulation of a generic engine model that is capable of defining the behavior of the desired engine. Also, a similar kind of behavior is required in the definition of motor in Mevea, where performance of motor against speed is defined in the form of spline. As explained earlier in the first chapter, two different techniques can be used to produce a similar torque speed map as shown previously in Figure 5.

Usually, this kind of torque-speed map is provided by the manufacturer along with the engine specifications. This maps is generated by testing the engine on testbed while measuring the relevant data by using a dynamometer. The principle of dynamometer for measuring engine data is illustrated in Figure 13. Using the principle of the dynamometer, torque *T* can be calculated by equation (2), as a product of force *F* and the length of moment arm b. The power *P* as produced by the engine and measured by the dynamometer is calculated by using equation (3), against measured engine speed ω_e.

$$T = F \cdot b \tag{2}$$

$$P = (^{2 \cdot \pi}/_{60}) \cdot \omega_e \cdot T \tag{3}$$

The above-explained methodology is the core concept of the first proposed technique stated in first chapter. If the dynamics of an engine is studied in depth it can be found that the power generated by the combustion process is in direct relation to the pressure acting on the piston. This pressure is exerted by the forces produced during the chemical reaction. It is hereby required to obtain significant torque against speed map as provided by the manufacturers in correspondence to specific configurations to be generalized. However, upon collection of data from the various manufacturer, it was found that the case-specific data was not enough to carry out the regression analysis. Therefore, this technique was not applied in this case for developing an engine model.

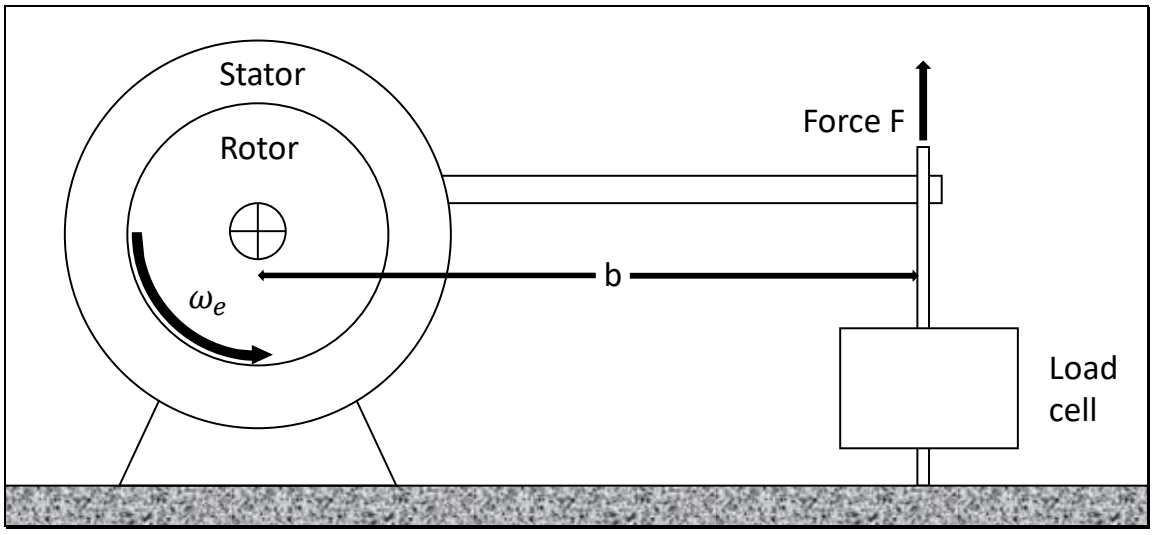


Figure 13. The principle of a dynamometer (according to Heywood 1988, p. 46).

In practice, the power produced by the engine is dependent on various parameters. These parameters affect the performance of an engine and are therefore called as performance parameters. The power produced by the engine is directly proportional to the product of fuels conversion efficiency η_f, volumetric efficiency η_v, angular speed *N*, displaced volume V_d, heating value of fuel Q_{HV}, density of air injected $\rho_{a,i}$, and fuel to air ratio (F/A) being injected into the cylinder per cycle. So the instantaneous power *P* can be calculated by using equation (4) (Heywood 1988, p. 57):

$$P = \frac{\eta_f \cdot \eta_v \cdot \omega_e \cdot V_d \cdot Q_{HV} \cdot \rho_{a,i} \cdot (F/A)}{2} \tag{4}$$

As the above-stated formulation is based on practical measurements which cannot be covered in the scope of this project. Therefore, an optimized model was desired which was the core concept of the second proposed technique. In traditional approaches for generic modeling techniques as used by Nair et al. (2010, pp. 753-757) and Chindamo et al. (2014, pp. 135–144), it can be found that lookup tables were used for the provision of torque vs speed maps. The use of such lookup tables for meeting user requirement would mean the inclusion of all lookup tables representing any configuration as selected by the user. In doing so, the objective of developing a generic model is lost and would result in increased length of coding. Whereas, from the study of Chiara et al. (2011, pp. 261-277, Noor et al. (2016, pp. 1917-1930), Chen (2013, pp. 2135-2138) and Guzzella & Sciarretta (2007, pp. 295-302), it is evident that polynomial can used for such purposes but the coefficients in the model are calculated experimentally.

On the basis of literature review, there are three generic engine models which use equations to obtain engine performance curves. First generic engine model found is in the shape of polynomial equation. This approach can be seen in Simscape which is a subsidiary of Simulink wherein physical systems are modeled. The available model follows the principle of engine power demand as required in this research. According to Simscape, the power demand function *p(w)* can be defined as third-order polynomial equation, which is a function of ratio of engine speed as shown in equation (5) (Generic engine, 2017):

$$p(w) = p_1 \cdot w + p_2 \cdot w^2 - p_3 \cdot w^3 \tag{5}$$

The coefficients of this equation p_1, p_2 and p_3 are bound by the relationships as shown in equation (6) and (7). The variable *w* in equation (5) is defined as ratio of engine speed ω_e to engine speed at rated power ω_p as shown in equation (8). Whereas, the performance curve can be estimated by multiplying power demand function with rated engine power P_m as in equation (9). The polynomial coefficients in equation (5) are pre-calculated and the corresponding values can be used for the defined case as represented in Table 1.

$$p_1 + p_2 - p_3 = 1 \tag{6}$$

$$p_1 + 2 \cdot p_2 - 3 \cdot p_3 = 0 \tag{7}$$

$$w = {}^{\omega_e}/_{\omega_p} \tag{8}$$

$$P = P_m \cdot p(w) \tag{9}$$

Table 1. Values for Polynomial of power function (Generic engine, 2017).

Power demand coefficient	**Engine Type**	
	Spark-Ignition	Diesel
p_1	1	0.6526
p_2	1	1.6948
p_3	1	1.3474

This model fulfills the criteria of this research but the values of the coefficients as described in table 1 are case independent. A similar model is also used in driveline dynamics (Jazar 2008, pp. 165-208), stated as power performance function. The engine power is estimated against engine speed by using equation (10) (Jazar 2008, p. 165). In this case, the coefficient of the third-order polynomial are not predefined and calculated for case-specific scenarios by utilizing equations (11), (12) and (13) (Jazar 2008, p. 166). Subsequently, the engine torque can also be calculated by applying equation (14) (Jazar 2008, p. 166). However, the values of coefficients *d*, *e*, and *f* are engine specific and their values are shown in Table 2.

$$P = \sum_{i=1}^{3} P_i \cdot \omega_e^i = P_1 \cdot \omega_e^1 + P_2 \cdot \omega_e^2 + P_3 \cdot \omega_e^3 \tag{10}$$

$$P_1 = d \cdot {}^{P_m}/_{\omega_p} \tag{11}$$

$$P_2 = e \cdot {P_m}/{\omega_p^2} \tag{12}$$

$$P_3 = f \cdot {P_m}/{\omega_p^3} \tag{13}$$

$$T = \frac{P}{\omega_e} = P_1 + P_2 \cdot \omega_e + P_3 \cdot \omega_e^2 \tag{14}$$

Table 2. Values of Parameters for Generic Engine (Jazar 2008, p. 166).

Coefficient for case-specific scenarios	**Engine Type**		
	Spark-Ignition	Diesel (indirect injection)	Diesel (direct injection)
d	1	0.6	0.87
e	1	1.4	1.13
f	-1	-1	-1

Engine model as defined by equation (10-14), can be used for plotting torque speed maps. The values for P_{m} and ω_p are taken as input from the user. The range of these inputs is case specific, as different vehicles have different operating conditions. Using the above formulations as given in equation (10-14) the engine behavior can be estimated and hence it is possible to generate power and torques against speed map. In order to evaluate the cases of Table 2 to be used for generic modelling of an engine all the cases were plotted against a reference value of P_m =160 kW and ω_p = 1900 rpm as shown in Figure 14. The result shows, that for the same rated power (P_m) and rated speed (ω_p), three different peak values are generated. For case 2 both the peak values diverge from the defined rated values. Whereas, the case 3 demonstrates that, the peak value for power is the same as rated power but the corresponding value for speed differs. But in case 1 it was observed that the peak values corresponds to reference values as provided. Therefore, coefficients of case 1 are adopted for this engine model, referred as model 1 thereafter.

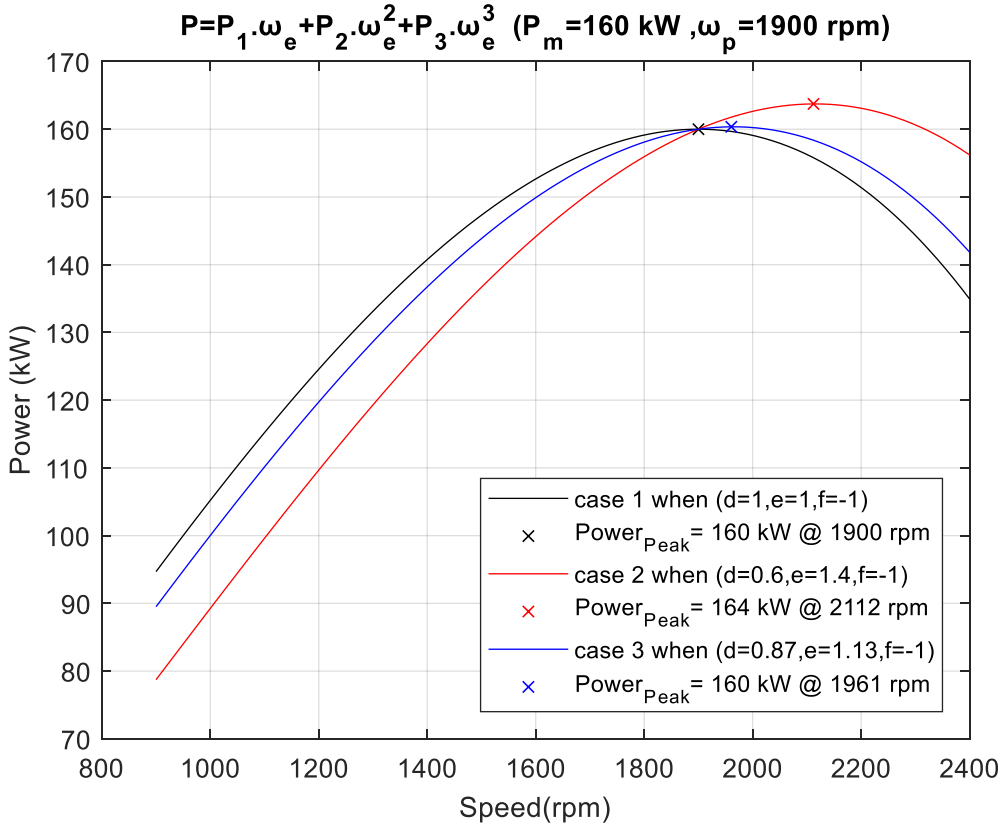


Figure 14. Coefficients comparison for polynomial engine model.

The other two engine models as found in literature are empirical formula method (model 2) and parabolic model (model 3). These two types of models were investigated by Jiang et al. (2012, pp. 316-321) and Ni & Henclewood (2008, pp.2695-2702) using equation (15) and equation (16) respectively. In order to test the accuracy of these three model, engine charateristics were plotted against real engine spline on the left side and compared for accuracy on the right side as shown in Figure 15.

$$T = T_m - \frac{T_m - T_p}{(\omega_p - \omega_t)} \cdot (\omega_t - \omega_e)^2 \tag{15}$$

$$P = \frac{P_m}{2 \cdot {\omega_p}^2} \cdot (3 \cdot \omega_p - \omega_t)^2 \cdot \omega_e - \frac{P_m}{2 \cdot {\omega_p}^2 \cdot (\omega_p - \omega_t)} \cdot (\omega_e - \omega_t)^2 \cdot \omega_e \tag{16}$$

Where, T_m represents maximum torque, T_p is torque at maximum power and ω_t is corresponding speed at maximum torque.

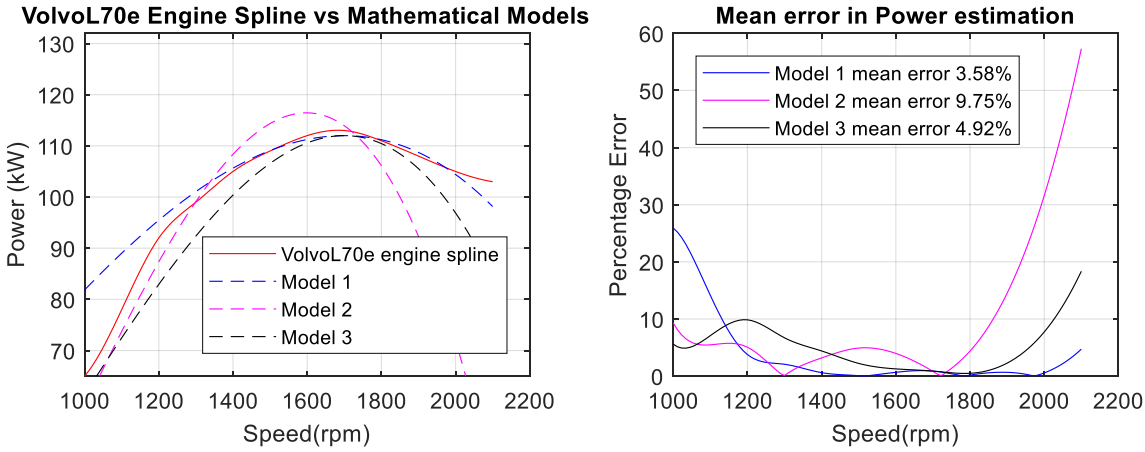


Figure 15. Comparison of generic engine models.

The available three models are evaluated for use in real-time simulation, against different specifications (least number of parameters, ease to access for parameter, minimum accuracy and Chi test score) using radar chart as shown in Figure 16. The evaluation criteria was set from a scale of 0-5, with 5 being the most feasible. Since, model 1 requires the least number of parameters which are relatively easier to access it has the highest score. Also, to evaluate minimum accuracy of each model and their respective Chi-Square statistics (Verma 2013, pp. 69-101), the models were compared against data of 8 real engines as generated by the manufacturer (see Appendix V). It was observed that the model 1 as proposed by Jazar (2008, pp. 165-208) is more feasible against all specifications in comparison to the other models. Therefore, model 1 is used for the purpose of engine modeling in the scope of this research.

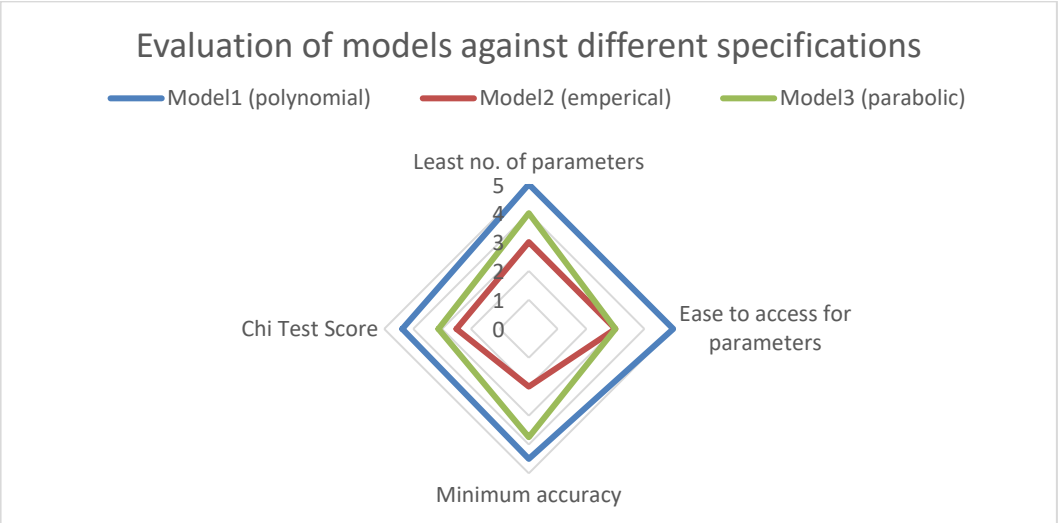


Figure 16. Evaluation of generic engine models.

3.2 Development of tire model

In the definition of tire model for use in Mevea Modeller, they are modeled as forces. Whereas, from modeling perspective it was found in the literature, that there are two tire models most commonly used in tire modeling. Also, both of these models that is Pacejka Magic Formula and LuGre tire model are available in Mevea Modeller (MeVEA Modeller [simulation program] 2017b p. 36). The role of traction control in automotive industry for ground vehicles is quite critical, as friction force helps in transmitting angular acceleration to forward acceleration of the vehicle. Hence, dynamic friction model helps in the interpretation of tire/road interaction. On the basis of LuGre model, Canudas & Tsiotras (1999, pp. 3746-3751) derived a new dynamic tire friction model (dependent on speed and surface) and validated its accuracy for use in vehicle dynamics. Therefore, LuGre model is used in the scope of this research as well. The LuGre friction model is based on the formulation that due to irregular surfaces, rigid bodies make contact at several points and are assumed to be intact through elastic bristles as shown in Figure 17.

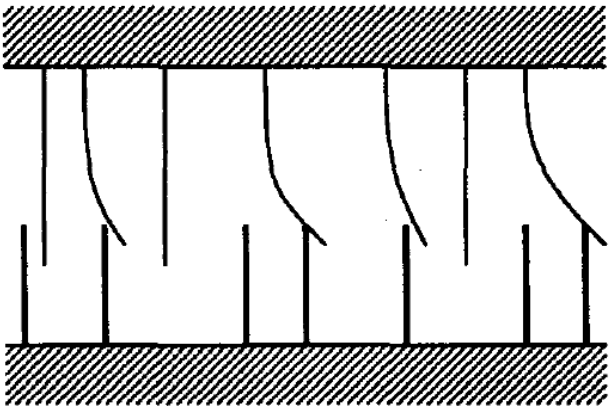


Figure 17. Surface representation as bristles with lower as rigid (C. Canudas de Wit et al. 1995, p. 420).

In the presence of large tangential force, the surfaces slip due to deflections in the bristles. These deflections denoted as *z* in equation (17) are random and therefore, average value is modeled in relation to relative velocity *v* between the surfaces (C. Canudas de Wit et al. 1995, p. 420):

$$\frac{dz}{dt} = v - \frac{|v|}{g(v)} \cdot z \tag{17}$$

As a result of tangential force, the bristles are deflected and friction force is originated. This friction force in equation (18) is a function stiffness σ_0, damping coefficient of the bristle σ_1 and accounting viscous friction σ_2 (C. Canudas de Wit et al. 1995, p. 420):

$$F = \sigma_0 + \sigma_1 \cdot \frac{dz}{dt} + \sigma_2 \cdot v \tag{18}$$

The velocity function as given in equation (19) is dependent on several factors such as lubrication, material properties and temperature. This function is based on Coulomb friction level F_c, level of stiction force F_s and Stribeck velocity v_s modeled as (C. Canudas de Wit et al. 1995, p. 420):

$$\sigma_0 g(v) = F_c + (F_s - F_c) \cdot e^{-(v/v_s)^2} \tag{19}$$

In Mevea for physical dimensioning the profile of tire is also defined as a spline. It eliminates the need for defining tire radius and tire width separately. In total four points in each X and Y coordinate can be defined for an optimized profile of tire for reduced computational power. Since, two point lead to generation of a flat profile and three points may generate a tapered profile, but the choice of defining points solely depends on user preferences. One such example of profile generation of the tire from a spline in Mevea can be seen in Figure 18.

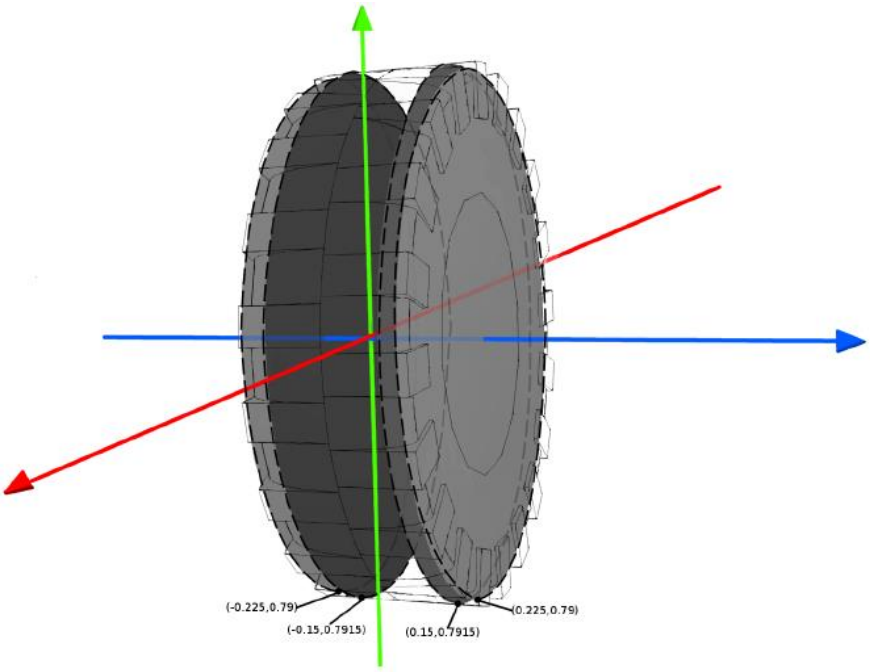


Figure 18. Tire Profile from spline (MeVEA Modeller [simulation program] 2017b, p. 37).

There are two types of representation for a tire; physical and visual representation. Physical properties such as dimensions, inertia, and mass communicate with the software in demonstrating the dynamic behavior of each tire during the simulation. The visual representation in form of graphics provides a user with visual feedback. The inertial properties of a tire and graphics linked to it are defined in dummy. In the description of physical representation, four definitions are required by the modeling configurator to evaluate the inertial properties of tire that are rim diameter, outer diameter, height or width of the tire and mass of the tire. In Mevea Modeller, tire for simplification purposes is assumed as a hollow cylinder. Therefore, by applying the principle of hollow cylinder, moment of inertia I_{xx} and I_{yy} can be calculated using equation (20) and (21) and due to symmetry I_{zz} is the same as I_{yy} (Marghitu & Dupac 2012, p. 140):

$$I_{xx} = \frac{1}{2} \cdot m \cdot (d_{out}^2 + d_{in}^2) \tag{20}$$

$$I_{yy} = I_{zz} = \frac{1}{12} \cdot m \cdot (3 \cdot d_{out}^2 + 3 \cdot d_{in}^2 + h^2) \tag{21}$$

Where m represents mass of tire, d_{in} internal diameter, d_{out} as outer diameter and h as thickness/width of the tire. During the literature study for tires, it was found that the weight of tire is linearly proportional to the section height SH, section width SW, and rim diameter RD. Also, one similar kind of formulation has been found in TyrFil (2017, p.2), where weight approximation can be made using equation (22). This equation estimates the mass of tire with an error approximation of ±15%, which is sufficient enough for demonstration purposes only. All the input in this equation are provided in inches resulting in the value of mass in lbs, whereas, after dividing by the conversion factor of 2.2 the value is converted into kg.

$$m = \frac{2.47 \cdot SH \cdot SW \cdot (SH + RD) \cdot 0.036}{2.2} \tag{22}$$

Graphics are quite important for visualization in any simulation program. Similarly, the graphics of each component of vehicle in Mevea are imported preferably in the format of '.3ds' file. To use separate graphics model for each value of tire model would require creating an equal number of models as well. Therefore, for simplification purposes, only one

model was taken into consideration for reduced computational power. Since, user can visually experiences any alteration in size through graphics of tire. The visualization of tire was controlled by modeling configurator utilizing the scaling property of imported graphics. As per definition of tire for interaction with the simulation environment, it is provided as spline. The tire spline demands values in two coordinates, x-values corresponding to the width of the tire and y-values corresponding to the outer diameter of tire. These dimensions of tire are directly taken from the user and formulated in tire model equations.

3.3 Development of gearbox model

In any vehicle, the gearbox is designed as per the requirement of the operating conditions. The transmission ratio of the gearbox is a dependent entity, thus, intermediate gear ratios are calculated on the basis of gear ratios for first and final gear. The gearbox is designed in such a way that in the first gear the engine can deliver maximum torque, whereas for the final gear maximum speed is desired. Usually, the value of gear ratio for final gear is designed in such a way that it is close to 1 and acts as a direct gear. In case the main requirement is vehicle maximum speed, the value of engine speed is taken as maximum economical speed and then final gear ratio n_i can be calculated by solving the relationship given by equation (23) (Jazar 2008, p. 191):

$$v_e = \frac{Re}{n_d \cdot n_i} \cdot \omega_e \tag{23}$$

where, v_e is vehicle speed, Re is radius of wheel, n_d is transmission ratio and n_i is i^{th} gear ratio. In general practice, the values for first and final gear are available beforehand. To proceed further with the designing of gearbox, the values for intermediate gears can be calculated using two methods (Harald N. et al. 2011, pp. 109-114):

1. Geometric gear steps
2. Progressive gear steps

In the first method, a constant gear ratio is defined called as step jump which acts as a preliminary step in evaluating values for intermediate gears. In the second method, the speed span for each interval of gears is kept constant such that the difference of maximum and

minimum velocity is the same for each consecutive gears. This way the gear step keeps decreasing with higher gears. In other words, spending more time in lower gears which consume more fuel, than higher gears comparatively. Therefore, in this research the first methodology is adopted and the gear step C_g is calculated by using equation (25) which is obtained by rearranging equation (24) for generic formulation (Harald N. et al. 2011, p. 111):

$$n_i = n_l \cdot C_g^{(l-i)} \tag{24}$$

$$C_g = \sqrt[(l-i)]{n_i/n_l} \tag{25}$$

Where, n_i and n_l are gear ratios with *i* and *l* as respective gear index number starting from *i*=1 to *l*.

3.4 Tool selection

When developing a generic modeling configurator two special kinds of tools were required. As the preprogrammed simulation models were in the format of XML files. The first requirement was the presence of a tool capable of editing XML files. Commercially, there are several available resources which can be used for this requirement. The second tool required was the presence of graphical user interface. This tool was developed in a similar as used by Pidd (1992, p. 238) in his studies for data-driven simulators. A detailed evaluation of all the available resources to formulate the required model for further development of modeling configurator is carried out in the next section. XML is a class of programming language which is classified under special purpose languages (Wilhelm & Seidl 2010, pp. 1-2). The use of XML documents allows information to be stored in the structured hierarchical way. The structure of XML document can be decomposed into 6 classifications: elements, attributes, text, entities, comments, and CDATA (Character Data) (Jacobs 2006, pp. 20-22). This structure is followed by Mevea Modeller in the generation of model files for simulation. For editing of XML documents, there are numerous editors available, which require basic knowledge of XML as a prerequisite. The allowance of such editors were not viable for use in this project, as for user integration, the program needed to be simple and easily operable. This task was dealt by using object oriented approach, since it allowed reusability, ease of handling particular tasks and ability to model particulars of real world in complete harmony among other benefits. This approach was extracted from the literature review done earlier.

There are several object-oriented programming languages which can be used to implement such approach through programming. Programming languages compatible with fulfilling the tool requirement are C++, Common Lisp, Dart, Java, Object Pascal, Objective C, Perl, PHP, Python, Ruby, Scala, Smalltalk, and Swift. Among all these, three are most widely used, C++, Java, and Python, and therefore can be used for developing a programming code for this purpose. The basic requirement for this programming code, is editing of the pre-programmed simulation files against each attribute as defined by the user.

The other requirement is the development of an interactive GUI, which can coordinate with the user for the selection of desired configuration and can subsequently edit the preprogrammed simulation files. Since, most of the users have little or no idea of programming languages, therefore, without GUI they cannot alter or modify the simulation files as desired. The presence of such tool plays a vital role as it serves as an interpreter between the user and the programming code. In order to build such interface by utilizing the above-stated programming languages, each language has different builder apps as well where one can write a code to create such interface. Whereas, keeping the industrial point of view, the presence of such builder apps in each industry is highly unlikely.

The development of a GUI platform on the basis of these programming languages alone make it complicated and demands a high level of expertise along with certain limitations. Thereby, to make it more efficient and user-friendly a blend was required. This is achieved by blending the programming language with a more user-friendly platform. Thus, allowing the modification procedure of simulation files demanding little programming knowledge. Hence, it can be summarized that the following requirements are to look forward in selection of tool for the development of modeling configurator, where:

1. It allows creation of graphical user interface for user interaction.
2. Configurations task can be defined.
3. Model can be divided into components/ submodels.
4. Individual functions of submodels can be defined.
5. It can read, write and save XML file, while maintaining its structural hierarchy.
6. Object-oriented approach can be implemented.
7. Generic algorithm/mathematical models can be defined.
8. It can be executed as standalone application.

App building (MATLAB App Building, 2017) feature in MATLAB, allows the freedom of developing a GUI, which justify the first four required capabilities. MATLAB also offers two different ways in the handling of XML files, which is a sublime requirement for the modeling configurator as stated above, xml2struct and Document Object Model (DOM) node. The first one is the use of xml2struct command which reads and decomposes the whole XML structure to MATLAB structure. This allows quite a flexibility while accessing the data and is capable of handling large XML files as well. Despite these advantages, this technique is not applicable, since, the modeling configurator needs to return the modified XML files back to Mevea for simulation. The main reason for this failure happens as the conversion of MATLAB structure to XML structure produces drastic outcomes and the readability of XML for Mevea is not achieved. The other one is by the use of DOM node, where the whole XML file is stored in this DOM node under an allocated memory which can be called and edited by using java commands according to the attributes as provided by the programming code depending on user selection. In doing so, the approach of object-oriented programming could be easily applied for the development of this research. While applying the second technique, the user-specified configuration is embedded into the existing preprogrammed simulation file, which leads to the generation of newly modified simulation files. This technique quantifies the remaining capabilities and hence makes MATLAB as the most efficient and user-friendly choice for the development of modeling configurator. Once the user has selected the desired configuration and is satisfied, the modeling configurator leads the user directly into simulation environment, where the virtual model of the desired vehicle can be experienced and tested.

In reviewing the mentioned capabilities, MATLAB offers such dimensionality meeting all the required criteria. Whereas, from research and industrial perspective the presence of MATLAB, is indisputable to some extent due to its wide scope of functionalities. In literature also, traces of MATLAB as a tool in modeling and simulation can be found. Therefore, MATLAB was opted as a primal tool after evaluation of the above set of requirements.

3.5 Working of modeling configurator

The whole process of modeling configurator is classified in three stages: initialization, vehicle modeling, and execution. The working of modeling configurator is shown in Figure

19, where the first two step on the right side represent the initialization stage, next two steps represent the vehicle customization stage and the last step represents the execution stage. The first stage of initialization is carried out in two steps. In the first step when the application is executed the graphical user interface is developed. As mentioned earlier, app building feature of MATLAB is utilized for the creation of graphical user interface. In this stage when the app is executed, GUI is generated and in the beginning user is provided with three choices of vehicle selection through the provision of buttons; Tractor, Wheel Loader, and Fork Lifter. Also in the meantime, the respective simulation file for the vehicle is stored via DOM node into the modeling configurator. In the second step once the vehicle has been selected, modeling configurator loads the stored information of selected vehicle against each attribute (engine, tire and gearbox) and displays onto the GUI in the form of tab/panel. Depending on this choice the user is provided with different tabs each representing a different attribute.

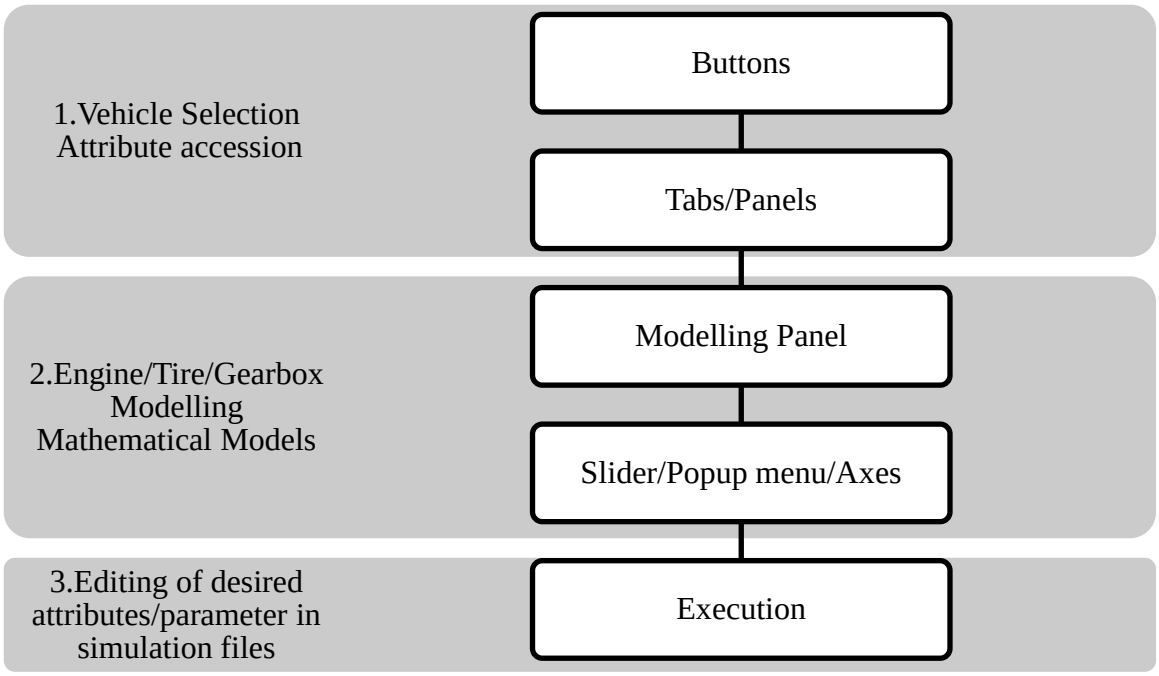


Figure 19. Different stages of modeling configurator.

The second stage of vehicle modeling is the core of this research. It allows the user to model each attribute in the modeling panel by interacting with the parameters of each attribute. The parameters against each attribute as shown in Figure 20 are: rated power and rated speed for engine; section width, aspect ratio and number additional tire set for tire; number of forward

and reverse gears for gearbox. During this stage, based on the generic mathematical models defined in the code, modeling configurator evaluates the altering parameters and simultaneously displays the modifications through visual representations in the GUI. In this stage, the assistance of slider, popup menus, and axes are used to help the user in modifying the attributes as desired. The limits for each parameter are predefined in the first stage upon selection of vehicle. Each slider and popup menu carry their own callback function which are being executed upon user interaction. Furthermore, the generic mathematical model for each attribute is defined which are called upon in the callback function using the approach of nested functions. In this stage, for illustration purposes, axes are also defined which show the effect on each attribute due to changes in different parameters correspondingly.

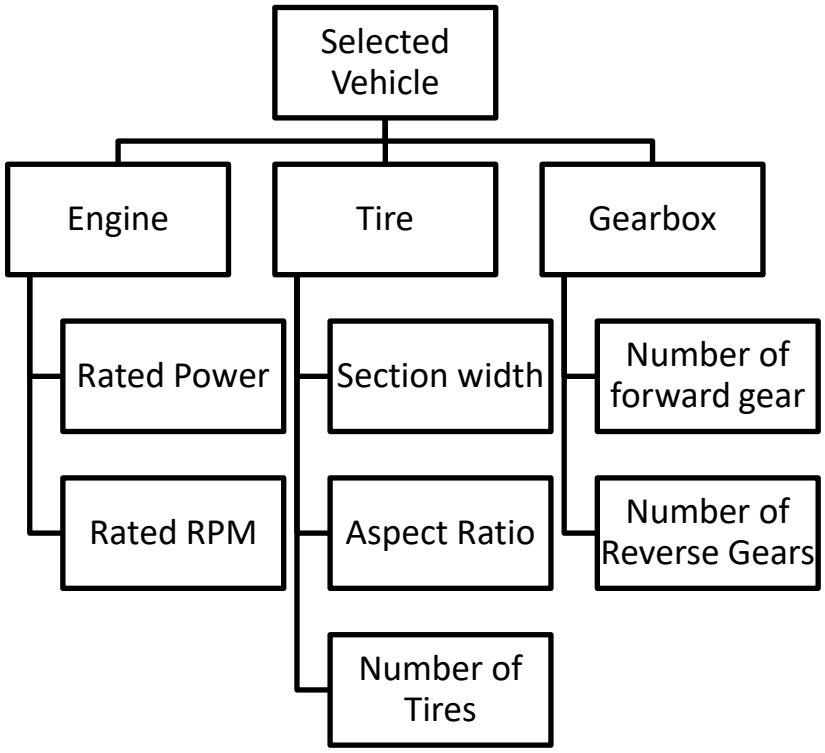


Figure 20. Generic Layout of Vehicle Modelling.

The last stage of execution is all about the editing of simulation files. Modelling configurator acts as a translation medium in this stage, by embedding user customization gathered in second stage into the preprogrammed simulation files. This stage is executed once the desired modification in the attributes have been made. The respective values of the attributes are translated into the stored simulation files and saved as a new file. A generic function is also adopted here for the modification of simulation file so that any model can be modified

pertaining to a predefined variable for each model file. MATLAB allows the closing and opening of files also, therefore, once the user has selected the desired configuration, upon proceeding he/she enters directly into the simulation environment. The simulation environment in this research is provided by Mevea Solver which also has compatibility with simulators and allowing user to experience the modified vehicle on a simulator platform with realistic feedback.

This whole process is carried out by defining different functions for each execution, whereas, to make it more generic the use of nested functions is adopted. The adaptation of all the models for modeling purpose in modeling configurator, solely depends upon their construction in Mevea. As during the construction phase in Mevea Modeller, one might give any name/tag to the attribute. For example, the person can name the attribute (engine) as ‘engine’, ‘engine1’ or ‘engine2’. Similarly, the model can carry one tire model for every wheel or on the contrary separate tire model for each wheel. Due to time restrictions of the project this approach couldn’t be followed. However, methodology of this approach is highlighted in future recommendations. Hence all the pertinent information concerning the adaptability of each model are extracted beforehand from the XML files, keeping in view their construction procedures followed when models were created for simulation purposes.

3.6 MATLAB code algorithm

The code for modeling configurator is programmed in MATLAB editor environment. The program code starts with the creation of a graphical user interface, after which the case variables for each vehicle are defined. In the next stage, the generic mathematical model is defined for each attribute with their respective callback functions which provide a user interactive environment. Whereas, the next stage is executed in two steps. The first step evaluates user selection of vehicle and then defines respective case-specific parameters stored in preprogrammed XML file of each vehicle. The second steps translate the data obtained from user selection and rewrite the model XML stored file earlier upon vehicle selection. A brief illustration of the algorithm followed in the programming code is shown in Figure 21**,** Figure 23**,** Figure 22**,** Figure 24 and Figure 25. The different case specific range for the modifiable attributes in the modeling configurator are given in Appendix II and MATLAB code in Appendix IV.

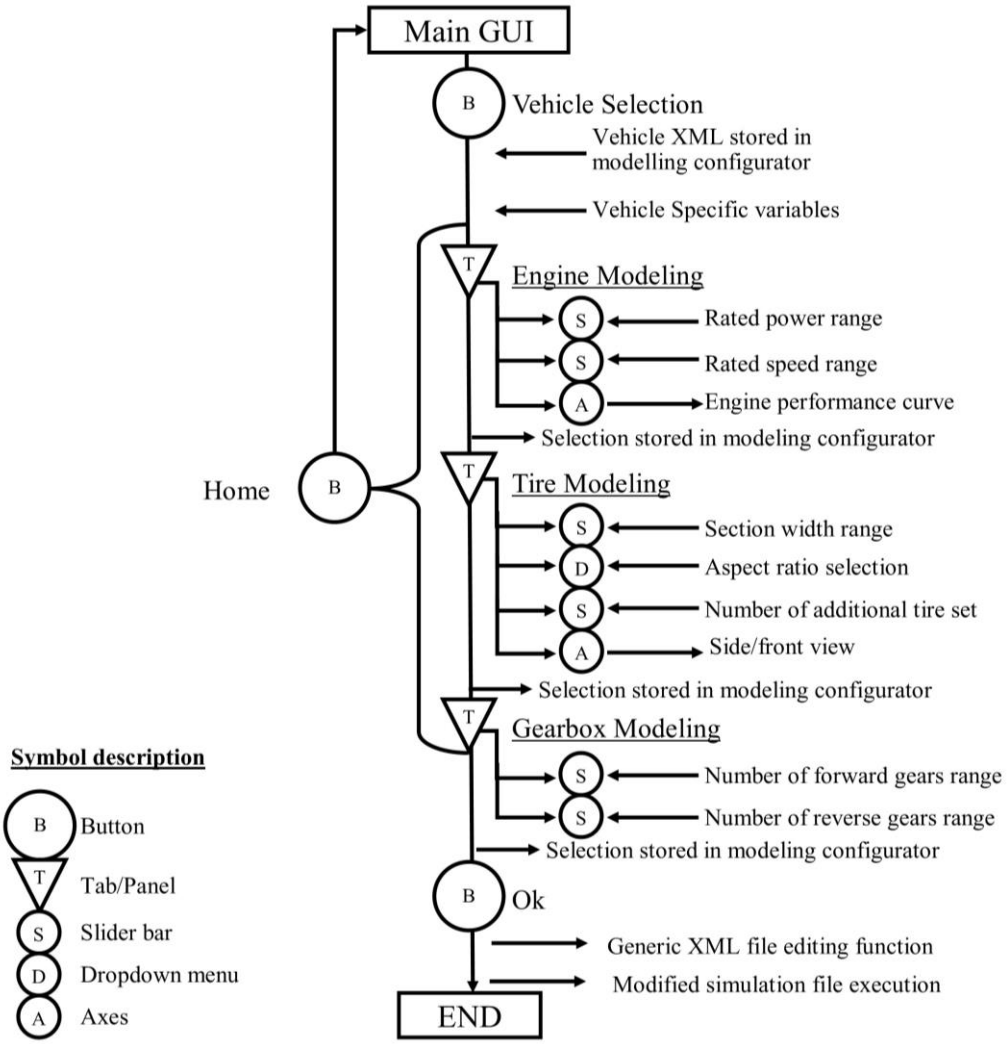


Figure 21. Algorithm of main GUI.

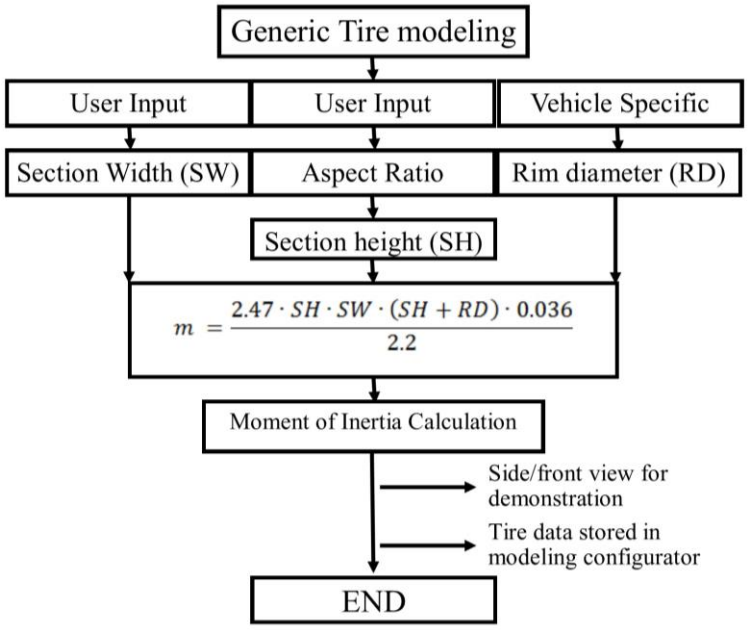


Figure 22. Algorithm of tire modeling.

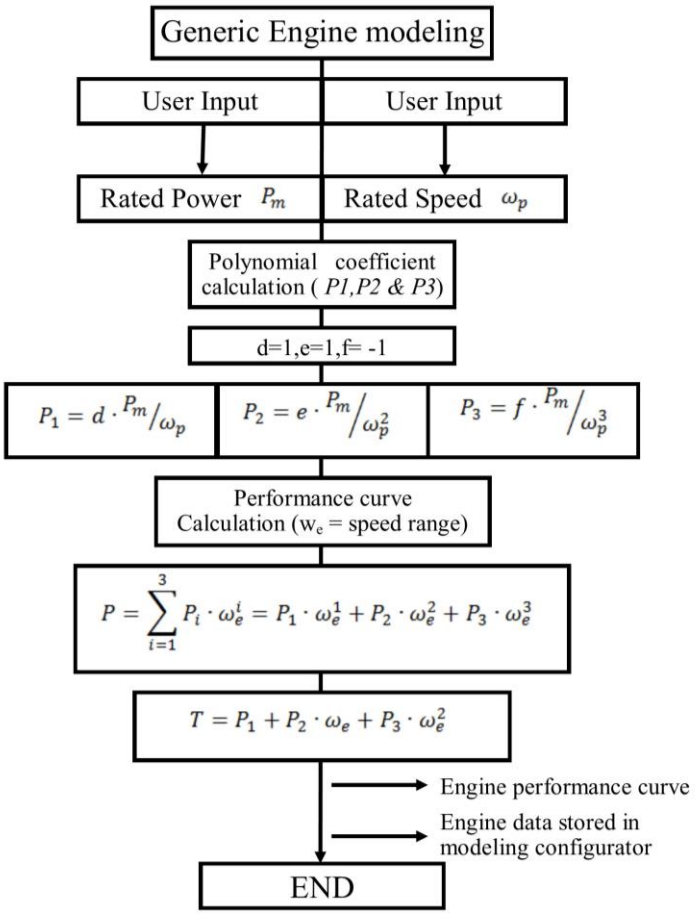


Figure 23. Algorithm of engine modeling.

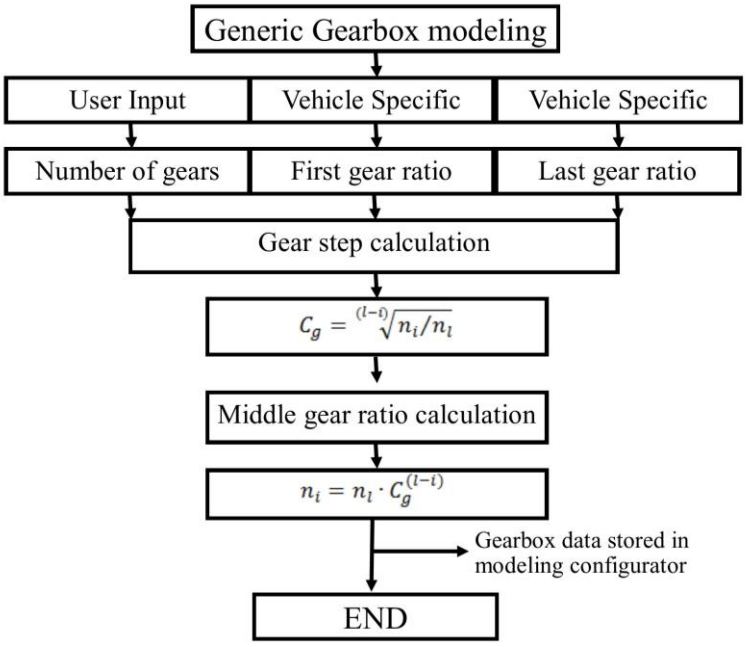


Figure 24. Algorithm of gearbox modeling.

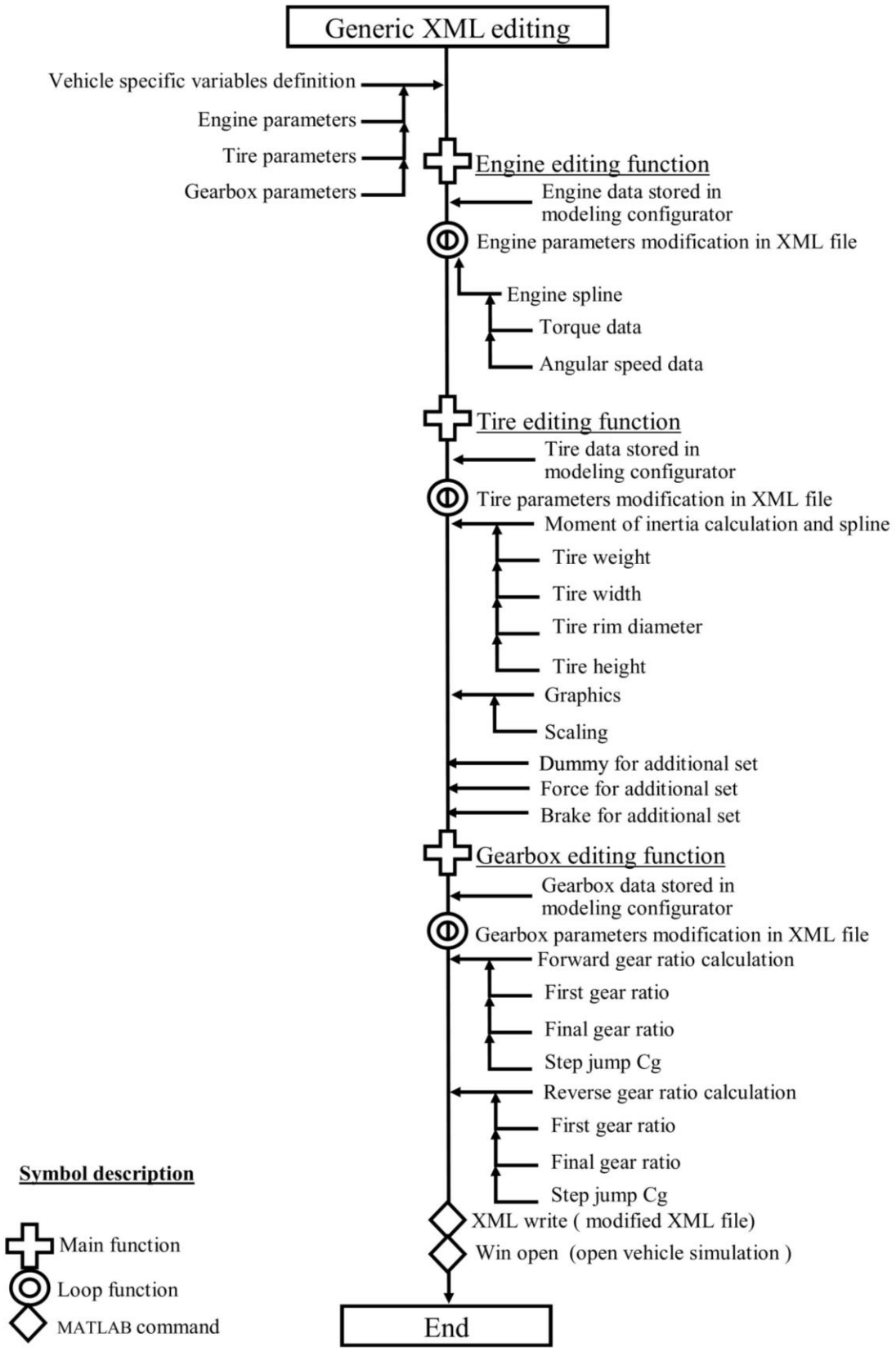


Figure 25. Algorithm of generic XML file editing.

3.7 Case studies

In the scope of this research, the vehicles under consideration are tractor, wheel loader and forklift. The main outcome of this research is the development of generic platform (modeling configurator) that is subtle enough for handling modifications in the stated predefined simulation models. Therefore, the construction and in-depth detail of each model is not discussed and prime focus was given to specific attributes under similarity conditions of all the model. In consideration of these attributes, the selection criteria for each attribute and their performance ranges are discussed under independent case study of each model.

3.7.1 Tractor

The tractor is the main tool needed in the field of agriculture and heavy labor. There are different classifications for tractor depending on the usage. Whereas, the model in practice as shown in Figure 26 for this case study was created for academic purposes in compliance with Valtra T series tractors (Jaiswal 2017, pp. 29-53). The engine power rating typically as found from different company catalogs for Valtra models vary from 56 kW to 294 kW for various engine types depending on the model and operating requirements (Compare Valtra models, 2017). For the series (T series) in consideration, however, power rating vary approximately between ~120-160 kW and the same is provided for user. Also from different catalogs of Valtra, it has been found that the rated speed may vary from 1900-2100 rpm, so in this case study, the provided range for rated speed is the same.

Figure 26. Valtra Tractor Model.

In case of the tire, there are different marking schemes used by the manufacturers which are classified in Table 3, whereas, the visual illustration of these parameters are shown in Figure 27. As mentioned earlier, the tire profile is defined by a spline. The x-values corresponds to the width which can be directly inserted as section width. The y-values correspond to the radius of the tire and can be calculated using equation (26) and (27), where R_e radius of wheel, SW section width, RD rim diameter, SH section heights and AR aspect ratio. In general when the dimensions of the tire have changed, the weight of tire should also change. In the earlier section, it can be found that for inertia calculations in equation (20) and (21), the weight of the tire is also required along with other dimensions. Whereas, the weight of tire can be estimated using equation (22).

Table 3. Different Tire marking as available in the market.

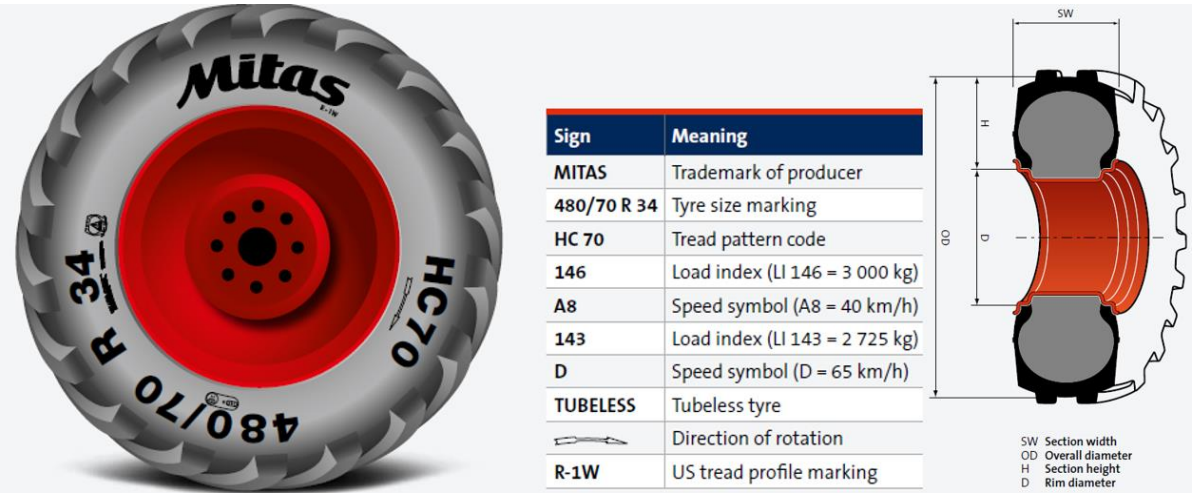
Tire Marking		**Nominal Section width (SW)**	**Aspect Ratio (AR) (Section height SH) SH/SW**	**Construction Type**	**Nominal Rim Diameter (RD)**	**Overall Diameter (OD)**
1.	A/B C D	A	B	C	D	
2.	A/B-D	A	B	-	D	
3.	A*B-C	B		-	C	A
4.	AB-C	A	B(reduced)	-	C	
5.	A-B	A		-	B	
6.	A*B	B				A

$$R_e = \frac{RD + 2(SH)}{2} \tag{26}$$

$$SH = SW \cdot AR \tag{27}$$

The range for section width and aspect ratio were based on different available tires in the market according to these rim diameters. Front tire range is specified from 320-600 mm with aspect ratios of 65,70,80,85 and for rear tire it is specified from 320-900 mm with aspect

ratios of 60,65,70,75,80,85,95. It has been observed that although certain tire sizes are available in the market, but for the particular model, they violate the tolerance limits. In light of such cases, the dimension are made available but with a warning sign so that the user is aware of consequences when considering the installation of such tires in practical. The rim diameter for both front and rear tire are variable and vary from model to model. It has been found from different catalogs that Valtra uses front rim diameter as 28’ and rear rim diameter as 38’ most commonly for their N, S, and T series models (Compare Valtra models, 2017). Therefore, for generalizing the model value for front and rear rim diameter are fixed as 28’ and 38’ respectively.



Sign	Meaning
MITAS	Trademark of producer
480/70 R 34	Tyre size marking
HC 70	Tread pattern code
146	Load index (LI 146 = 3 000 kg)
A8	Speed symbol (A8 = 40 km/h)
143	Load index (LI 143 = 2 725 kg)
D	Speed symbol (D = 65 km/h)
TUBELESS	Tubeless tyre
	Direction of rotation
R-1W	US tread profile marking

Figure 27. Tire marking (Mitas. 2017, pp. 10-12).

As depending on the requirements additional tires can be installed in practical. Similarly, the user is also provided with the option of adding multiple tires at both front and rear. However, keeping in view the constraints of the model as it can be seen in Figure 28 while operating, a limit of 3 is set for both front and rear. If an additional set of tires is allowed exceeding the defined limit it will result in a collision of front and rear tires.

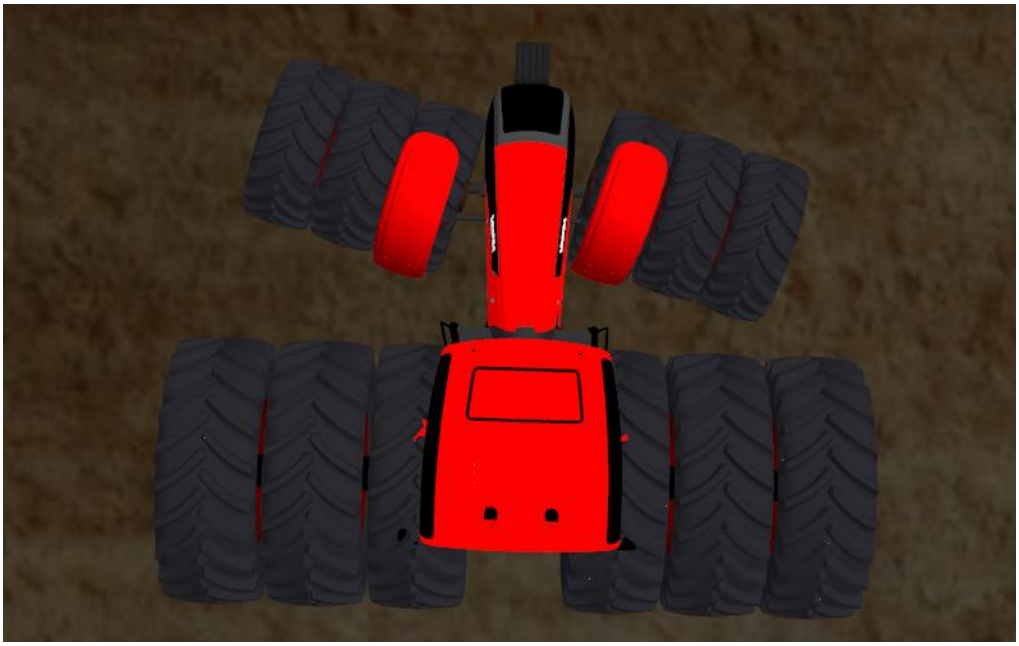


Figure 28. Top view of tractor model.

For the modeling of the gearbox for the tractor, the existing model carries a total six forward gears and 3 reverse gears. In practice, various company catalogs and brochures show that there can be several gear combinations depending on the technology used in the powertrain. For example, Valtra S series of tractor model is implemented with stepless AGCO Variable Transmission. Therefore, for academic purposes only, in this case the range for forwarding gears is provided as 3-10 and for reverse gears, a range of 1-5 is provided. Depending upon user selection, modeling configurator takes case specific predefined gear ratios of first and final gears and designs the gearbox in accordance with the geometric ratio technique as explained earlier.

3.7.2 Wheel loader

Wheel Loaders are serving as a key component in the area of construction and several other purposes. The involvement of high-risk factor in this critical sector means negligence at any point cannot be tolerated, as it may lead to catastrophic results. Therefore, the role of modeling and simulation in the development of this heavy machinery is of dire importance and hence, it leads to the adaptation of this case study as a part of this research. The model for wheel loader as shown in Figure 29 is provided by Mevea which is in accordance with Volvo L70F and is used here for academic purposes only. Due to this reason, the models of Volvo F-series are taken into account in the following case study.

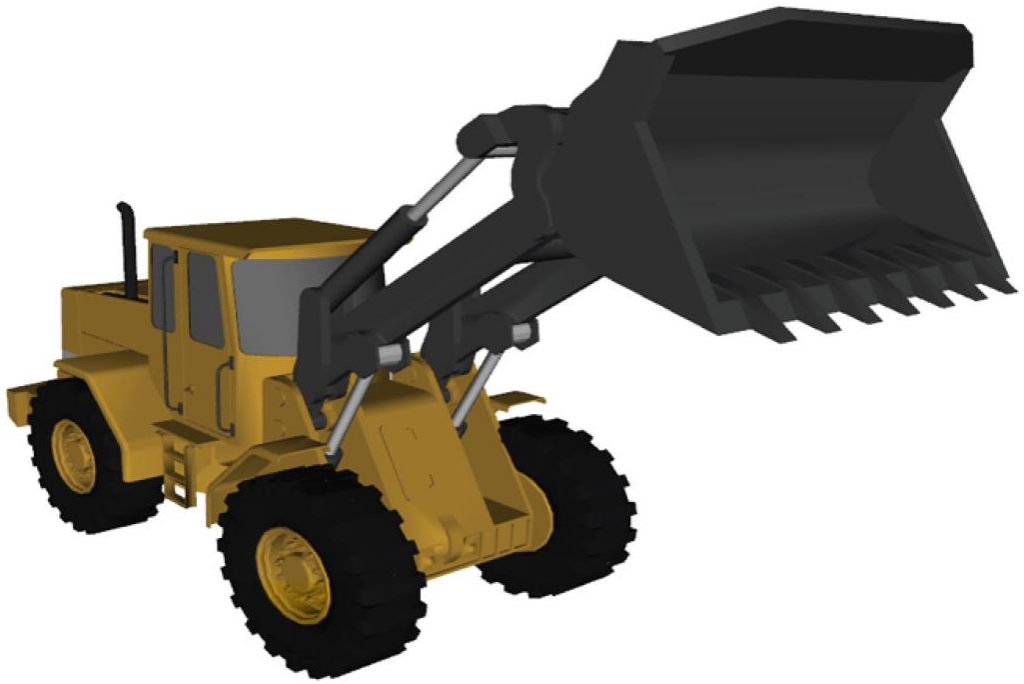


Figure 29. Model of Wheel Loader (MeVEA Modeller [simulation program] 2017b, p. 42).

All the available wheel loaders as manufactured by Volvo are reviewed from the online available catalogs. The attributes for this case study are extracted from the catalogs and summarized in Table 4. The enlisted models were selected in resemblance to tire marking as they commonly have a rim diameter of 25'. As it can be seen in the following table that engine power rating starts from 73.9 kW to a maximum of 259 kW. In case of rated speed, the values start from 1400 rpm to a maximum 2200 rpm. The availability of these large ranges in the modeling configurator may lead to the selection of a model which is either not realistic or it may not be available in the market.

Therefore, keeping in view the base model L70F, for both the available ranges median values were calculated, which resulted in a value of 173 kW for power rating and 1950 rpm for speed rating. In light of these median values for academic/demonstration purposes only, the range for rated power is provided as 110 kW to 190 kW and for rated speed as 1900 rpm to 2100 rpm encompassing model from L70E to L150D. In case of tires, unlike tractor, a wheel loader has same tire size at front and rear end. The user can modify a single tire which will lead to modification of all the tires. Since, all the models under evaluation have the same rim diameter, so the rim diameter is set to 25' for the model.

Table 4. Compact wheel loader models of Volvo (Volvo, - F Series 2017).

Sr.	Model Name	Engine Parameters		Tire marking
		Rated Power (kW)	Rated Speed (RPM)	
1.	L50C	74.6	2200	17.5R25
2.	L50D	74	2200	17.5R25
3.	L50E	73.9	2200	17.5R25
4.	L60E	102	1800	20.5R25
5.	L60F	114	1700	20.5R25
6.	L70D	91	2000	20.5R25
7.	L70E	112	1700	20.5R25
8.	L70F	125	1700	20.5R25
9.	L90C	113	2100	20.5R25
10.	L90D	113	2100	20.5R25
11.	L90E	121	1900	20.5R25
12.	L90F	128	1700	20.5R25
13.	L110E	154	1700	23.5R25
14.	L120C	148	2100	23.5R25
15.	L120D	148	2100	23.5R25
16.	L120E	164	1800	23.5R25
17.	L150C	183	2100	26.5R25
18.	L150D	186	2100	26.5R25
19.	L150E	198	1700	26.5R25
20.	L180C	203	2100	26.5R25
21.	L180D	206	2100	26.5R25
22.	L180E	221	1400	26.5R25
23.	L220D	257	1600	29.5R25
24.	L220E	258	1600	29.5R25
25.	L220F	259	1600	29.5R25

In addition to this, it can also be seen in the above table that for wheel loader aspect ratio of a tire is fixed to 92 as the section width end in a zero so the user can modify the tire in

relation to section width only. This lead to the range of tire section width from 525 mm to 675 mm. In order to calculate the weight of tire for inertia calculations, similar methodology is adopted as used in equation (20), (21) and (22). For the availability of an additional set of tires as shown in Figure 30, there were model constraints that limit the selection of such parameter. To generate a hassle-free simulation environment, therefore, this parameter is restricted to a value of 2.

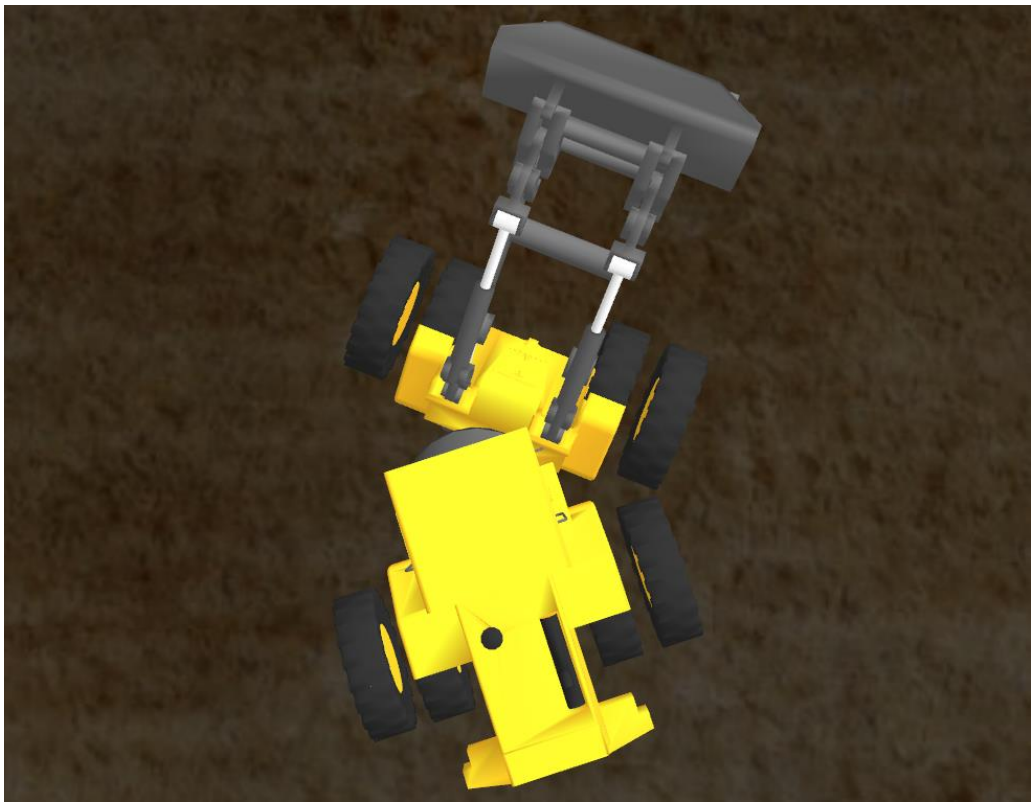


Figure 30. Top view of wheel loader.

The designing of the gearbox for wheel loader is similar to the methodology used in tractor model. However, in practice, the gear limits are defined in respect to the velocity of the vehicle which can be due to the use of progressive ratios used while designing the drivetrain. Most commonly a total of 4 forwards and 4 reverse gears are present, which can be inserted manually or automatically based on the availability of AUTO mode. For additional user experience, this limit has been defined with a combination of 3-10 for forward gears and 1-5 for reverse gear allowing the user to select the desired combination. Depending on the selected combination, modeling configurator designs the gearbox according to geometric ratios.

3.7.3 Forklift

Fork lifter is mainly used for the purpose of handling and transportation of equipment. In industries most commonly they are used indoor for the handling of goods and are smaller in size and power as well. The model in this case study is still not available in SIM studio and used for academic purposes only as shown in Figure 31. Due to the low power requirements along with diesel fork lift, it can be electric or gas powered. As the generic approach is adopted in this research, therefore only diesel-powered counterbalance forklift models are considered.

Figure 31. Model of Forklift.

The attributes modeling of forklift is the same for the engine but slightly different for tire modeling in comparison to the other two case studies explained earlier. The model used in this case study doesn't resemble any commercially available fork lift model. Therefore, as Cat Lift Trucks are the most commonly used forklifts worldwide so this study capitalizes on their available diesel powered counterbalance forklift models as enlisted in Table 5. As it can be seen in the following table that power rating varies from 28 kW to 129 kW but the focus in this case study is for fork lifter smaller in scale. It is due to this reason that models ranging from 28 kW to 55 kW are taken into consideration again for academic / demonstration purposes only. Similarly, in case of rated speed a range of 2200 rpm to 2500 rpm is selected. For the modeling of tire due to the above stated reason for commercial non-availability of similar models, the limitations are defined by constraints of the model in hand and a range of 70 mm to 110 mm is provided for section width of the tire. Whereas, the rim diameter is taken as 250 mm. In case of aspect ratio, it has been observed that unlike previous models, the section height of the tire is approximately the same as section width of the tire.

Therefore, for demostration purposes the aspect ratio for the tire was taken as 100 for this case study. For inertial calculations, similar methodology is adopted as shown in equation (20) and (21). With altering dimensions due to lack of sufficient means to evaluate weight changes thereby the weight is taken as default and kept as a constant. The slight disadvantage however, might be the user feedback at steering wheel. As this was not the prime objective of this research and thereby can be neglected. In case of modeling of the gearbox, the attribute is not included for this case study. Since in resemblance to practical models, virtual model is also designed with an automatic gearbox and hence is not provided to the user.

Table 5. Diesel-powered Forklift models (Counterbalance Lift Trucks 2017).

Sr.#	**Model Name**	**Engine Parameters**		**Tire marking**	
		Rated Power (kW)	**Rated Speed (RPM)**	**Section width (mm)**	**Rim diameter (mm)**
1.	DP15N	28	2500	165	254
2.	DP18N	28	2500	165	254
3.	DP20CN	28	2500	165	254
4.	DP20N	35.3	2250	177	304
5.	DP25N	35.3	2250	177	304
6.	DP30N	35.3	2250	228	381
7.	DP35N	35.3	2250	250	381
8.	DP40N	55	2200	209	381
9.	DP45N	55	2200	300	381
10.	DP50CN	55	2200	300	381
11.	DP50N	55	2200	300	381
12.	DP55N	55	2200	300	381
13.	DP100N	129	2200	254	508
14.	DP120N	129	2200	254	508
15.	DP135N	129	2200	304	508
16.	DP150N	129	2200	304	508
17.	DP160N	129	2200	304	508

4 RESULTS AND DISCUSSION OF GENERALIZED MODELING

The presented approach addresses all the research questions stated earlier. The process of vehicle customization has been made generic and user-friendly as shown in Figure 32. The user can easily select the vehicle and then modify the attributes as desired. At present three type of vehicle were selected tractor, wheel loader, and forklift. Although each vehicle had its own distinct features, still somehow there were some similarities in terms of operating mechanism. In order to encompass all these vehicles for modeling via one platform as discussed earlier, some attributes (engine, tire and gearbox) were selected for modeling based on their similarities. Furthermore, case-specific ranges were defined for the parameters involved in the modeling of each attribute.

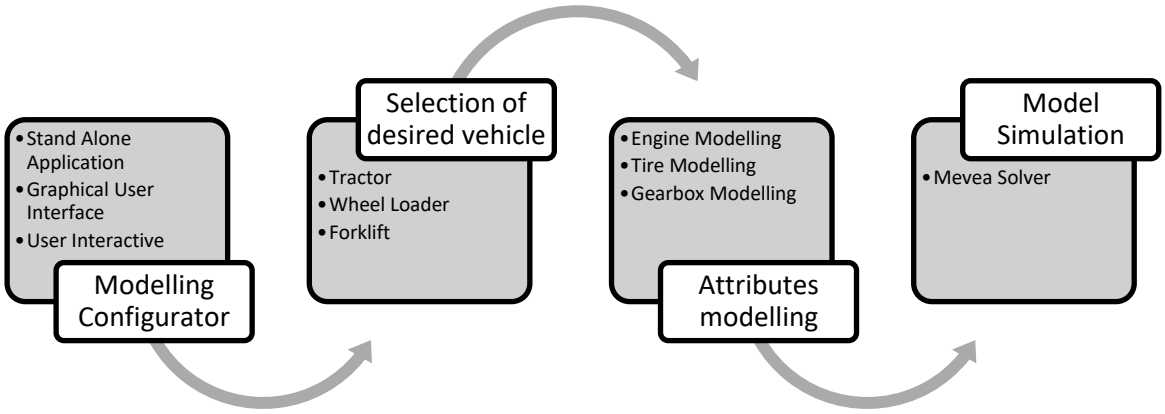


Figure 32. Process flow of editing simulation files.

The presence of a more user-friendly environment was dealt by generating an interactive graphical user interface as shown in Figure 33 by taking assistance from App building feature of MATLAB. Whereas, the object-oriented approach was followed in MATLAB editor while compiling the programming code for generic modeling configurator. The scope of this research is achieved by applying generic approach at two stages. The first generic approach is applied during the mathematical modeling of each attribute. Irrespective of selected vehicle model, the configurator follows the respective mathematical models, however, the ranges for each parameters are case specific. The different ranges for these parameters have been discussed in detail, during their respective case studies. The successful modification of vehicle models via modeling configurator have been shown in appendix III.

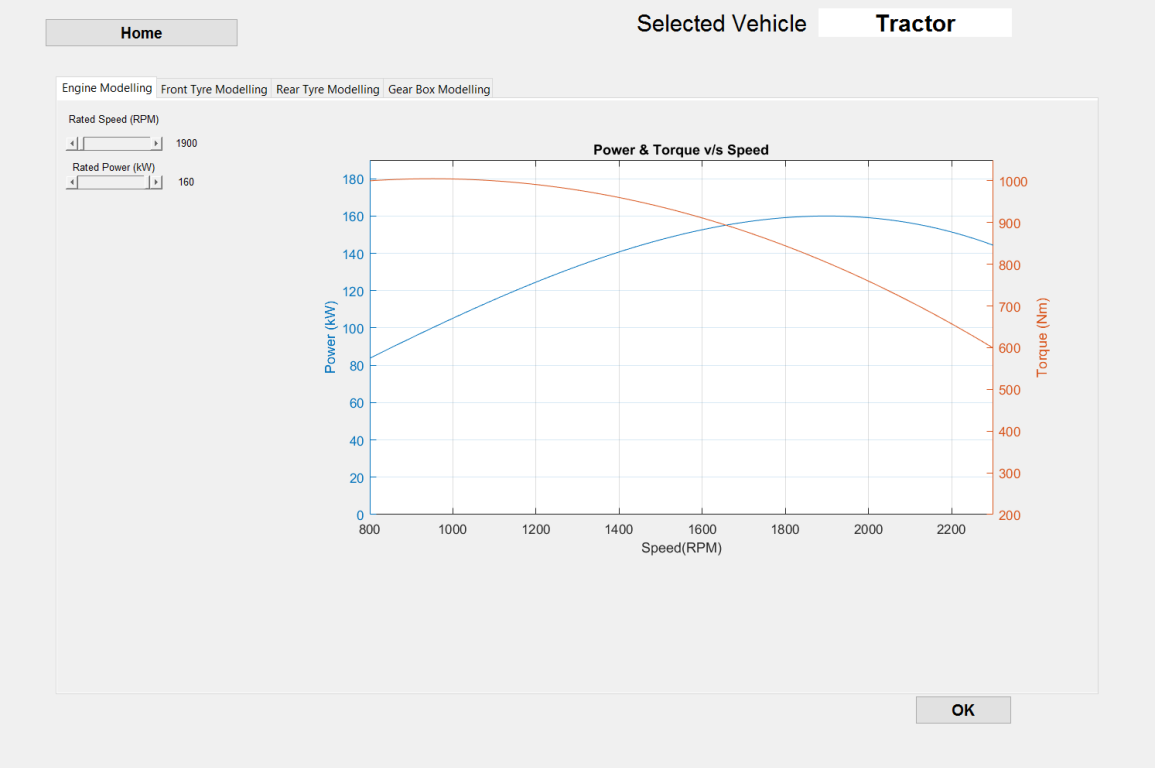


Figure 33. Overview of Modelling Configurator.

The generation of a mathematical models of each attribute were based on some assumptions. These assumptions were made as the user is provided with a wide range of each parameter. Correspondingly, it is highly unlikely to find lookup tables for real data against selected configuration. Thus, leading to configurations whose data or plots were either not available in the literature or hard to find. Also, even if data or plot are generated, still storing this huge amount of data for all the possible configurations that the user might select was impossible and reduced the efficiency of the process. Therefore, on such demands, few assumptions were made in the mathematical modeling such as estimation of engine performance curve by polynomial equation, estimation of mass of tire based on physical dimensions and use of geometric gear step in modeling of gearbox, which are validated in the following section.

4.1 Validation of implemented models

In total three models were used in modeling configurator for respective attributes (engine, tire and gearbox). Furthermore, in development of engine model, three models were discussed in detail. However, model 1 was finalized for the development of modeling configurator on the basis of its accuracy. In order to compare the estimation of these models against real engine performance curves, statistical analysis was performed. To evaluate

model on a broader scope, 8 real engines were selected as reference with different power ratings. Pearson's chi-squared test (goodness to fit) was used as a method for performing statistical analysis. The data obtained from each model was equally distributed into 100 points. Therefore, according to the definition of Pearson's chi-squared test (goodness to fit), the degree of freedom becomes 99. To evaluate this hypothesis at a significance value (probability) of 0.1%, the critical chi-square value is 148.23. In order to pass the Pearson's chi-squared test (goodness to fit), the chi-square value of each engine model against respective real engine data, should be less than this critical value. All the results obtained against this analysis are presented in Figure 34. The results show, that model 1 under this hypothesis can be used for the estimation of power curves for 7 engines out of 8. Also, from the details attached in Appendix V, it was observed that the actual performance curve of this engine did not follow a traditional behavior. It was due to this reason none of the three models passed the hypothesis. If the results of model 2 and model 3 are observed, they failed to pass the test in 4 and 3 cases respectively. This analysis validates the selection of model 1 for estimating the power curve of an engine, where traditional behavior is expected from engine performance curve. Therefore, it is concluded, that the use of model 1 in applications of real-time simulation is an optimum choice.

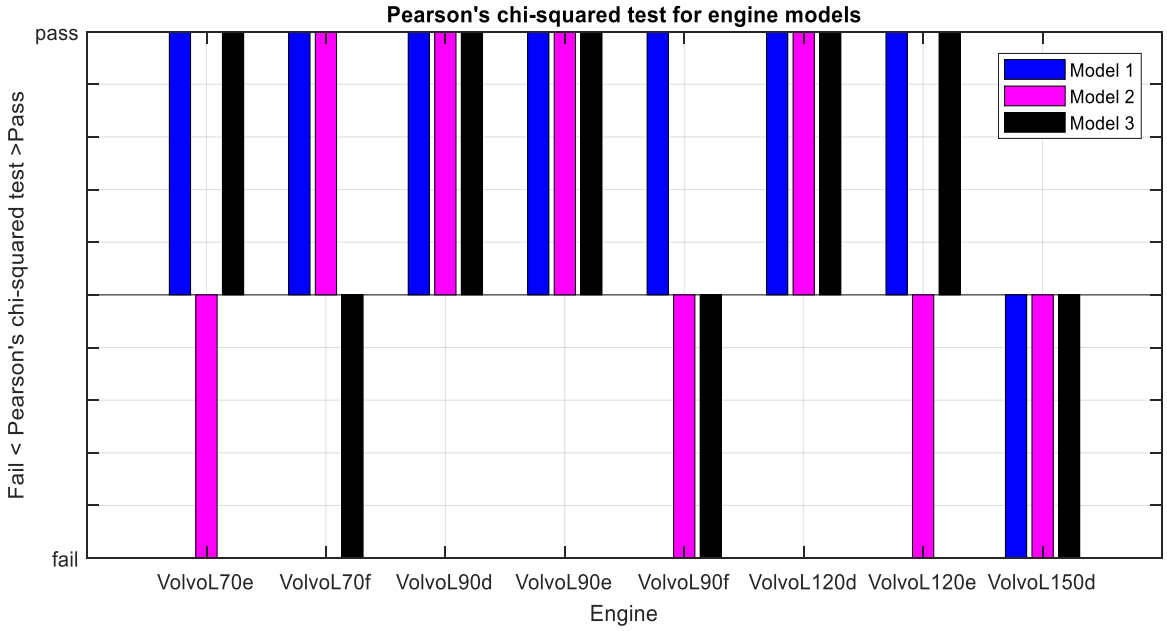


Figure 34. Chi Square test for engine mathematical models.

Modeling configurator also provides user the freedom for customization of size of tire. The alteration in the dimensions/size of the tire directly affect the inherent inertial properties. Therefore, the corresponding modified mass and inertia values were calculated using equation (20-22). It was assumed that the mass is directly proportional to the section height, section width, and diameter of the rim. For the comparison of values, standard tires were selected for both front and rear end, as tires can have different rim diameters.

The market specifications of the tires follows three basic parameters; rim diameter, section width and aspect ratio/section height. For comparing the values of mass obtained from tire model against mass of real data, the aspect ratio is kept constant as 70 and the rim diameter was fixed at 28’ and 38’ for front and rear end respectively. The values were evaluated for both the tires at different section widths. In case of the front tire the standard tire values of 320/70R28, 360/70R28, 420/70R28, and 480/70R28 are selected, whereas, for rear tire 480/70R38, 520/70R38 and 580/70R38 are selected. As, the mass values of these tires were available from manufacturer’s catalogue (TyrFil. 2017) which were included in case study for tractor. Since, the model gives an estimated value for the mass of tire with an error of ±15%. The values of actual mass are plotted against estimated mass along with error bars in relation to the diameter of the tire as shown in Figure 35. It can be seen from the following plots that the actual values are within the tolerance range of calculated values. Furthermore, the plots suggest that since all the three parameters are in direct relation to the mass of tire if two parameters are kept constant the mass increases linearly with increase in values of the third parameter.

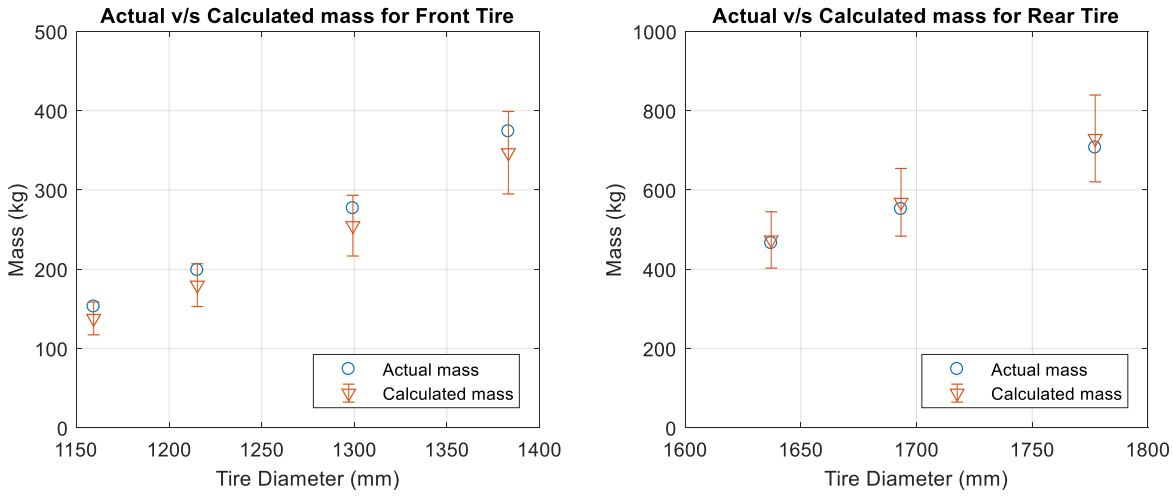


Figure 35. Evaluation of estimated tire mass against actual mass.

The modeling of the gearbox is provided for the model of tractor and wheel loader. Since, the existing model of forklift carries an automatic gearbox configuration. From the literature, it was found that the designing of the gearbox is carried out using two different techniques. As explained earlier the methodology of geometric gear step was followed in this research. For the validation of this methodology, the gear ratios calculated in the gearbox model for the case study of wheel loader are compared with the gearbox model of real data. The specifications as provided by the manufacturer are listed in Table 6 below. By applying equation (23), the gear ratio for first and last gear can be calculated and then the geometric gear step in equation (25) is used to design the gearbox. Later the velocity at intermediate gears in economic working condition is calculated using equation (23). The results obtained from the gearbox model are then compared with manufacturer's specification as shown in Figure 36. Since the gear-speed plot of gearbox model in equation (23) used for modeling configurator, is also in close resemblance to the real model, it also validates the implementation of geometric gear step.

Table 6. Drive Train Specification of real data (Volvo, - F Series 2017).

Model	**Real data**
Torque Multiplication	2.66:1
Forward/Reverse Gear	Maximum speed (km/h)
1	7.4
2	14.3
3	26.5
4	44.0
Measured with tires	20.5R25 L2
Economic working range	w_{min} = 1100 rpm w_{max} = 1600 rpm

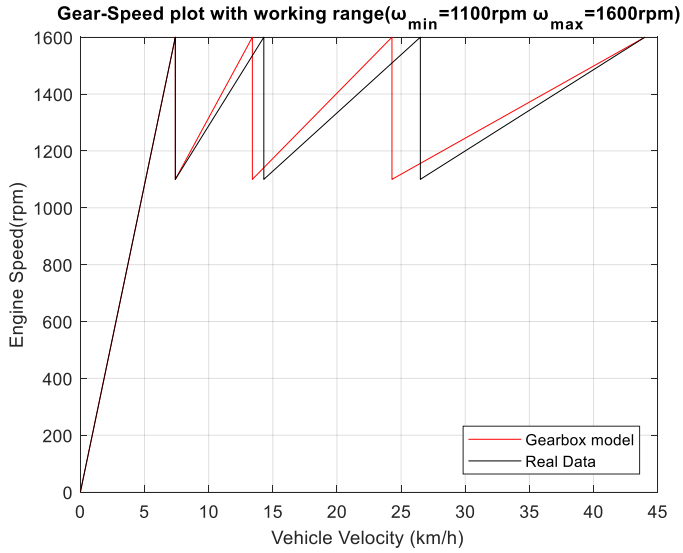


Figure 36. Comparison of Gear-speed plot.

4.2 Modelling configurator

All the corresponding models for tractor, wheel loader and fork lifter were modified with generic modeling configurator and successfully tested in Mevea Solver. Tractor model is selected as a case example for comparing engine performance curve as read by Mevea Solver from the XML file. The engine of the said model was modified using modeling configurator and the response in Mevea Solver is presented in Figure 37. It is observed that the performance curve follows the behavior of a real engine where the torque is maximum at low engine speed and gradually drop to lower values when engine rpm is increased.

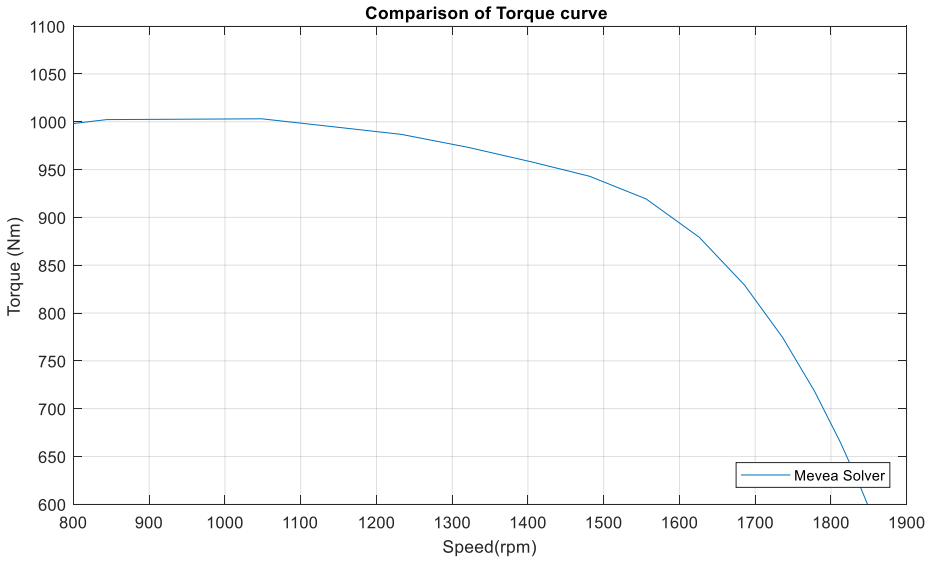


Figure 37. Engine response during the simulation.

Secondly, the results from Mevea Solver are then compared with the torque vs speed maps as inputted by the modeling configurator in the execution stage as shown in Figure 38. The economic working range of the model is between 1100 rpm to 1600 rpm. During the lifetime of a vehicle, it mostly operate in this range, so the behavior of the model is critical within this range. Therefore, the economic working range has been highlighted as a green region in the plot. It can be seen from the following figure that the response of the model in simulation environment over the highlighted range is quite similar to modeling configurator torque curve. Based on this analysis the model promises a realistic response to the user while experiencing the desired configuration in a virtual environment. After the economic range, however, the torque behavior changes in Mevea Solver. This is due to the reason, as stated in equation (1) if the motor rotational reference speed is low, the maximum torque is not achieved and thereby results in lower torque values.

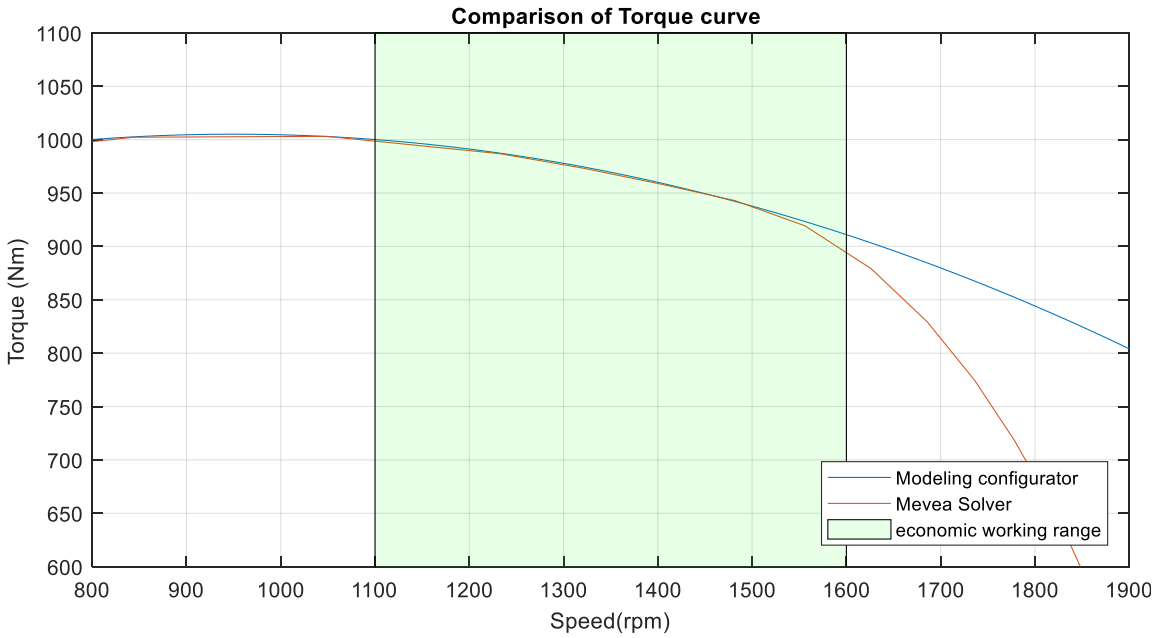


Figure 38. Comparison of torque curve from Mevea Solver and modelling configurator.

In case of gearbox modeling, the tractor model is tested against maximum configuration of 10 forward and 5 reverse gears, whereas, other case studies also produce the same results. The response of the model during several gear shifts in Mevea Solver is shown in Figure 39. The plot shows 10 ascending and 15 descending steps, whereas each step represents the corresponding gear shift as inputted during the simulation of the model. These steps or gear shift validate the modeling of gearbox via modeling configurator against selected

configuration. Also, it can be seen that the peak value of gear index represents the maximum number of forwarding gears. The plot validates the provision of any configuration to the user as desired. Since this is the maximum possible configuration as for now, hence it also verifies that the adopted approach in modeling configurator is capable of handling any configuration selected.

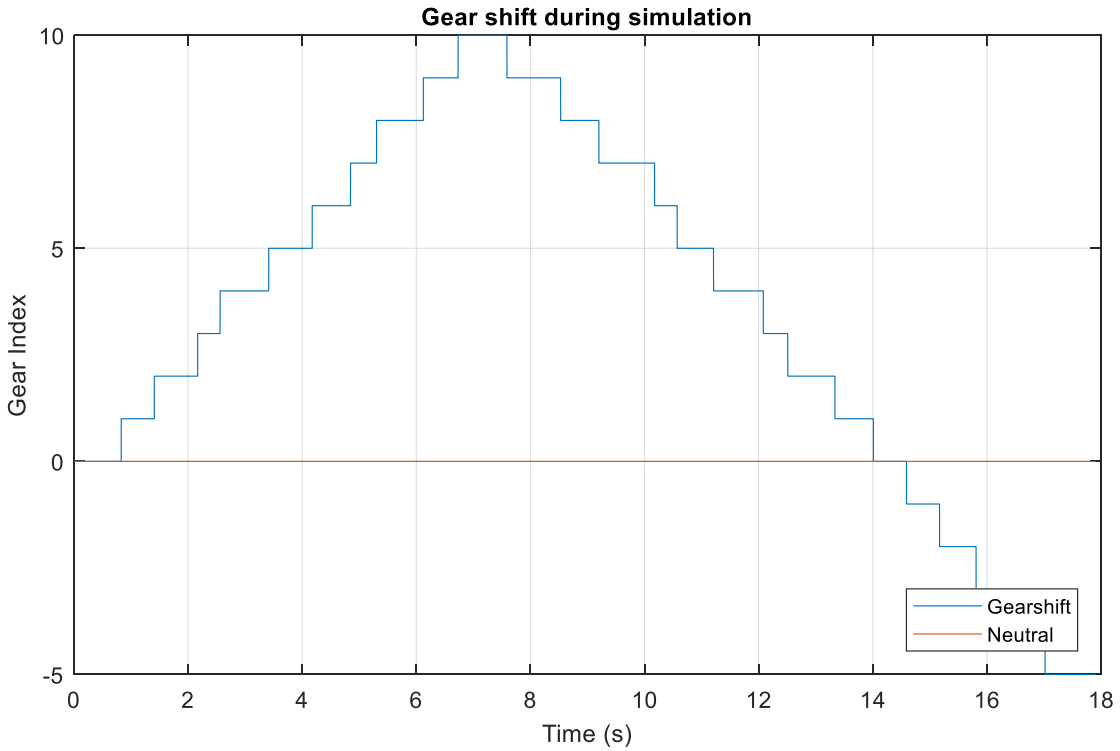


Figure 39. Gear Index plot of the model in Mevea Solver.

5 CONCLUSION

The main outcome of this research was to develop the existing modeling configurator to a more advanced level and generic in form as well. This research addresses all sort of shortcomings in the previous model of modeling configurator along with the solution to the earlier stated research questions. MATLAB was selected as the tool for this purpose which offers two different techniques for XML files that are xml2struct and DOM node with later being used for its compatibility with the addressed requirements. The use of object-oriented programming in MATLAB editor environment with the addition of interactive graphical user interface, modeling configurator serves as a more user-friendly one tool solution. To avoid lengthy coding, a generic approach was adopted for mathematical models extracted from literature against each attribute (engine, tire and gearbox). The newer version of modeling configurator enables the user to select the desired vehicle from different vehicle options. Whereas, the user is provided with a specified range for each parameter, where any value can be selected in between. The limitations were however, defined against each parameter of selected vehicles in respective case studies. For realistic experiences, these limits were extracted from vehicle specific manufacturer data, which were presented in tabular form. Also generic algorithm was adopted for modifying XML files, allowing the use of same function in the programming code for any vehicle.

The methodolgy adopted during the development of mathematical model for each attribute is validated against real data. The performance curve of engine obtained from modeling configurator is compared and validated against real data for several engines. The tire model is validated against real data for both front and rear tire against market available tires for tractor. Similarly, the generic model for gearbox modeling was also compared and vaildated in gear-peed plot against real model of wheel loader. Also, all the vehicle models against various configurations of attributes were succesfully modeled via modeling configurator and tested in the simulation enviroment, where also the modifications were verified.

5.1 Drawbacks and areas of improvement

During the testing, however, it was found that in case of additional tires for tractor model, the steering mechanism encountered oscillation when the input was provided, as the tire

mass attached on both the sides increased. This phenomenon occurs due to the low values of damping and spring constant defined within the steering definition for the model. Thereby, additional function for the editing of steering is created. Hence, when the user opts for an additional set of the tires at the front, such problem is encountered by calling steering editing function and arbitrarily increasing the values for damping and spring constants for smooth user experience. Also, the increase in tire size influences the friction values defined in the model, which are also vaguely addressed in the programming of modeling configurator for providing a smooth virtual response to the user.

In Mevea Modeller there is a limitation of ten inputs for the brake. This means the addition of multiple tires for tractor model with two additional sets both at front and rear tire exceed the limit, as twelve inputs are added for the brake function. However, the modification of model through generic modeling configurator did not encounter any such problems during testing in Mevea Solver. The issue was forwarded to Mevea Oy and will be addressed in the newer version.

5.2 Future implementations

The generic engine model defined by using polynomial equation, serves the purpose for virtual simulation but, however, when the testing of such virtual engine is compared with real engine data the behavior might not always be the same. Therefore, to bring this model closer to realistic engine performance curve, it needs to be defined by the use of NURBS equation, rather than simple polynomial equation. This resulting NURBS equation will be a function of all the performance parameters of an engine as stated earlier in equation (4). Also, the tire model for evaluation of the mass was solely based on one manufacturer, which can be made applicable for other manufacturers as well.

In case of addition of new preprogrammed simulation files (more vehicle models) the modeling configurator needs to be revised with additional coding line for each attribute and parameters defined. This need wasn’t addressed due to time constraint but can be elimintaed in future. Thus, any file produced from Meavea Modeller can modified via modeling configurator without any additional effort. In such way modeling configurator can be dealt as a self-independent tool, where it can modify any number of simulation files without the addition of further programming lines. In this way, generalized approach can be further

extended during the development phase and it can serve as a universal tool for any vehicle model. Similarly, it can be dealt as a separate application or embeded as an integral part of Mevea Solver for instant modifications of the preprogrammed simulation model. Apart from this, the steering mechanism and friction values are also influenced due to parametrization of these attribute. Both of these are directly linked to user feedback. Therefore, generic model can be implemented for these attributes as well and can also be included as an integral part of newer version of modeling configurator.

LIST OF REFERENCES

Ammon Dieter. 2005. Vehicle dynamics analysis tasks and related tyre simulation challenges. Vehicle System Dynamics, 43:sup1. Pp. 30-47.

Auweraer, H. V., Donders, S., Mas, P., & Janssens, K. 2008. Breakthrough Technologies for Virtual Prototyping of Automotive and Aerospace Structures. In: Product Engineering, Tools, and Methods Based on Virtual Reality. Talab, D., & Amditis, A. Dordrecht: Springer, Dordrecht. Pp. 397-418.

Bakker, E., Pacejka, H., & Lidner, L. 1989. A New Tire Model with an Application in Vehicle Dynamics Studies. SAE Technical Paper 890087.

Besselink, I.J.M., Pacejka, H.B., Schmeitz, A.J.C. & Jansen, S.T.H. 2005. The MF-Swift tyre model: Extending the Magic Formula with rigid ring dynamics and an enveloping model. JSAE Review, Vol. 26:2. Pp. 245-252.

Bureau of Labor Statistics, 2016. Census of Fatal Occupational Injuries Charts, 1992-2015. [web document]. [Referred 12.3.2017]. 22 p. Available in PDF-file: https://www.bls.gov/iif/oshwc/cfoi/cfch0014.pdf

Caldwell, M. R., & Fernandez, A.J.1998. A generic model of hierarchy for systems analysis and simulation. Agricultural Systems, 57:2. 197-225.

Canudas de Wit, C., Olsson, H., Astrom, K. and Lischinsky, P. 1995. A new model for control of systems with friction. IEEE Transactions on Automatic Control. 40:3. Pp.419-425.

Canudas de Wit, C. & Tsiotras, P. 1999. Dynamic tire friction models for vehicle traction control. Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304), Phoenix, AZ. Vol.4. Pp. 3746-3751

Chiara F., Wang J., Chinmaya B. P., Hsieh M. & Yan, F. 2011.Development and experimental validation of a control-oriented Diesel engine model for fuel consumption and brake torque predictions. Mathematical and Computer Modelling of Dynamical Systems, 17:3. Pp.261-277.

Chen W. 2012. The Simulation for Automobile Power Performance and Fuel Economy. Applied Mechanics and Materials, Vols. 253-255. 2013. Pp. 2135-2138,

Chindamo, D., Gadola, M., & Romano, M. 2014. Simulation tool for optimization and performance prediction of a generic hybrid electric series powertrain. International Journal of Automotive Technology, 15:1, Pp.135–144.

Compare Valtra models. 2017. [Valtra, Technical Specifications webpage]. [Referred 22.12.2017]. Available: http://www.valtra.com/productcompare.aspx

Counterbalance Lift Trucks 2017. [Cat Diesel Forklift Truck Models and Specifications webpage]. [Refered 25.12.2017]. Available: https://www.catlifttruck.com/products/counterbalance-lift-trucks/diesel-powered-forklift-trucks

Crossley P. R. & Cook J. A., 1991. A nonlinear engine model for drivetrain system development," Control 1991. Control '91. International Conference on. Edinburgh, 1991. Vol.2. Pp. 921-925.

Eldred, M., Hart, W., Bohnhoff, W., Romero, V., Hutchinson, S., & Salinger, A. 1996. Utilizing object-oriented design to build advanced optimization strategies with generic implementation. In: 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization. Bellevue: United States. Dept. of Energy. Pp. 1568-1582.

Embley, D. W., Liddle, S. W., & Pastor, Ó. 2011. [Chapter 1:] Conceptual-Model Programming: A Manifesto. In: Handbook of Conceptual Modeling. Embley, D. W., & Thalheim, B. Berlin: Springer-Verlag Berlin Heidelberg. Pp.3-16.

Eriksson L., Wahlström J., & Klein M. 2010. Physical Modeling of Turbocharged Engines and Parameter Identification. In: Automotive Model Predictive Control. Thoma, M., Allgöwer, F. Morari, M. London. Springer. Lecture Notes in Control and Information Sciences, Vol 402. Pp. 53-71.

Farzin P., Mehdi, P. M. A., & Mansour, B. 2012. Baseline Tuning Methodology Supervisory Sliding Mode Methodology: Applied to IC Engine. IAES International Journal of Advances in Applied Sciences. 1. Pp.116-124.

Ferryman, J. M., Worrall, A. D., Sullivan, G. D., & Baker, K. D. 1995. A generic deformable model for vehicle recognition. In: BMVC '95 Proceedings of the 1995 British conference on Machine vision. 1. Birmingham: BMVA Press Surrey, UK. Pp.127-136.

Février, P. & Fandard, G.2008. Thermal and mechanical tyre modeling for handling simulation. ATZ Worldwide, 110: 5. Pp.26-31.

Gallrein, A. & Bäcker, M. 2007. CDTire: A tire model for comfort and durability applications. Vehicle System Dynamics, 45 :(SUPPL. 1). Pp. 69-77.

Gargouri, F., Franc, C., Ducateau, O., & Boufare`s, F. 1997. Towards a generic model for object-oriented information system modeling. Journal of Intelligent Manufacturing, 8:1. Pp.31–39.

Generic engine. 2017. [Internal combustion engine with throttle and rotational inertia and time lag - MATLAB - MathWorks Nordic webpage]. [Refered 15.10.2017]. Available: https://se.mathworks.com/help/physmod/sdl/ref/genericengine.html

Guo, K. 1989. A Unified Tire Model for Vehicle Dynamic Simulation Relation to Steering Braking and Driving. The 5th International Pacific Conference on Automotive Engineering. Beijing. SAE NO. 891189.

Guo K. & Lu D. 2007.UniTire: unified tire model for vehicle dynamic simulation. Vehicle System Dynamics, 45:S1, Pp.79-99.

Guzzella, L. & Sciarretta, A. 2013. Case Study 3: IC Engine and Flywheel Powertrain In: Vehicle Propulsion Systems: Introduction to Modeling and Optimization. Guzzella, L. & Sciarretta, A. Third Edition. Springer Heidelberg New York Dordrecht London. 2013. Pp.283-290.

Gyimesi, M. 2008. Web Services with generic simulation models for discrete event simulation. Mathematics and Computers in Simulation, 79:4. Pp.964-971.

Harald N., Bernd B., Joachim R. & Wolfgang N. 2011. Power Conversion: Selecting the Ratios. In: Automotive Transmissions: Fundamentals, Selection, Design and Application. Springer Heidelberg Dordrecht London New York. 2001. Pp. 109-114.

He, Z., Liu, G., Wang, H., & Yang, X. 2008. Research on Collaborative Simulation Platform for Mechanical Product Design. In: Global Design to Gain a Competitive Edge. Yan, X.T., Eynard, B., Ion, W. J. Pp. 459-467.

Health and Safety Executive 2017. Fatal injuries arising from accidents at work in Great Britain 2017 [web document]. [Referred 13.11.2017]. 16 p. Available in PDF-file: http://sbel.wisc.edu/documents/TR-2014-15.pdf

Heywood, J. B. 1988. Internal combustion engine fundamentals. McGraw-Hill. Series in mechanical enginerring.

Jacobs, S. 2006. Foundation XML for Flash. Edition 1. Apress.

Jaiswal, S. 2017. Modeling configurator for real-time simulator of a tractor. Master's thesis Lappeenranta University of Technology, Mechanical Engineering. 71 p.

Jan, H. 2016. Autonomous Systems Operationalization Gaps Overcome by Modelling and Simulation. In: Modelling and Simulation for Autonomous Systems. J. Hodicky. Cham: Springer International Publishing. Pp. 40-47.

Jazar, R. N. [Chapter 4:] Driveline Dynamics. In: Vehicle Dynamics: Theory and Application. Boston: Springer, Boston, MA. 2008. Pp.165-216.

Jiang F., Molin W., & Lin L. 2012. Software Design of Engine Characteristic Simulation. Journal of Software. Vol. 7:2. Pp. 316-321.

Jiang, L., Lin, B., Guo, K., & Li, S. 2012. Modeling and simulation of standing still maneuver for driving simulator. World Automation Congress Proceedings 2012, Puerto Vallarta, Mexico. 24-28.06.2012. Pp. 1-3.

Jiang, M. 2011. Virtual Reality Boosting Automotive Development. In: Virtual Reality & Augmented Reality in Industry. Ma, D.,Gausemeier J.,Fan, X.,Grafe M.Shanghai Jiao Tong University Press, Shanghai. Pp.171-180.

Julian, W. 2009. Automotive development processes: processes for successful customer oriented vehicle development. Heidelberg: Springer-Verlag Berlin Heidelberg.

Kaikko, E-P. 2015. Development of Generic Simulation Model for Mobile Working Machines. Master's thesis Lappeenranta University of Technology, Mechanical Engineering. 77 p.

Kirchner, S., & März, L. 2002. Towards self-adaptive production systems: Modular generic simulation models for continuous replanning and reconfiguration. International Journal of Production Research, 40:15. Pp.3627-3640.

Kuiper E. & Oosten V. J. J. M. 2008. The PAC2002 advanced handling tire model. Vehicle System Dynamics, 45:sup1. Pp.153-167.

Kuznetsov A. G., Sergey V. K. & Dmitriy S. V. 2016. A mathematical model of a diesel engine for simulation modelling of the control system. Global Journal of Pure and Applied Mathematics, 12:1. ISSN 0973-1768. Pp. 213-228.

Lee, J.J., Powell, R.E., Ni, A.E. & Tsou, P. 1999. Integration of SEA tire model with vehicle model. Noise and Vibration Conference and Exposition. Traverse City, MI, United States. 17 May 1999 through 20 May 1999.

Lee, S., & Rolland, C. 2000. Achieving Reuse Via a Generic Model of Reusable Components. International Journal of Modelling and Simulation, 20:1, Pp.12-19.

Limin, X. 2002. Vehicle shape recovery and recognition using generic models. Proceedings of the 4th World Congress on Intelligent Control and Automation (Cat. No.02EX527). 2, Shanghai. IEEE. Pp. 1055-1059.

Liu, Q., Solis, D. & Pan, W. 2002. Analysis of the STI tire model. SAE Technical Papers.

Marcin S., Andrzej A., Antoni J. & Andrzej W. 2007. Mathematical model of four-stroke combustion engine working process. Journal of KONES Powertrain and Transport, Vol.14:3 Pp. 587-595.

Marghitu, D. B., & Dupac, M. 2012. Advanced Dynamics: Analytical and Numerical Calculations with MATLAB. Edition 1. New York. Springer-Verlag New York.

MATLAB 2017. App Building. [web document]. [Referred 16.09.2017]. 522 p. Available in https://www.mathworks.com/help/pdf_doc/matlab/buildgui.pdf

MeVEA Modeller [simulation program]. 2017a. Version 7.70.2476. Reference Manual for Solver Library 7.70. (help menu: Mevea Solver reference manual). 206 p.

MeVEA Modeller [simulation program]. 2017b. Version 2.0. Mevea Modeller: Beginner tutorials, Version 2.0 (help menu: Show tutorials). 52 p.

Mitas. 2017. Agricultural Tyres Technical Databook Edition 2017/2018. [web document]. [Referred 02.11.2017]. 224 p. Available in PDF-file: http://www.mitas-tyres.com/underwood/download/files/mitas_agro_catalogue_2017-2018_a5_en.pdf

Mittal, S., & Frayman, F. 1989. Towards a generic model of configuration tasks. IJCAI'89 Proceedings of the 11th international joint conference on Artificial intelligence, 2. Michigan: Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.1989. Pp. 1395-1401.

Na, S.D., Park, D.W. & Yoo, W.S. 2017. Rigid ring with Bouc-Wen tire model for vehicle dynamic analysis. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 231:19. Pp. 3530-3540.

Nair, A., & Rajagopal, K. R. 2010. Generic model of an electric vehicle for dynamic simulation and performance prediction. 2010 International Conference on Electrical Machines and Systems. Incheon. IEEE. Pp. 753-757.

Neagu, G. 1997. Conceptual Modelling in Generic Prototyping Approach. IFAC Workshop on Manufacturing Systems: Modelling, Management, and Control, 30. Vienna. Pp. 345-350.

Negrus, E. & Cocosila, M. 1997. Flexible model to simulate wheel pass over singular road obstacles. Vehicle System Dynamics, 27:(sup001). Pp.322-325.

Ni D. & Henclewood D. 2008. Simple Engine Models for VII-Enabled In-Vehicle Applications. IEEE Transactions on Vehicular Technology. Sept. 2008. 57:5. Pp. 2695-2702.

Noor C.W. M., Mamat R., Najafi G., Yasin M.H. M., Ihsan C.K. & Noor M.M. 2016. Prediction of marine diesel engine performance by using artificial neural network model. Journal of Mechanical Engineering and Sciences, 10:1. e-ISSN: 2231-8380. Pp1917-1930.

Oosten Van, J. & Pacejka, H.B. 2000. SWIFT-tyre: An accurate tyre model for ride and handling studies also at higher frequencies and short road wavelengths. International ADAMS User Conference. Orlando. 19-21 June, 2000.

Paul W. 2018. The Hohenheim Tyre Model: A validated approach for the simulation of high volume tyres – Part I: Model structure and parameterization. Journal of Terramechanics, Volume 75, 2018, Pp. 3-14, ISSN 0022-4898.

Pidd, M. 1992. Guidelines for the design of data-driven generic simulators for specific domains. Simulation, 59:4. 1992. Pp.237-243.

Rajanna, S., Williams, E. J., Ülgen, O. M., & Rothe, V. 2012. Development And Use Of A Generic AS/RS Sizing Simulation Model. 26th EUROPEAN Conference on Modelling and Simulation. Koblenz. Pp. 549-555.

Rakotomamonjy, A., Riche, R. L., Gualandris, D., & Harchaoui, Z. 2008. A comparison of statistical learning approaches for engine torque estimation. Control Engineering Practice, 16:1. Pp.43-55.

Reif, K. 2015. Gasoline Engine Management: Systems and Components. Wiesbaden. Springer Vieweg.2015.

Santayana, G. 1905. The Life of reason: Reason in Common Sense. Volume I. Charles Scribner's Sons.

Sylvain D , Julien F & Franz G. (Gie Sara). 1997. A tyre model for interactive driving simulators. Vehicle System Dynamics, 27:S1, Pp.326-329.

Tian, Y., Yan, Y., Parkin, R. M., & Jackson, M. R. 2008. Development of a Visualized Modeling and Simulation Environment for Multi-domain Physical Systems . In: Global Design to Gain a Competitive Edge. Yan, X.T., Eynard, B., Ion, W. J. Pp. 469-478.

TyrFil. 2017. Weight Estimation Guide. [web document]. [Referred 07.12.2017]. 52 p. Available in https://www.accellatirefill.com/wp-content/uploads/Weight-Book.pdf

Verma, J.P. [Chapter 3:] Chi-Square Test and Its Application. In: Data Analysis in Management with SPSS Software. Springer, India. 2013. Pp.69-101.

Volvo, - F Series 2017. [Introduction - Volvo, -F series : Volvo Construction Equipment webpage]. [Refered 26.12.2017]. Available: https://www.volvoce.com/global/en/product-archive/wheel-loaders/volvo-f-series/

Weyrich, M., & Steden, F. 2013. Automated Configuration of a Machine Simulation Based on a Modular Approach. In: Smart Product Engineering, Lecture Notes in Production Engineering. Abramovici, M. & Stark, R. Berlin: Springer. Pp. 603-612.

Wilhelm, R., & Seidl, H. 2010. Compiler Design: Virtual Machines. Heidelberg: Springer.

Åsbogård, M. 2004. Evaluating Hybrid Vehicle Concepts using Optimal Control and Generic Powertrain Modelling. IFAC Proceedings Volumes, 37:22, Pp.469-474.

Literature review

Literature review for generic modeling is simulation

Database	Keywords	Results
ProQuest	Title (‘Generic’ AND ‘Model’) Abstract (‘Vehicle’) Title (‘Generic’) Abstract (‘Vehicle’ AND ‘User’ OR ‘Parameter’)	2 82
ScienceDirect	Title (‘Generic Vehicle’) Ti, Ab, Key (‘User’ OR ‘Parameter’) Title (‘Generic Model’) Ti, Ab, Key (‘Vehicle Machine User’)	14 79
Scopus	Title (‘Generic’ AND ‘Model’) Ti, Ab, Key (‘User’ AND ‘Vehicle OR Machine’) Title (‘Generic’ AND ‘Model’) Ti, Ab, Key (‘User OR Parameter’ AND ‘Vehicle OR Machine’)	14 49
SpringerLink	Title (‘Generic Model’) At least one word (‘Vehicle Machine’)	28
Google Scholar	Title (‘Generic Model’) At least one (‘Vehicle OR Machine) (in title) Title (‘Generic Model User’) At least one (‘Vehicle OR Machine) (in title)	17 8
Directory of Open Access Journals	Abstract (‘Generic Model Machine’)	23
Emerald Insight	Title (‘Generic Model’)	8
IEEE Xplore	Title (‘Generic Model’) Ab(‘Vehicle’) Title (‘Generic Model’) Ab(‘Machine’)	3 5
Taylor & Francis Online	Title (‘Generic Model’) Anywhere (‘Vehicle OR Machine’	44
Total included after scrutinizing (title> duplicity>abstract >conclusion)		15

Literature review

Literature review for generic engine modeling.

Database	Keywords	Results
ProQuest	Title (‘Engine’) AND Abstract (‘mathematical model’) AND Abstract (‘Polynomial’)	14
ScienceDirect	With words in title, abstract or keywords (‘Engine AND mathematical model AND polynomial’)	7
Scopus	Title (‘Engine’) AND Abstract (‘mathematical model’) AND Abstract (‘Polynomial’)	12
SpringerLink	Title (‘Engine’) At least one word (‘polynomial’) Exact phrase (‘mathematical model’)	42
Google Scholar	Exact phrase (‘engine mathematical model’) At least one word (‘polynomial’)	52
Emerald Insight	Title (‘Engine’) AND Abstract (‘mathematical model’) AND Abstract (‘Polynomial’)	52
Taylor & Francis Online	Title (‘Engine’) keyword (‘mathematical model) Title (‘Engine’) keyword (‘mathematical model)	37 8
Total included after scrutinizing (title> duplicity>abstract >conclusion)		12

Literature review

Literature review for tire modeling

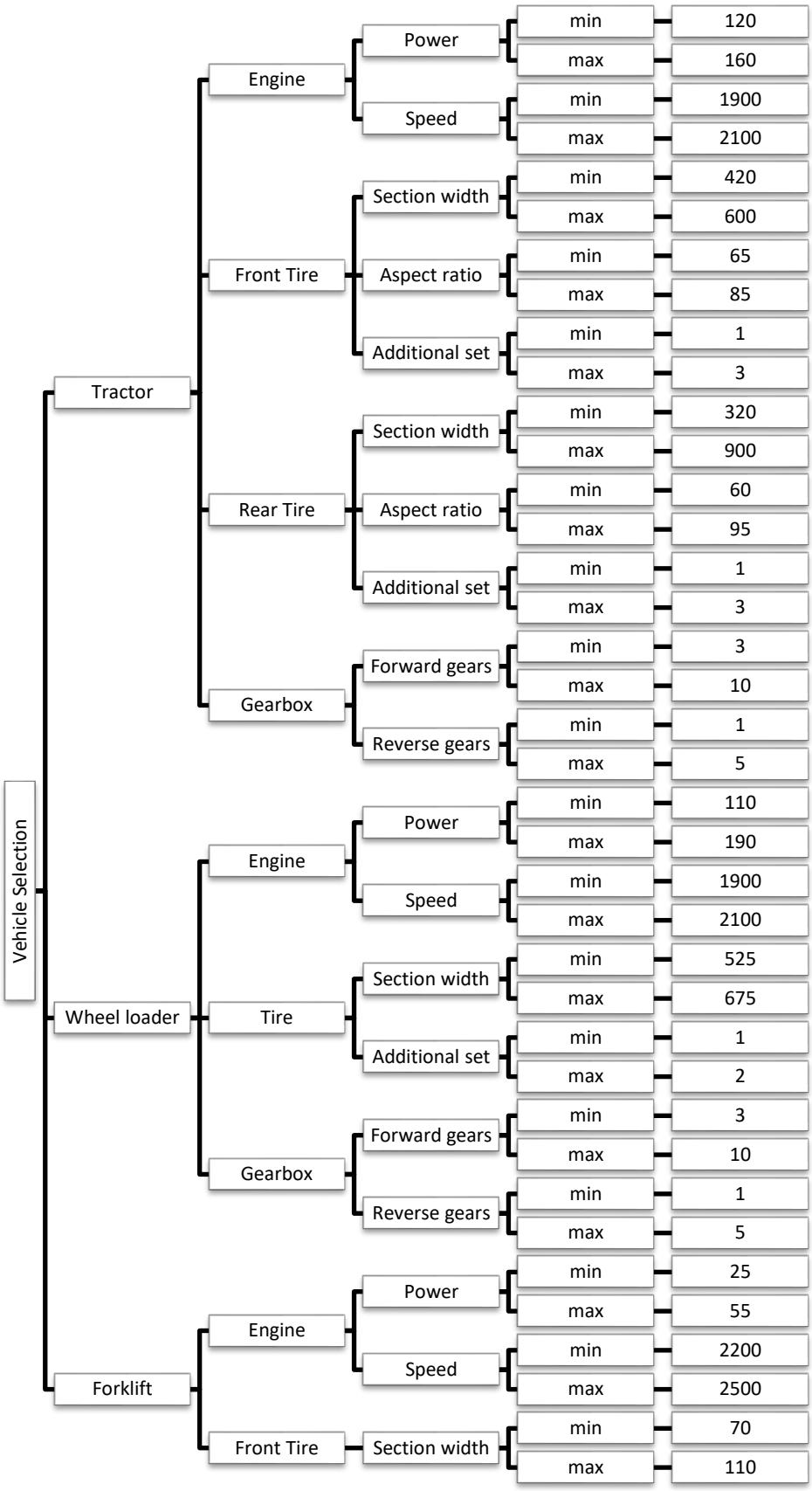
Database	Keywords	Results
ProQuest	Title (‘Tire Model’) Abstract (‘mathematical)	53
ScienceDirect	With words in title, abstract or keywords (‘Tire model mathematical’)	113
Scopus	Title (‘tire model’) Abstract (‘mathematical model’)	69
SpringerLink	Title (‘Tire Model’) Exact phrase (‘mathematical)	7
Google Scholar	In the title of artice With all of the words (‘simulation’) Exact phrase (‘tire model’) At least one (‘mathematical model’)	127
Taylor & Francis Online	Title (‘Tire Model’) Anywhere (‘mathematical)	59
Total included after scrutinizing (title> duplicity>abstract >conclusion)		16

Literature review

Literature review for gearbox modeling

Database	Keywords	Results
ProQuest	Title (‘gearbox modeling) Abstract (‘mathematical)	5
ScienceDirect	With words in title, abstract or keywords (‘gearbox mathematical modeling’)	30
Scopus	Title (‘gearbox model) Keyword (‘mathematical’)	14
SpringerLink	Title (‘gearbox’) Exact phrase (‘mathematical model’)	7
Google Scholar	In the title of artice With all of the words (‘mathematical) Exact phrase (‘gearbox)	24
Taylor & Francis Online	Title (‘Gearbox’)	29
Total included after scrutinizing (title> duplicity>abstract >conclusion)		2

Case specific parameter range for each attribute



Vehicle customization through modeling configurator

Figure 40. Left; default model of vehicles and Right; modified models from generic modeling configurator

Modeling Configurator

Modeling Configurator

```
function Thesis
```

```
clc
clear all
f = figure('NumberTitle','off','name','Vehicle Modification
Panel','resize','off','Visible','on','Position',[500,500,1200,800]);
movegui(f,'center')
```

XML file Editing

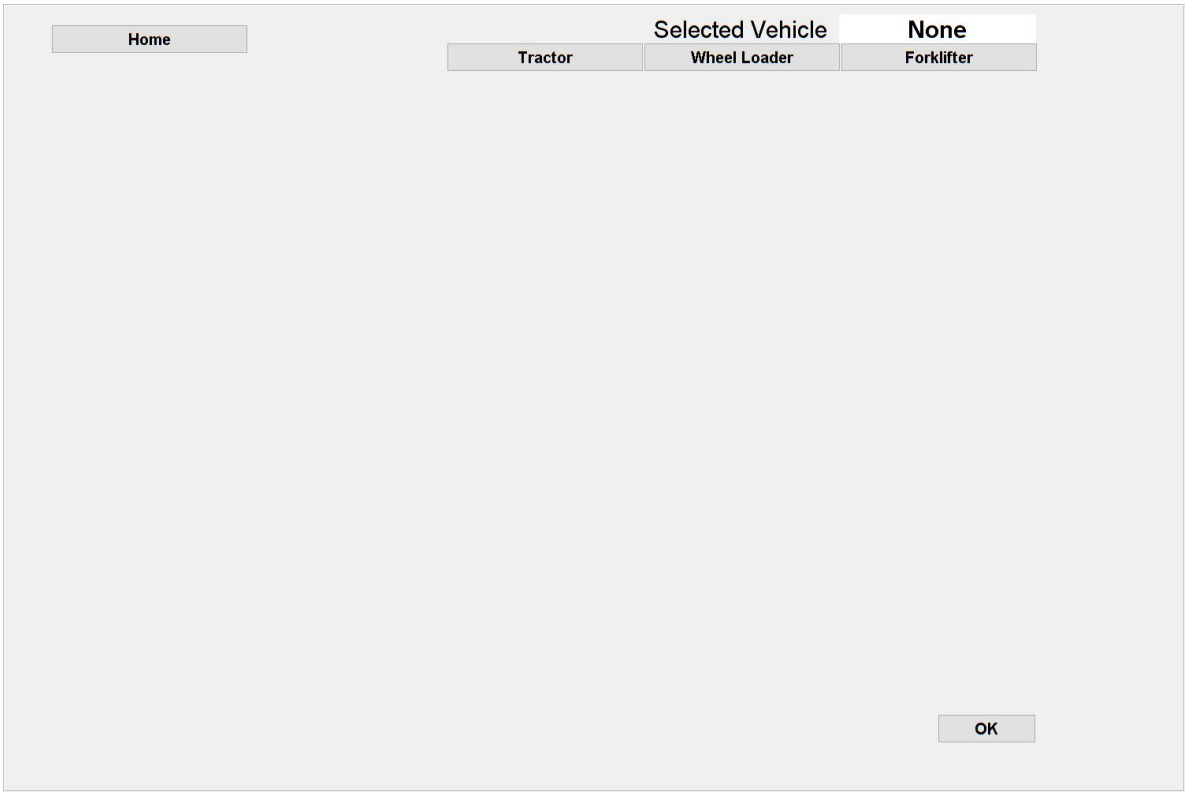
```
xmlfile = fullfile('Tutorial_3_model.xml');
docNode = xmlread(xmlfile);
xmlwrite(docNode);
machine=0;
w=1;P=0;T=0;VSelect=0;frd=0;rrd=0;fwidthmin=0;fwidthmax=0;fheightmax=0;rwidthmi
n=0;rwidthmax=0;rheightmax=0;
fwidth=0;rwidth=0;fheight=0;rheight=0;n=11;fnum=1;rnum=1;fweight=0;rweight=0;ffirst
gear=0;tmaxnum=1;


weight=0;width=0;height=0;heighto=0;RimDia=0;dummy=0;graphics=0;Spline=0;number
=0;
force=0;brake=0;sign=0;Engine=0;gearnumf=6;gearnumr=3;gearbox=0;
```

Buttons Creation

```
all=uibuttongroup('Visible','on');
Tractorb = uicontrol('Parent',all,'Style','pushbutton','BackgroundColor','default',...
    'String','Tractor','FontSize',[12],'FontWeight','bold',...
    'Position',[450,730,200,30],'Callback',@Specs_Callback);
Craneb = uicontrol('Parent',all,'Style','pushbutton',...
    'String','Wheel Loader','FontSize',[12],'FontWeight','bold',...
    'Position',[650,730,200,30],'Callback',@Specs_Callback);
Forklifterb = uicontrol('Parent',all,'Style','pushbutton',...
    'String','Forklifter','FontSize',[12],'FontWeight','bold',...
```

Modeling Configurator

```
    'Position',[850,730,200,30],'Callback',@Specs_Callback);
OK = uicontrol('Style','pushbutton',...
    'String','OK','FontSize',[12],'FontWeight','bold',...
    'Position',[950,50,100,30],'Callback',@OKbutton_Callback);
Home = uicontrol('Style','pushbutton',...
    'String','Home','FontSize',[12],'FontWeight','bold',...
    'Position',[50,750,200,30],'Callback',@Backbutton_Callback );
Select = uicontrol('Style','text','String','Selected Vehicle',...
    'FontSize',[18],'Position',[650,760,200,30]);
mod=uitabgroup('Visible','off','Position',[.05,.1,.9,.8]);
VSelect=uicontrol('Visible','on','Style','text',...
        'String','None','FontSize',[18],'BackgroundColor','White',...
        'FontWeight','bold','Position',[850,760,200,30]);
default=uicontrol('Visible','off','Style','text','BackgroundColor','Red',...
    'String','Please Select a Vehicle?','FontSize',[12],...
    'FontWeight','bold','Position',[500,50,400,30]);
```



Modeling Configurator

Returning Home

```
function Backbutton_Callback(hObject,eventdata,handles)
w=1;
delete(mod);
set(VSelect,'String','None');
all.Visible ='on';
default.Visible='on';
machine=0;
end
```

Vehicle Specification Selection

```
function Specs_Callback( hObject,eventdata,handles)
machine=get(hObject,'String');
set(VSelect,'String',machine);
default.Visible='off';
VSelect.Visible='on';
all.Visible ='off';
   mod=uitabgroup('Visible','on','Position',[.05,.1,.9,.8]);
   engine = uitab(mod,'Title','Engine Modelling');
   ftyre = uitab(mod,'Title','Front Tyre Modelling');
   curve=axes('Visible','off','Parent',engine,'position',[0.35 0.2  0.6 0.7]);
   curve1=axes('Visible','off','Parent',ftyre,'position',[0.35 0.2  0.6 0.7]);
   curve2=axes('Visible','off','Parent',ftyre,'position',[0.35 0.2  0.6 0.7]);

switch machine
   case 'Tractor'
      xmlfile = fullfile('Tractor_Model_real.xml');
      docNode = xmlread(xmlfile);
      xmlwrite(docNode);
      rtyre = uitab(mod,'Title','Rear Tyre Modelling');
      curve3=axes('Visible','off','Parent',rtyre,'position',[0.35 0.2  0.6 0.7]);
```

Modeling Configurator

```
    curve4=axes('Visible','off','Parent',rtyre,'position',[0.35 0.2  0.6 0.7]);
    gear = uitab(mod,'Title','Gear Box Modelling');
    rpmmin=1900; rpmmax=2100;wlim=200;tlimx=1050;tlimy=200;
    pwrmin=120; pwrmax=160;
    fwidthmin=420;fwidthmax=600;
    frd=711;aratiof=65;rrd=965;aratior=60;
    rwidthmin=320; rwidthmax=900;
    fnum=1; rnum=1;ffirstgear=4;gearnumf=4;gearnumr=4;
    FrontAR={'65','70','80','85'};RearAR={'60','65','70','75','80','85','95'};
    tyrelim=2200;tmaxnum=3;


case 'Wheel Loader'
    xmlfile = fullfile('WheelloaderReal.xml');
    docNode = xmlread(xmlfile);
    xmlwrite(docNode);
    set(ftyre,'Title','Tyre Modelling')
    gear = uitab(mod,'Title','Gear Box Modelling');
    rpmmin=1900; rpmmax=2100;wlim=200;tlimx=1250;tlimy=200;
    pwrmin=110; pwrmax=190;
    fwidthmin=525;fwidthmax=675;rwidthmax=fwidthmax;
    frd=635;aratiof=80;rrd=frd;aratior=aratiof;
    fnum=1;ffirstgear=4;gearnumf=4;gearnumr=4;
    FrontAR={'92'};tmaxnum=2;
    tyrelim=1700;
case 'Forklifter'
    xmlfile = fullfile('RoclaFinal.xml');
    docNode = xmlread(xmlfile);wlim=200;tlimx=300;tlimy=0;
    xmlwrite(docNode);
    set(ftyre,'Title','Front Tyre Modelling')
    rpmmin=2200; rpmmax=2500;
    pwrmin=25; pwrmax=55;
    fwidthmin=70;fwidthmax=110;rwidthmax=fwidthmax;
    frd=250;aratiof=100;rrd=frd;aratior=aratiof;
```

Modeling Configurator

```
        FrontAR={'100'};
        tyrelim=500;
end
```

Engine selection

```
%max rpm selection
wo=(rpmmin+rpmmax)/2;
pv=(pwrmax+pwrmin)/2;
rpmtext = uicontrol('Parent',engine,'Style','text',...
    'String','Rated Speed (RPM)','Position',[10,550,100,50]);
rpmnum = uicontrol('Parent',engine,'Style','text',...
    'String',wo,'Position',[110,560,50,15]);
rpmvalue = uicontrol('Parent',engine,'Style','slider',...
    'Min',rpmmin,'Max',rpmmax,'Value',wo,'SliderStep',[((rpmmax-rpmmin)/10)/(rpmmax-
rpmmin) ((rpmmax-rpmmin)/10)/(rpmmax-rpmmin)],...
    'Callback',@rpmvalue_Callback,'Position',[10,560,100,15]);

%max power selection
pwrtext = uicontrol('Parent',engine,'Style','text',...
    'String','Rated Power (kW)','Position',[10,500,100,50]);
pwrnum = uicontrol('Parent',engine,'Style','text',...
    'String',pv,'Position',[110,520,50,15]);
pwrvalue = uicontrol('Parent',engine,'Style','slider',...
    'Min',pwrmin,'Max',pwrmax,'Value',pv,...
    'SliderStep',[((pwrmax-pwrmin)/10)/(pwrmax-pwrmin) ((pwrmax-
pwrmin)/10)/(pwrmax-pwrmin)],...
    'Callback',@pwrvalue_Callback,'Position',[10,520,100,15]);
% Engine selection
```

Front Tyre selection

```
%Width
fwidthtext = uicontrol('Parent',ftyre,'Style','text',...
```

Modeling Configurator

```
    'String','Width (mm)','Position',[-15,540,100,50]);
fwidthnum = uicontrol('Parent',ftyre,'Style','text',...
    'String',(fwidthmin+fwidthmax)/2,'Position',[110,560,50,15]);
fwidthvalue = uicontrol('Parent',ftyre,'Style','slider',...
    'Min',fwidthmin,'Max',fwidthmax,'Value',(fwidthmin+fwidthmax)/2,'SliderStep',[5/(40)
10/(40)],...
    'Callback',@fwidthvalue_Callback,'Position',[10,560,100,15]);
aratiofmenu = uicontrol('Style','popup','Parent',ftyre,'enable','on',...
    'String', FrontAR,'Position', [10,520,100,15],'Callback',@Front_Aspect_Ratio);
fheighttext = uicontrol('Parent',ftyre,'Style','text',...
    'String','Aspect Ratio','Position',[-15,535,100,20]);
fheightnum = uicontrol('Parent',ftyre,'Style','text',...
    'String','','FontSize',[14],'Position',[350,550,50,25]);
heightwar = uicontrol('Visible','off','Style','text',...
    'ForegroundColor','red','String','Model Limit
Exceeded','FontSize',[14],'Position',[520,650,200,25]);
fstandtyre = uicontrol('Parent',ftyre,'Style','text','BackgroundColor','white',...
    'String','','FontSize',[14],'Position',[150,520,300,80]);
```

Number of Front Tyre

```
switch machine
    case 'Forklifter'
    otherwise
    fnumtext = uicontrol('Parent',ftyre,'Style','text',...
        'String','Number of Tyres','Position',[-10,460,150,50]);
    fnumnum = uicontrol('Parent',ftyre,'Style','text',...
        'String',fnum*2,'Position',[110,480,50,15]);
    fnumvalue = uicontrol('Parent',ftyre,'Style','slider',...
        'Min',1,'Max',tmaxnum,'Value',1,'SliderStep',[1/(tmaxnum-1) 1/(tmaxnum-1)],...
        'Callback',@fnumvalue_Callback,'Position',[10,480,100,15]);
end
```

Modeling Configurator

Rear Tyre selection

```
switch machine
    case 'Forklifter'
    case 'Wheel Loader'
    otherwise
%Width
rwidthtext = uicontrol('Parent',rtyre,'Style','text',...
    'String','Width (mm)','Position',[-15,540,100,50]);
rwidthnum = uicontrol('Parent',rtyre,'Style','text',...
    'String',(rwidthmin+rwidthmax)/2,'Position',[110,560,50,15]);
rwidthvalue = uicontrol('Parent',rtyre,'Style','slider',...
    'Min',rwidthmin,'Max',rwidthmax,'Value',(rwidthmin+rwidthmax)/2,'SliderStep',[1/29
2/29],...
    'Callback',@rwidthvalue_Callback,'Position',[10,560,100,15]);
aratiormenu = uicontrol('Style','popup','Parent',rtyre,'enable','on',...
        'String', RearAR,...
        'Position', [10,520,100,15],'Callback',@Rear_Aspect_Ratio);
rheighttext = uicontrol('Parent',rtyre,'Style','text',...
    'String','Aspect Ratio','Position',[-15,535,100,20]);
rheightnum = uicontrol('Parent',rtyre,'Style','text',...
    'String','','FontSize',[14],'Position',[350,550,50,25]);
rheightwar = uicontrol('Parent',rtyre,'Visible','off','Style','text',...
    'ForegroundColor','red','String','Model Limit
Exceeded','FontSize',[14],'Position',[420,550,200,25]);
rstandtyre = uicontrol('Parent',rtyre,'Style','text',...
    'String','','FontSize',[14],'BackgroundColor','white','Position',[150,520,300,80]);
end
```

Number of Rear Tyre

```
switch machine
    case 'Tractor'
```

Modeling Configurator

```
    rnumtext = uicontrol('Parent',rtyre,'Style','text',...
    'String','Number of Tyres','Position',[-10,460,150,50]);
    rnumnum = uicontrol('Parent',rtyre,'Style','text',...
    'String',rnum*2,'Position',[110,480,50,15]);
    rnumvalue = uicontrol('Parent',rtyre,'Style','slider',...
    'Min',1,'Max',3,'Value',1,'SliderStep',[1/2 1/2],...
    'Callback',@rnumvalue_Callback,'Position',[10,480,100,15]);
    otherwise
end
```

Gearbox Modelling

```
switch machine
    case 'Forklifter'
    otherwise
    fwdgeartext = uicontrol('Parent',gear,'Style','text','FontSize',[10],...
    'String','Number of Forward Gears=','Position',[10,550,190,50]);
    fwdgearvalue = uicontrol('Parent',gear,'Style','text','FontSize',[10],...
    'String','6','Position',[210,550,50,50]);
    fwdgearmenu = uicontrol('Style','slider','Parent',gear,...
    'Min',3,'Max',10,'Value',6,'SliderStep',[1/7 1/7],'FontSize',[10],...
    'Position', [20,560,160,15],'Callback',@Fdw_Gears);

    rvrsgeartext = uicontrol('Parent',gear,'Style','text','FontSize',[10],...
    'String','Number of Reverse Gears=','Position',[10,500,190,50]);
    rvrsgearvalue= uicontrol('Parent',gear,'Style','text','FontSize',[10],...
    'String','3','Position',[210,500,50,50]);
    rvrsgearmenu = uicontrol('Style','slider','Parent',gear,...
    'Min',1,'Max',5,'Value',3,'SliderStep',[1/4 1/4],'FontSize',[10],...
    'Position',[20,510,160,15],'Callback',@Rvrs_Gears);
end
function Fdw_Gears(hObject,eventdata,handles)
    gearnumf = round(get(hObject,'Value'));
```

Modeling Configurator

```
    set(fwdgearvalue,'String',gearnumf);


end
function Rvrs_Gears(hObject,eventdata,handles)
    gearnumr = round(get(hObject,'Value'));
    set(rvrsgearvalue,'String',gearnumr);
end
```

Engine Modelling

```
function rpmvalue_Callback(hObject,eventdata,handles)
    wo=round(get(hObject,'Value'),-1);
    set(rpmnum,'String',wo);
    Engine_Curves;
end
function pwrvalue_Callback(hObject,eventdata,handles)
    pv=round(get(hObject,'Value'));
    set(pwrnum,'String',pv);
    Engine_Curves;
end
function Engine_Curves
    p1=(pv)/(wo);
    p2=(pv)/(wo.^2);
    p3=(-1)*(pv)/(wo.^3);
    delete (curve)
    w=linspace(0,rpmmax+wlim,100);
    P=(p1)*(w)+(p2)*(w.^2)+(p3)*(w.^3);
    curve=axes('Parent',engine,'position',[0.3 0.3  0.6 0.6]);
        for i=1:1:100
        P(1,i)=(p1)*(w(1,i))+(p2)*(w(1,i).^2)+(p3)*(w(1,i).^3);
        T(1,i)=9.5488*1000*((p1)+(p2)*(w(1,i))+(p3)*(w(1,i).^2));
        end
    grid on; yyaxis left; plot(curve,w,P);ylabel('Power (kW)');
```

Modeling Configurator

```
    axis([000 rpmmax+wlim 0 pwrmax+30]);
    yyaxis right; plot(curve,w,T); ylabel('Torque (Nm)');axis([800 rpmmax+wlim tlimy
tlimx]);
    title('Power & Torque v/s Speed'); xlabel('Speed(RPM)')
end
```

Front Tyre Modelling

```
function fwidthvalue_Callback(hObject,eventdata,handles)
    fwidth=round(get(hObject,'Value'));
    set(fwidthnum,'String',fwidth);
    Front_Tire_Calculation;
    Front_Tire_Plot;
    Tire_Warning;
end
function Front_Aspect_Ratio(hObject,event,handles)
    fwidth=str2num(get(fwidthnum,'String'));
    str = get(hObject, 'String');
    val = get(hObject,'Value');
    aratiof=str2num(str{val});
    Front_Tire_Calculation;
    Front_Tire_Plot;
    Tire_Warning;
end
function Front_Tire_Calculation

fweight=(2.47*0.03*((fwidth/25.4)*aratiof/100)*(fwidth/25.4)*(((fwidth/25.4)*aratiof/100
)+(frd/25.4)))/(2.2);
    fheight=frd+2*(fwidth*aratiof/100);
    fheightmax=frd+2*(fwidthmax*85/100);
    rheightmax=rrd+2*(rwidthmax*100/100);
                x=['Standard Tyre Size = ',num2str(fwidth),'/',num2str(aratiof),'R',...
       num2str(round(frd/25.4)),'   ','Overall Diameter    = ',...
```

Modeling Configurator

```
        num2str(fheight),'mm       ','Weight of Tyre  = ',num2str(round(fweight)),'kg      ' ];
    switch machine
        case'Forklifter'
      Inertia=docNode.getElementsByTagName('Dummy_FR').item(0);

fweight=str2num(Inertia.getElementsByTagName('Inertia').item(0).getAttribute('mass'));
    x=['Standard Tyre Size =
',num2str(round(fheight/25.4)),'*',num2str(round(fwidth/25.4)),'*',...
        num2str(round(frd/25.4)),'   ','Overall Diameter    = ',...
        num2str(round(fheight)),'in       ','Weight of Tyre  = ',num2str(round(fweight)),'kg      '
];
%     case'Forklifter'
    end

    set(fstandtyre,'String',x);
    height=fheight;
end
function Tire_Warning
    if height>=tyrelim
        for i=1:1:2
        set(heightwar,'Visible','On');
        pause(0.2);
        set(heightwar,'Visible','Off');
        pause(0.1);
        end
    end
end
function Front_Tire_Plot
    delete (curve1);delete (curve2);
    % x=fwidth/600
    % y=fheight/1492
    curve1=axes('Parent',ftyre,'position',[0.25 0.3  0.15 0.5]);
    rectangle('Position',[0 0 fwidth fheight],'FaceColor',[.3 .3 .3],'Curvature',[0.2,1])% Front
```

Modeling Configurator

```
View
   rectangle('Position',[(2*fwidth/9) 0 fwidth/9 fheight],'FaceColor',[.4 .4 .4],'Curvature',[0
1])% Grooves
   rectangle('Position',[(6*fwidth/9) 0 fwidth/9 fheight],'FaceColor',[.4 .4 .4],'Curvature',[0
1])% Grooves
   title('Front View');axis([0 rwidthmax 0 rheightmax]);
   ylabel('Diameter');xlabel('Width');grid on;
   curve2=axes('Parent',ftyre,'position',[0.65 0.3  0.3 0.5]);
   rectangle('Position',[0 0 fheight fheight],'FaceColor',[.3 .3 .3],'Curvature',[1 1])%Side
View
   rectangle('Position',[(fheight-frd)/2 (fheight-frd)/2 frd frd],'FaceColor',[0.6 .6
0.6],'Curvature',[1 1])%Rim
   rectangle('Position',[(fheight-fheightmax/5)/2 (fheight-fheightmax/5)/2 fheightmax/5
fheightmax/5],'FaceColor',[1 1 1],'Curvature',[1 1])%Rim hole

   rectangle('Position',[(fheight-fheightmax/4)/2 (fheight-fheightmax/4)/2 fheightmax/25
fheightmax/25],'FaceColor',[0 0 0],'Curvature',[1 1])%Bolts
   rectangle('Position',[(fheight+fheightmax/5)/2 (fheight-fheightmax/4)/2 fheightmax/25
fheightmax/25],'FaceColor',[0 0 0],'Curvature',[1 1])%Bolts
   rectangle('Position',[(fheight-fheightmax/4)/2 (fheight+fheightmax/5)/2 fheightmax/25
fheightmax/25],'FaceColor',[0 0 0],'Curvature',[1 1])%Bolts
   rectangle('Position',[(fheight+fheightmax/5)/2 (fheight+fheightmax/5)/2 fheightmax/25
fheightmax/25],'FaceColor',[0 0 0],'Curvature',[1 1])%Bolts
   title('Side View');axis([0 rheightmax 0 rheightmax])
   ylabel('Diameter');xlabel('Diameter');grid on;
   % axes(curve1);
   % imshow('Fronttirefront.jpg');
end
function fnumvalue_Callback(hObject,eventdata,handles)
   fnum=round(get(hObject,'Value'));
   set(fnumnum,'String',fnum*2);
end
```

Modeling Configurator

Rear Tyre Modelling

```
function rwidthvalue_Callback(hObject,eventdata,handles)
    rwidth=round(get(hObject,'Value'));
    set(rwidthnum,'String',rwidth);
    Rear_Tire_Calculation;
    Rear_Tire_Plot;
    Tire_Warning;

end
function Rear_Aspect_Ratio(hObject,event,handles)
    rwidth=str2num(get(rwidthnum,'String'));
    str = get(hObject, 'String');
    val = get(hObject,'Value');
    aration=str2num(str{val});
    Rear_Tire_Calculation;
    Rear_Tire_Plot;
    Tire_Warning;
end
function Rear_Tire_Calculation

rweight=(2.47*0.03*((rwidth/25.4)*aration/100)*(rwidth/25.4)*(((rwidth/25.4)*aration/100
)+(rrd/25.4)))/2;
    rheight=rrd+2*(rwidth*aration/100);
    rheightmax=rrd+2*(rwidthmax*95/100);
    y=['Standard Tyre Size = ',num2str(rwidth),'/',num2str(aration),'R',...
        num2str(round(rrd/25.4)),'   ','Overall Diameter    = ',...
        num2str(rheight),'mm       ','Weight of Tyre  = ',num2str(round(rweight)),'kg      ' ];
    set(rstandtyre,'String',y);
    height=rheight;
end
```

Modeling Configurator

```
function Rear_Tire_Plot
    delete (curve3);delete (curve4);
    curve3=axes('Parent',rtyre,'position',[0.25 0.3  0.15 0.5]);
    rectangle('Position',[0 0 rwidth rheight],'FaceColor',[.3 .3 .3],'Curvature',[0.3,1])%
Grooves
    rectangle('Position',[(2*rwidth/9) 0 rwidth/9 rheight],'FaceColor',[.4 .4 .4],'Curvature',[0
1])% Grooves
    rectangle('Position',[(6*rwidth/9) 0 rwidth/9 rheight],'FaceColor',[.4 .4 .4],'Curvature',[0
1])% Grooves
    title('Front View');axis([0 rwidthmax 0 rheightmax])
    ylabel('Diameter');xlabel('Width');grid on;
    curve4=axes('Parent',rtyre,'position',[0.65 0.3  0.3 0.5]);
    rectangle('Position',[0 0 rheight rheight],'FaceColor',[.3 .3 .3],'Curvature',[1 1])%Side
View
    rectangle('Position',[(rheight-rrd)/2 (rheight-rrd)/2 rrd rrd],'FaceColor',[0.6 .6
0.6],'Curvature',[1 1])%Rim
    rectangle('Position',[(rheight-fheightmax/5)/2 (rheight-fheightmax/5)/2 fheightmax/5
fheightmax/5],'FaceColor',[1 1 1],'Curvature',[1 1])%Rim hole

    rectangle('Position',[(rheight-rheightmax/5)/2 (rheight-rheightmax/4)/2 rheightmax/30
rheightmax/30],'FaceColor',[0 0 0],'Curvature',[1 1])%Bolts
    rectangle('Position',[(rheight+rheightmax/7)/2 (rheight-rheightmax/4)/2 rheightmax/30
rheightmax/30],'FaceColor',[0 0 0],'Curvature',[1 1])%Bolts
    rectangle('Position',[(rheight-rheightmax/5)/2 (rheight+rheightmax/7)/2 rheightmax/30
rheightmax/30],'FaceColor',[0 0 0],'Curvature',[1 1])%Bolts
    rectangle('Position',[(rheight+rheightmax/7)/2 (rheight+rheightmax/7)/2 rheightmax/30
rheightmax/30],'FaceColor',[0 0 0],'Curvature',[1 1])%Bolts
    title('Side View');axis([0 rheightmax 0 rheightmax]);
    ylabel('Diameter');xlabel('Diameter');grid on;
end
function rnumvalue_Callback(hObject,eventdata,handles)
    rnum=round(get(hObject,'Value'));
```

Modeling Configurator

```
    set(rnumnum,'String',rnum*2);
end
```

```
end
```

OK Button

```
function OKbutton_Callback(hObject,eventdata,handles)
    if machine==0
    default.Visible='on';
    else
switch machine
```

Tractor XML File editing

```
    case 'Tractor'
```

```
%Engine Editing
Engine='Spline_Engine-3';
Edit_Engine_XML;

%Tyre Editing
brake = 'Tyre_FrontRight;Tyre_FrontLeft;Tyre_RearRight;Tyre_RearLeft;';
if fheight==0
else
    %Front Right tire editing

weight=fweight;width=fwidth;height=fheight;heighto=2*746;RimDia=frd;dummy='Dum
my_FrontTyreR';sign=1;

graphics='Graphics_FrontTyreR';Spline='Spline_FrontTyres';number=fnum;force='Tyre_F
rontRight';
% Gearbox Editing
gearbox='Gearbox';
Edit_Gear_XML;
```

Modeling Configurator

```
    Edit_Tyre_XML;
    %Front Left tire editing

weight=fweight;width=fwidth;height=fheight;heighto=2*746;RimDia=frd;dummy='Dum
my_FrontTyreL';sign=-1;

graphics='Graphics_FrontTyreL';Spline='Spline_FrontTyres';number=fnum;force='Tyre_F
rontLeft';
    Edit_Tyre_XML;
    Edit_Steering_XML
end
if rheight==0
else
    %Rear Right tire editing

weight=rweight;width=rwidth;height=rheight;heighto=2*980;RimDia=rrd;dummy='Dum
my_RearTyreR';sign=1;

graphics='Graphics_RearTyreR';Spline='Spline_RearTyres';number=rnum;force='Tyre_Re
arRight';
    Edit_Tyre_XML;
    %Rear Left tire editing

weight=rweight;width=rwidth;height=rheight;heighto=2*980;RimDia=rrd;dummy='Dum
my_RearTyreL';sign=-1;

graphics='Graphics_RearTyreL';Spline='Spline_RearTyres';number=rnum;force='Tyre_Re
arLeft';
    Edit_Tyre_XML;
end
% brake update
xtyre=docNode.getElementsByTagName('Brake_Input').item(0);
xtyre.setAttribute('PrimName',char(brake))
```

Modeling Configurator

```
xmlwrite('Tractor_Model.xml',docNode);
winopen('Tractor_Model.xml')
winopen('Tractor_Model.mvs')
```

Wheel Loader XML File editing

```
case 'Wheel Loader'
```

```
%Engine Editing
Engine='Spline_Volvo_D6E_LBE3';
Edit_Engine_XML;

%Tyre Editing
rheight=fheight;
brake = 'Tyre_FrontR;Tyre_FrontL;Tyre_RearR;Tyre_RearL;';
if fheight==0
else
    %Front Right tire editing

weight=fweight;width=fwidth;height=fheight;heighto=2*792;RimDia=frd;dummy='Dum
my_TyreF_R';sign=1;

graphics='Graphics_TyreFRight';Spline='Spline_Tyre';number=fnum;force='Tyre_FrontR';
    % Gearbox Editing
    gearbox='GearBox';
    Edit_Gear_XML;
    Edit_Tyre_XML;
    %Front Left tire editing

weight=fweight;width=fwidth;height=fheight;heighto=2*792;RimDia=frd;dummy='Dum
my_TyreF_L';sign=-1;

graphics='Graphics_TyreFLeft';Spline='Spline_Tyre';number=fnum;force='Tyre_FrontL';
    Edit_Tyre_XML;
```

Modeling Configurator

```
end
if rheight==0
else
   %Rear Right tire editing

weight=fweight;width=fwidth;height=fheight;heighto=2*792;RimDia=frd;dummy='Dum
my_TyreR_R';sign=1;

graphics='Graphics_TyreRRight';Spline='Spline_Tyre';number=fnum;force='Tyre_RearR';
   Edit_Tyre_XML;
   %Rear Left tire editing

weight=fweight;width=fwidth;height=fheight;heighto=2*792;RimDia=frd;dummy='Dum
my_TyreR_L';sign=-1;

graphics='Graphics_TyreRLeft';Spline='Spline_Tyre';number=fnum;force='Tyre_RearL';
   Edit_Tyre_XML;
end
xtyre=docNode.getElementsByTagName('Brake_Input').item(0);
xtyre.setAttribute('PrimName',char(brake))
      xmlwrite('Tutorial_3.xml',docNode);
      winopen('Tutorial_3.xml')
      winopen('WheelLoader.mvs')
```

ForkLifter XML File editing

```
   case 'Forklifter'
%Engine Editing
Engine='SplineMotorFront';
Edit_Engine_XML;
% rheight=fheight;
%Tyre Editing
brake = 'Tyre_FR;Tyre_FL;Tyre_R;';
```

Modeling Configurator

```
if fheight==0
else
   tyre=docNode.getElementsByTagName('SplineFrontTyre').item(0);
   tyre.removeAttribute('File');
   tyrex=docNode.createElement('x');
   docNode.getElementsByTagName('SplineFrontTyre').item(0).appendChild(tyrex);
   tyrey=docNode.createElement('y');
   docNode.getElementsByTagName('SplineFrontTyre').item(0).appendChild(tyrey);
%Front Right tire editing

weight=fweight;width=fwidth;height=fheight;heighto=428;RimDia=frd;dummy='Dummy
_FR';sign=1;

graphics='Graphics_FR';Spline='SplineFrontTyre';number=fnum;force='Tyre_FR';gearbox
='Rocla_GearBoxFR';

ffirstgear=str2num(docNode.getElementsByTagName(char(gearbox)).item(0).getAttribute
('ifw0'));

   Edit_Tyre_XML;

%Front Left tire editing

weight=fweight;width=fwidth;height=fheight;heighto=428;RimDia=frd;dummy='Dummy
_FL';sign=-1;

graphics='Graphics_FL';Spline='SplineFrontTyre';number=fnum;force='Tyre_FL';gearbox
='Rocla_GearBoxFL';
   Edit_Tyre_XML;
end
      xmlwrite('RoclaFinal00.xml',docNode);
      winopen('RoclaFinal00.xml');
```

Modeling Configurator

```
        winopen('RoclaFinal.mvs');
    otherwise
```

```
        xmlwrite('try.xml',docNode);
end % switch simulation file

% close
    end
w=1;
delete(mod);
set(VSelect,'String','None');
all.Visible ='on';
default.Visible='on';
machine=0;
end% Simulation File Preparation
```

Generic XML File Editing

```
function Edit_Engine_XML
    if w(1,1)==0
    w=w*(2*3.14)/60;
    engine=docNode.getElementsByTagName(char(Engine)).item(0);

    for i=1:1:12
        engine.setAttribute('NPoints',num2str(i));
        x = sym('x', [1 12]);
        y = sym('y', [1 12]);
        engine.getElementsByTagName('x').item(0).setAttribute(char(x(i)),num2str(w(1,(i-
1)*9+1)));
        engine.getElementsByTagName('y').item(0).setAttribute(char(y(i)),num2str(T(1,(i-
1)*9+1)));
    end
    end
%
```

Modeling Configurator

```
docNode.getElementsByTagName('Volvo_D6E_LBE3').item(0).setAttribute('MbrMax','3e
4');
end
function Edit_Gear_XML
% Front gear XML editing
ffirstgear=str2num(docNode.getElementsByTagName(char(gearbox)).item(0).getAttribute
('ifw0'));
% ffirstgear=(ffirstgear)/(heighto/height).^2;

docNode.getElementsByTagName(char(gearbox)).item(0).setAttribute('ngearf',num2str(ge
arnumf));
   nminus1=ffirstgear;
   for i=1:1:10
   fgear = sym('ifw', [1 10]);
      if i>gearnumf

docNode.getElementsByTagName(char(gearbox)).item(0).removeAttribute(char(fgear(i-
1)));
      else
      Cg=(ffirstgear/0.73).^(1/(gearnumf-1));
      fgearvalue=nminus1/Cg;
      nminus1=fgearvalue;

docNode.getElementsByTagName(char(gearbox)).item(0).setAttribute(char(fgear(i)),num2
str(fgearvalue));
      end
   end
% Rear gear XML editing

docNode.getElementsByTagName(char(gearbox)).item(0).setAttribute('ngearr',num2str(ge
arnumr));
   nminus1=ffirstgear;
   for i=1:1:10
```

Modeling Configurator

```
   rgear = sym('ibw', [1 10]);
      if i>gearnumr

docNode.getElementsByTagName(char(gearbox)).item(0).removeAttribute(char(rgear(i-
1)));
      else
      Cg=(ffirstgear/2.05).^(1/(gearnumr-1));
      rgearvalue=nminus1/Cg;
      nminus1=rgearvalue;

docNode.getElementsByTagName(char(gearbox)).item(0).setAttribute(char(rgear(i)),num2
str(rgearvalue));
      end
   end

docNode.getElementsByTagName(char(gearbox)).item(0).setAttribute('ifw0',num2str((ffir
stgear)/(heighto/height).^2));

docNode.getElementsByTagName(char(gearbox)).item(0).setAttribute('ibw0',num2str((ffir
stgear)/(heighto/height).^2));

end

function Edit_Steering_XML
      docNode.getElementsByTagName('Steering').item(0).setAttribute('C','5e4');
      docNode.getElementsByTagName('Steering').item(0).setAttribute('K','5e4');
end
function Edit_Tyre_XML
%Tyre Physical Properties
Izz=(1/2)*(weight)*((height/2000).^2+(RimDia/2000).^2);
Ixx=(1/12)*(weight)*(3*(height/2000).^2+3*(RimDia/2000).^2+(width/1000).^2);
Iyy=Ixx;
Inertia=docNode.getElementsByTagName(char(dummy)).item(0);
```

Modeling Configurator

```
Inertia.getElementsByTagName('Inertia').item(0).setAttribute('mass',num2str(weight));
Inertia.getElementsByTagName('I').item(0).setAttribute('Izz',num2str(Izz));
Inertia.getElementsByTagName('I').item(0).setAttribute('Iyy',num2str(Iyy));
Inertia.getElementsByTagName('I').item(0).setAttribute('Ixx',num2str(Ixx));

%Tyre Visual Properties
docNode.getElementsByTagName(char(graphics)).item(0).setAttribute('Scale',num2str(hei
ght/heighto));
tyre=docNode.getElementsByTagName(char(Spline)).item(0);
tyre.setAttribute('NPoints',num2str(number*3));

                for i=1:1:number*3
      y = sym('y', [1 number*3]);
      x = sym('x', [1 number*3]);

tyre.getElementsByTagName('y').item(0).setAttribute(char(y(i)),num2str(height/2000));
      tyre.getElementsByTagName('x').item(0).setAttribute(char(x(i)),num2str((-
width/2000)+((i-1)*(width/2000))));
   end
   xtyre=docNode.getElementsByTagName(char(force)).item(0);
      xtyre.getElementsByTagName('FrictionLat').item(0).setAttribute('s1','5e3');
      xtyre.getElementsByTagName('FrictionMb').item(0).setAttribute('s1','5e3');
   if number>=2
      for i=1:1:number-1;
      tyred = sym(char(dummy), [1 number-1]);
      tyreg = sym(char(graphics), [1 number-1]);
      tyref = sym(char(force), [1 number-1]);
      brake= [brake,char(tyref(i)),';'];

      xtyre=docNode.getElementsByTagName(char(dummy)).item(0).cloneNode(1);
      docNode.getElementsByTagName('Dummies').item(0).appendChild(xtyre);
      docNode.renameNode(docNode.getElementsByTagName(char(dummy)).item(0),'',
char(tyred(i)));
```

Modeling Configurator

```
        oldtyre=docNode.getElementsByTagName(char(tyred(i))).item(0);
        oldtyre.setAttribute('VisualizationGraphics',char(tyreg(i)));
        oldtyre.setAttribute('RelTo',char(tyref(i)));

oldtyre.getElementsByTagName('Position').item(0).setAttribute('z',num2str(((i*width*sign
)+(i*80*sign*((heighto/height).^2)))/1000));

        xtyre=docNode.getElementsByTagName(char(graphics)).item(0).cloneNode(1);
        docNode.getElementsByTagName('Graphics').item(0).appendChild(xtyre);
        docNode.renameNode(docNode.getElementsByTagName(char(graphics)).item(0),'',
char(tyreg(i)));
        oldtyre=docNode.getElementsByTagName(char(tyreg(i))).item(0);
        oldtyre.setAttribute('Body',char(tyred(i)));

        xtyre=docNode.getElementsByTagName(char(force)).item(0).cloneNode(1);
        docNode.getElementsByTagName('Tyre').item(0).appendChild(xtyre);
        docNode.renameNode(docNode.getElementsByTagName(char(force)).item(0),'',
char(tyref(i)));

        end

    end
end
```

```
end %function Thesis
```

Published with MATLAB® R2017a

Comparison of engine models

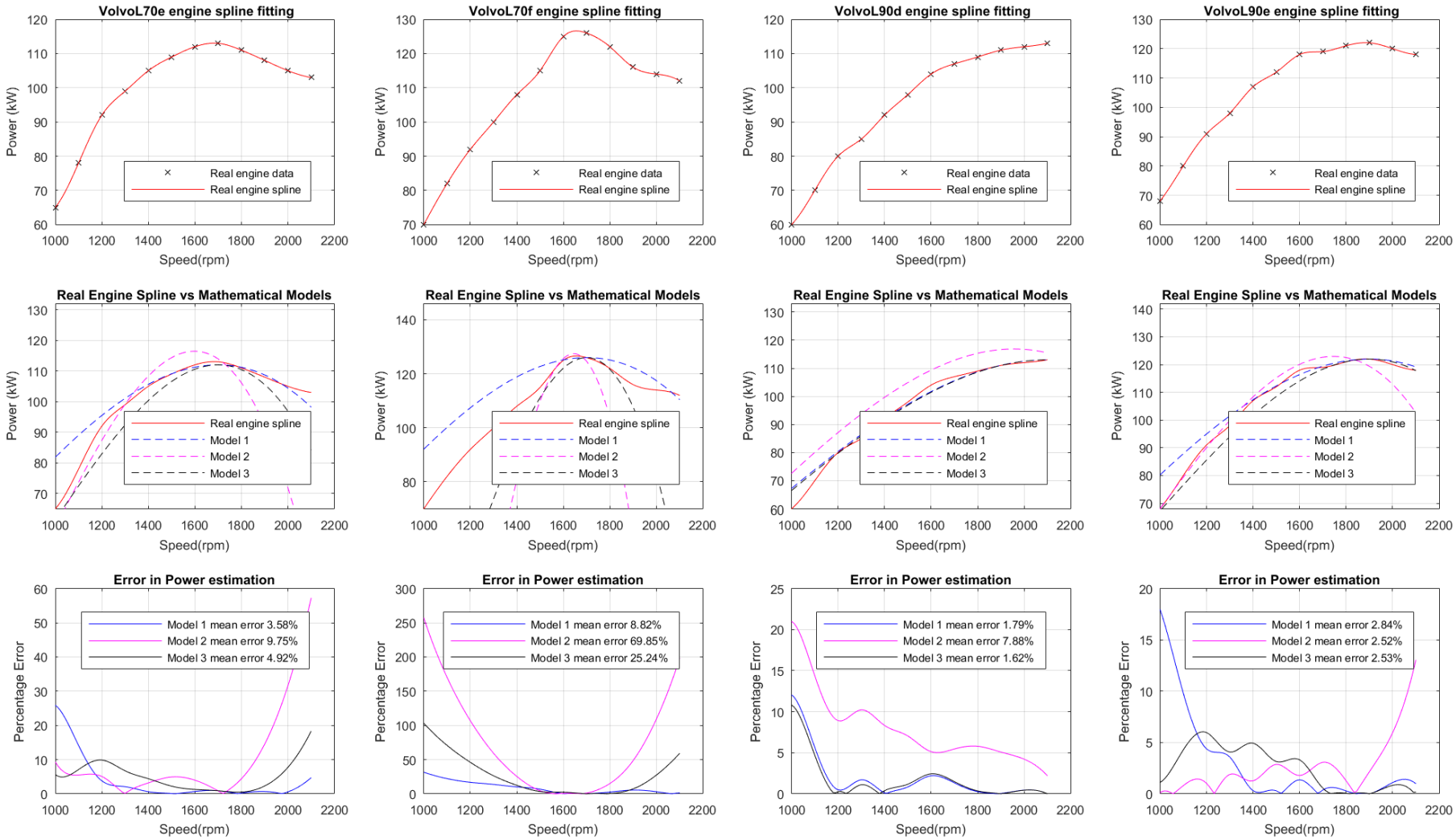


Figure 41 Evaluation of real engines 1-4.

Comparison of engine models

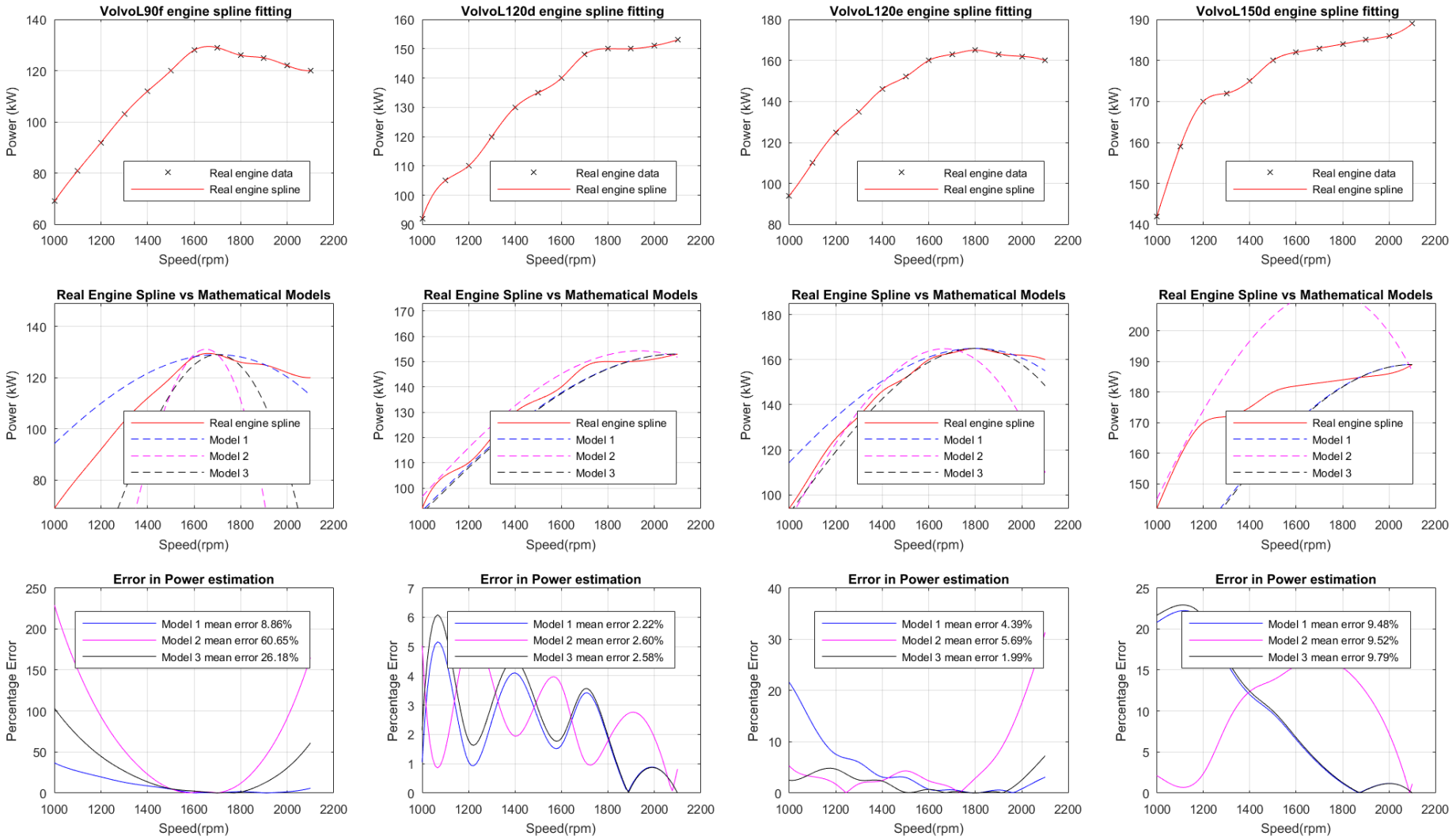


Figure 42 Evaluation of real engines 5-8