YRKESHÖGSKOLAN NOVIA

Operating a Single-Cylinder Engine with a Multi-Cylinder Engine 1D-Model in the Loop with Real-Time Combustion Feedback

André Ahlskog

Master's Thesis

Master's degree programme in Automation Technology

Vaasa 2023

DEGREE THESIS

Author: André Ahlskog

Degree Programme and place of study: Automation Technology, Vaasa

Supervisor(s): Ray Pörn

Title: Operating a Single-Cylinder Engine with a Multi-Cylinder Engine 1D-Model in the Loop with Real-Time Combustion Feedback

---

Date: 4.5.2023 Number of pages: 68

Abstract

With increasingly stringent emission regulations, greenhouse gas reduction targets and growing demand for more sustainable energy, the development of internal combustion engine (ICE) technology has become more critical than ever. Short development cycles and rapid introduction of carbon-free fuels such as hydrogen and ammonia, in combination with complex combustion systems, further increases the need to improve and accelerate the development of ICE. Single-cylinder engine (SCE) platforms and real-time modelling can be effective tools to meet this demand. This thesis has aimed to combine a real-time capable engine model with an SCE platform to enable the SCE to run more according to real multi-cylinder engine (MCE) conditions.

A literature review of real-time modelling within the ICE industry has been made to achieve the goals set up within this work. The main focus was on applications utilising Gamma technologies GT-SUITE simulation software, especially fast-running models (FRM) and GT-xRT. For the SCE and model coupling, an already fast-running MCE model built up in GT power was further modified and converted into an even faster xRT-model. The xRT-model has then been implemented into an already existing Simulink environment. To achieve real-time combustion feedback, a Speedgoat/Simulink Real-Time system was utilised, with measured cylinder pressure from the SCE sent from Speedgoat over User Datagram Protocol (UDP) to the Simulink environment.

The outcome of this thesis is a real-time capable MCE model with combustion feedback from the SCE, in a sense creating a model/hardware-in-the-loop (MiL/HiL) system. The simulated MCE can model real engine dynamics missing from an SCE research platform, such as turbocharger (TC) system and engine-driven auxiliaries. With combustion feedback to the model, it can also automatically simulate and calculate quantities that cannot be measured, e.g., in-cylinder conditions.

Language: English

Key Words: simulation, combustion engine, real-time, hardware in the loop

Acknowledgements

I want to express my heartfelt gratitude to everyone who contributed to completing my Master's thesis.

First and foremost, I am most thankful to my supervisors, Jari Hyvönen and Ray Pörn, who have shown remarkable patience and support throughout the process. Your guidance, insights, and valuable feedback have been instrumental in shaping my thesis direction and bringing it to completion.

I would also like to thank my colleagues and team members, especially Ben Smulter, Martin Nylund, and Tom Kaas, who have actively realised the project described in this thesis. Your contributions, ideas, and hard work have been invaluable, and I am grateful for the collaborative spirit that we have shared.

Furthermore, I want to acknowledge the support provided by Daniel Schimmel at Gamma Technologies, whose excellent assistance has been critical to accomplishing my work. Your dedication and technical expertise have contributed significantly to the quality of my thesis, and I appreciate all the resources and tools you have provided.

Finally, I must thank my family and wife for their love, patience, and unwavering support throughout this journey. Your encouragement and understanding have been a constant source of strength, and I am grateful for the sacrifices you have made to help me achieve my goals.

Once again, I extend my sincere gratitude to all those who have helped me in any way throughout the completion of this thesis.

List of abbreviations

BR	Burn Rate
DTC	Diagnostic Trouble Code
EATS	Emission After Treatment System
ECS	Engine Control System
ECU	Engine Control Unit
FPGA	Field Programmable Gate Array
FRM	Fast Running Model
GHG	Greenhouse Gases
GSP	Global Simulation Platform
HiL	Hardware-in-the-Loop
HRR	Heat Release Rate
ICE	Internal Combustion Engine
MBD	Model-Based Design
MCE	Multi-Cylinder Engine
MiL	Model-in-the-Loop
MVC	Mean Value Cylinder
MVE	Mean Value Engine
NEDC	New European Driving Cycle
NN	Neural Network
RCP	Rapid Control Prototyping
RDE	Real driving emissions
RT	Real-Time
SCE	Single Cylinder Engine
SiL	Software-in-the-loop
TC	Turbocharger
TDC	Top Dead Centre
UDP	User Datagram Protocol
VTAB	Virtual Truck and Bus

Table of Figures

Figure 1: Typical single-cylinder experimental layout [6]. 2
Figure 2: Turbine temp and TC-efficiency vs engine power. The relative difference compared to MCE measured values. 2
Figure 3: Engine model example in GT-ISE [7]. 5
Figure 4: Typical mean value model workflow [12]. 7
Figure 5: Detailed model conversion into FRM by simplifying volumes and pipes. 8
Figure 6: Comparison between Mean Value Model and FRM [8]. 9
Figure 7: "In-the-loop" testing steps with GT-model as a plant. Adapted from [19]. 10
Figure 8: Model-in-the-Loop schematic [22]. 11
Figure 9: An example of a HiL simulation setup [25]. 12
Figure 10: The HiL test bench at AGCO Fendt [26]. 12
Figure 11: HiMSEN HiL system layout [9]. 13
Figure 12: FRM to HiL/RCP interface in Simulink [10]. 15
Figure 13: Overview of GSP and Simulink model, adapted from [27]. 16
Figure 14: Co-simulation of engine model and ECU on a HiL platform [33]. 17
Figure 15: System structure including different models and control units [34]. 18
Figure 16: Direct FRM model creation versus traditional approach [28]. 19
Figure 17: Virtual calibration results [37]. 21
Figure 18: A modelling approach for the end-to-end simulation process [30]. 21
Figure 19: Digital twin setup in VEBIC [31]. 22
Figure 20: Simulink model of engine and electrical equipment [39]. 23
Figure 21: Cycle-mean value model [42]. 24
Figure 22: Digital twin diagnostic process [32]. 25
Figure 23: Layout of the transient SCE HIL system [43]. 26
Figure 24: Example of Simulink Real-Time and Speedgoat testing environment [46]. 27
Figure 25: Example of RCP workflow [50]. 28
Figure 26: Comparison between UDP and TCP [53]. 29
Figure 27: A burn rate added as a combustion object in the cylinder template. 31
Figure 28: Example of a combustion rate vs crank angle. 31
Figure 29: SI Wiebe combustion model template. 32
Figure 30: SI Wiebe example, estimated burn rate in red and measured burn rate in black. 32
Figure 31: Steady-state method (start/stop). 33
Figure 32: Sequential cylinder pressure analysis model. 34
Figure 33: Combustion profile actuation. 35
Figure 34: Measured cylinder pressure to imposed HRR workflow. 35
Figure 35: Example of RLTDependence object: Valve to seat heat transfer coefficient as a function of cylinder pressure. 37
Figure 36: Example of actuated heat transfer multiplier using signals instead of RLTDependence. 37
Figure 37: Cylinder copy. 38
Figure 38: MathEquation template setup for real-time modelling. 39
Figure 39: C compiler options for MathEquation. 39
Figure 40: Turbocharging system-related RLTs in the FRM. 40
Figure 41: Turbocharging system-related signals and calculations in the xRT-model. 40
Figure 42: xRT model instability with large time steps (0.5 ms). 41
Figure 43: Flow stability multiplier. 42
Figure 44: xRT model stability improved with the flow stability multiplier. 42
Figure 45: “UDP send” example. 44

Figure 46: “UDP Receive”, vector build up and heat release calculation. 44
Figure 47: "Build_pcyl_vector": Function overview. 45
Figure 48: "Build_pcyl_vector": Vector packet and cycle selectors. 45
Figure 49: "Build_pcyl_vector": Removing extra elements, sending vector to and triggering HRR calc once per cycle. 46
Figure 50: Simulink heat release calculation overview. 47
Figure 51: Heat release calculation subroutine. 48
Figure 52: Heat release to GT: an overview. 49
Figure 53: Dynamic lookup table. 49
Figure 54: Reset subroutine. 49
Figure 55: Real-time HR GT model overview. 50
Figure 56: Overview of the SCE - to - MCE model system. 50
Figure 57: Left: Sending HR profile to all cylinders separately. Right: Sending HR profile only to lead cylinder. 51
Figure 58: Different colours indicate different time steps. Green elements are 0.1 s (~once per cycle), and red elements are the same as GT 0.1- 0.2 ms (0.6-1.2° CA). 51
Figure 59: Factor of real-time for various models. 52
Figure 60: Combustion profile vs. timestep. Top: Cumulative Heat Release from SCE. Bottom: Crank angle from GT. 53
Figure 61: Burn rate- and Heat release rate-based cylinder pressure. 54

Contents

Acknowledgements II
List of abbreviations III
Table of Figures IV
1 Introduction 1
1.1 Background and problem description 1
1.2 Scope and Limitations 3
1.3 Disposition 3
2 Theoretical framework 5
2.1 GT-SUITE 5
2.1.1 Mean value model 6
2.1.2 Fast running model (FRM) 7
2.1.3 xRT 9
2.2 In-the-Loop Testing 10
2.3 Real-time modelling within the internal combustion engine Industry 12
2.4 Simulink Real-Time and Speedgoat 26
2.5 User Datagram Protocol (UDP) 28
3 Method 30
3.1 Combustion feedback and modelling 30
3.1.1 Impose reference burn rates 31
3.1.2 Impose Spark-Ignition (SI) Wiebe parameters 32
3.1.3 Impose measured heat release rate for each point / Steady-state 33
3.1.4 Sequential cylinder pressure analysis model 34
3.1.5 Impose instant cumulative burned fuel fraction 35
3.2 GT xRT model conversion and comparison to baseline 36
3.3 Simulink Real-Time & Speedgoat + Simulink & GT-POWER co-simulation 42
3.3.1 Real-time cylinder pressure vector via UDP 43
3.3.2 Heat release calculation 46
3.3.3 Instant cumulative heat release to GT 48
4 Results 50
4.1 Real-time capability 50
4.2 Accuracy vs timestep 52
4.3 Heat release rate vs Burn rate 53
5 Discussion and conclusions 55
7 References 57

1 Introduction

This thesis was written for the Research and Technology Development department of Wärtsilä. Wärtsilä is a company that offers comprehensive solutions throughout the entire lifecycle for the marine and energy sectors, using internal combustion engines (ICE) as the primary power source. This chapter aims to explain the reasoning behind this thesis. It begins with a brief overview of the background and problem at hand, then goes on to outline the main purpose and goal of the thesis. The introduction also includes the scope and limitations of the thesis and ends with a disposition.

1.1 Background and problem description

To address climate change and mitigate its adverse impacts, greenhouse gas emissions (GHG) need to be reduced. Climate change is already causing many problems, including heatwaves, droughts, and storms in an ever-increasing frequency and severity, as well as rising sea levels and coastal flooding [1]. These impacts can have severe consequences for human health, the environment, and the economy, and they are likely to worsen if GHG emissions are not decreased. According to the United Nations Framework Convention on Climate Change (UNFCCC) [2], global GHG emissions reached an all-time high in 2018 and are still increasing. To address this situation and reach the goals of the Paris Agreement, it will be necessary to implement policies and measures to reduce GHG emissions and transition to a low-carbon, sustainable energy system. This will require significant global, regional, and local efforts and involve the development and deployment of advanced technologies, as well as changes in how we produce and consume energy.

To achieve the goals set out in the Paris Agreement, which includes limiting global temperature increase to below two °C above pre-industrial levels, countries, regions, and other organisations have set GHG reduction targets to address climate change and shift to a carbon-free, sustainable energy system. One of these targets is the EU's: Fit for 55, which aims to reduce EU emissions by at least 55 % by 2030 and target climate neutrality by 2050 [3]. Another target is the International Maritime Organization (IMO) which has set out to lower carbon emissions from shipping by "at least 40% by 2030 and reduce the total annual GHG emissions by at least 50% by 2050 (compared to 2008 levels)" [4].

To meet these severe emission reduction targets and enable the rapid shift away from fossil fuels and energy sources towards renewable and sustainable fuels, Wärtsilä is developing engines to be run on alternative/carbon-free fuels for both marine and energy sectors [5]. Plans are to introduce engines capable of running on hydrogen and ammonia within the coming years. In addition to the product portfolio transitioning to zero-carbon fuel, Wärtsilä announced its "Set for 30" pledge to achieve strict climate goals by 2030. Wärtsilä is targeting carbon neutrality in its own operations, which "covers direct GHG emissions from the company's own operations, including research and development and engine testing areas, as well as purchased energy" [5]. This requires high effort and high-priority ICE development with a special focus put on SCE testing.

Single-cylinder engine (SCE) testing is an efficient, robust, and flexible tool for combustion process development and mechanical testing of engine parts. It enables performance testing and verification in very early phase projects and a shorter time to market for new products. It is both

faster and more cost-effective compared to conventional multi-cylinder engine (MCE) testing. SCE testing enables specifying and controlling variables that are not explicitly controllable in MCEs. Some examples are the possibility to freely choose charge air and back pressure ratios, exhaust gas recirculation (EGR) control, speed and load combinations that are not possible on an MCE etc. In general, both hardware and control parameters are much easier to modify on an SCE compared to an MCE.

As they are dedicated research and test engines, SCEs differ from MCEs in that they generally are not equipped with any turbocharging (TC) systems or engine-driven auxiliaries (fuel, oil, and cooling pumps). Instead, charge air is supplied by large compressors, back pressure is adjusted by means of valves in the exhaust pipe, and auxiliaries are usually stand-alone electrically driven units (Figure 1). To determine parameters such as charge air pressure, back-pressure, exhaust temperature before the turbine, and friction-related data, presently, simple mathematical models are implemented within Simulink. These, however, require extensive calibration against known references in order to give reasonable and accurate outputs, and are most reliable at mid to high engine loads (above 50% of nominal load), see Figure 2. The current way of testing is also limited to steady-state measurements since transient behaviour can't be modelled with current methods.

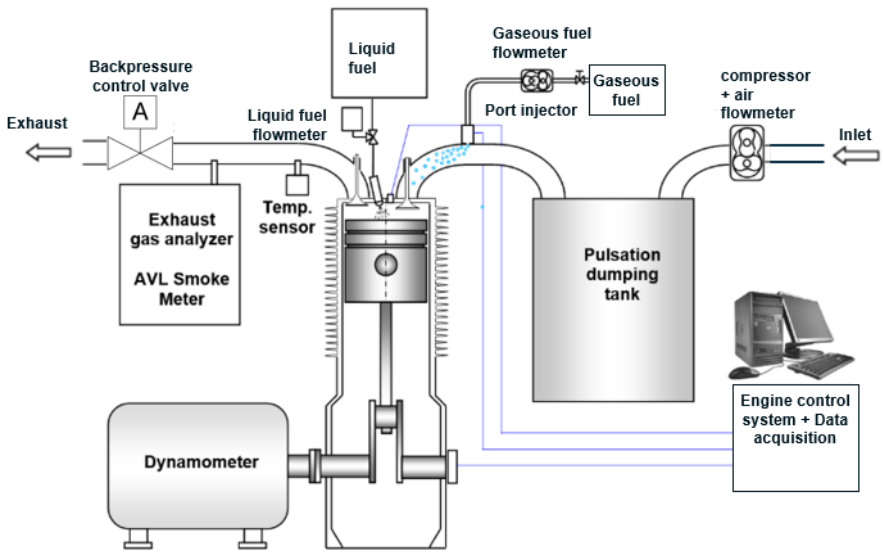


Figure 1: Typical single-cylinder experimental layout [6].

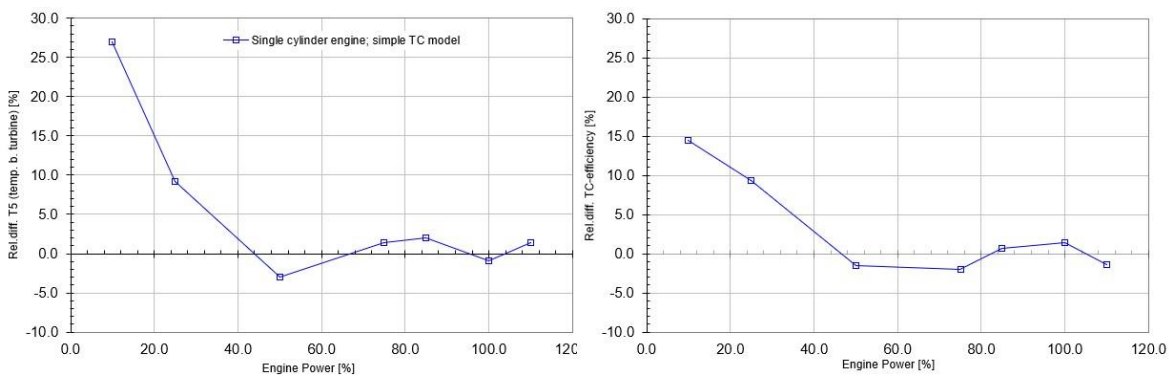


Figure 2: Turbine temp and TC-efficiency vs engine power. The relative difference compared to MCE measured values.

Currently, SCE testing within Wärtsilä is not automatically linked to engine simulation. It is a time-consuming and challenging task to validate simulation models and measurement results. By utilising the 0D/1D-Simulation tool GT-power (GTP), simulation models of the engine would enable running

the SCE more realistically and according to actual MCE conditions. With simulation running in parallel, it is possible to do immediate validation checks on test results. Other benefits include more data availability, such as in-cylinder conditions, which cannot be physically measured (trapping ratio, compression temperature, lambda etc.), TC system data for improved TC modelling, friction data etc. A real-time model running in parallel would also make transient testing possible, e.g. engine loading with realistic TC behaviour (inertia, turbo lag).

1.2 Scope and Limitations

The purpose of this thesis is to enable the SCE to run according to the actual MCE conditions and to calculate the quantities that cannot be measured automatically. This will be achieved by automating parts of the engine simulation process to run in parallel with the SCE. The starting point is a literature review regarding real-time simulation of the ICE industry. The key objective is to give a deeper understanding of the different benefits of real-time simulation and the challenges it poses. Furthermore, the other objectives include “In-the-loop” simulation systems, Speedgoat & Simulink Real-Time environment and UDP communication implementation. The main target is to implement the real-time actuation of combustion data into the MCE simulation model via Simulink based on measured cylinder pressure (combustion feedback) from the SCE. The approaches to real-time engine modelling and combustion development within the ICE industry will be compared to those described in this thesis. The scope also includes an internal comparison between this approach and other ways of implementing combustion profiles to the simulation software, such as SI Wiebe combustion profile, imposed reference combustion profile, and sequential cylinder pressure analysis model.

The simulation model starting point is a detailed eight-cylinder Wärtsilä 8L25 dual fuel MCE model with 2-stage turbocharging. The SCE platform is, in this case running with identical engine main parameters such as cylinder dimensions, engine speed, compression ratio, and valve timing. This work only applies to the W25 SCE engine and its corresponding simulation model. The simulation model is already converted to a fast-running model (FRM) in GT-Power, which is capable of up to 0.5-1x real-time speeds. It has also earlier been connected to Simulink in a co-simulation environment with the possibility to receive data and send it to the SCE via Modbus. In this thesis, the focus lies in speeding up the FRM even more by converting it to an xRT-model, utilising a GT-xRT license (different solver compared to a normal GT license) and imposing a heat release profile calculated from cylinder pressure measurements done in real-time on the SCE. The SCE is already incorporating a Speed Goat HIL system, which can log and send real-time data via UDP protocol. A Simulink model to receive the cylinder pressure data via UDP, calculate the corresponding heat release, and, in turn, send the instant cumulative heat release to GT will be made and implemented. Finally, a comparison between a burn rate analysis based on the detailed GT simulation model and the real-time model using SCE combustion feedback will be made.

1.3 Disposition

The structure of this thesis is as follows: The second chapter starts with a brief overview of GT simulation software, followed by an outline of “in the loop” simulation and a literature review of

real-time modelling within the ICE industry. A short look at Simulink Real-Time & Speedgoat and UDP communication concludes the chapter. The third chapter presents the proposed implementation of the thesis. Several combustion feedback methods are discussed. Further, the GT xRT conversion and the Simulink model build-up for combustion feedback from SCE to MCE model are presented. The fourth chapter presents the results and key findings of this work. Also, real-time capability and accuracy are demonstrated. The final chapter concludes the thesis and gives suggestions for future work.

2 Theoretical framework

This chapter aims to give a theoretical framework for the thesis, beginning with an overview of the simulation software that has been used and its various ways of implementing real-time modelling and a look into “in-the-loop” simulation. Following this, a literature review on real-time simulation within the internal combustion engine industry will be done. The goal is to provide a deeper understanding of the various benefits and challenges that real-time simulation presents. Additionally, other relevant topics related to this thesis will also be discussed, including the Speedgoat and Simulink real-time environment and UDP communication. The idea is to give motivation and a framework for implementing the “real-time SCE-combustion feedback to MCE-model” loop.

2.1 GT-SUITE

GT-SUITE is a prominent software tool for engine and vehicle simulation, widely utilised by engine manufacturers and suppliers to assess a diverse range of topics related to engine and vehicle efficiency. [7] It includes an engine library called GT-POWER which can be used for both transient and steady-state simulations and can be applied to any ICE. One-dimensional (1D) fluid dynamics is the basis for the simulation, providing the user with multiple options to model advanced concepts. It is comprehensive and ideal for integrating all aspects involved in engine and vehicle development. It uses a graphical user interface called GT-ISE (Integrated Simulation Environment) in which model build-up, running and post-processing are handled in an easy way. GT-ISE reduces the need for input data entry as it only requires the definition of unique geometrical elements. Models can be constructed with a point-and-click graphical user interface (Figure 3), utilising a library of templates supplied by GT or by user-defined objects [8]

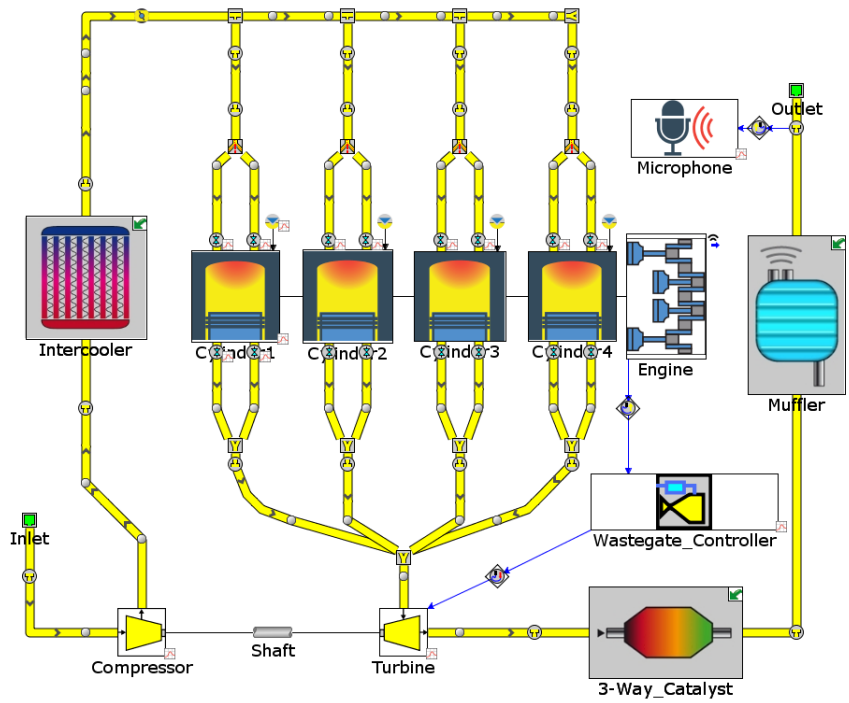


Figure 3: Engine model example in GT-ISE [7].

Typical GT-Power applications can include [8]:

- Inlet and exhaust system design and tuning
- Combustion analysis
- Control system design
- Turbocharger system matching (including variable geometry turbines and wastegates etc.)
- Transient engine response
- Exhaust after-treatment systems

GT-Power can also be used to predict both transient and steady-state behaviour of different engine systems such as [8]:

- Power, torque, and volumetric efficiency of an engine
- Temperatures, pressures, and flow rates in all components of the system
- Burn rates of different combustion concepts
- Nitrogen oxide (NOx) and soot emissions
- Exhaust after treatment chemistry

2.1.1 Mean value model

A high-fidelity 1D model of an engine can have simulation run times in the order of tens to hundreds of times slower than in real-time. [9] [10] To enable HiL systems together with engine and emission controls, models capable of real-time simulation speeds are needed. A mean value model is one way of achieving real-time simulation. It uses a simplified map-based cylinder known as a mean-value cylinder (MVC). The MVC model uses maps to approximate the airflow and fuel distribution in a cylinder, resulting in faster computation compared to models with detailed cylinder components since it is not predicting any combustion or gas exchange processes. This type of model is often used for control system design studies or simulating long vehicle transients, where speed is a priority over the precise characterisation of engine behaviour. [8]

An MVC uses three inputs to approximate cylinder performance. [8] The first is the volumetric efficiency, which is used to calculate the simulated air mass flow rate through the cylinder. The second input is the indicated efficiency, which represents the amount of fuel energy that translates into work performed on the piston. The last input is the exhaust energy fraction, which represents the proportion of fuel energy that heats the exhaust gases. In most cases, any remaining fuel energy is assumed to be lost through heat transfer. However, it is possible to transfer some or all this energy to other components providing a more detailed way of simulating engine heat rejection than the map-based method. The three inputs to the MVC can either be defined as maps or calculated by an external control system. This is useful when the quantities are dependent on more than two input variables, and the flexibility allows the MVC to be dependent on various parameters in the model, such as engine speed, intake manifold pressure, temperature, A/F ratio, valve timing, etc. Artificial neural networks (ANNs) are used to map the three controlling quantities to various parameters, including engine speed, intake manifold pressure, temperature, A/F ratio, valve timing, etc. To determine the relationship between MVC controlling quantities and their dependent parameters and to generate the training and validation data for the ANNs, the detailed engine model needs to be simulated at many different operating conditions. In some cases, several thousands of simulation points might be needed. [11] A design of experiments tool (DOE)

integrated into the GT simulations software can be used to generate the simulation cases. Figure 4 shows a typical mean value model workflow.

Since the MVC model uses a mapped approach, it results in a steady flow within the model, i.e., no in-cycle pressure dynamics. [8] This allows for simplification of the engine air handling system and reduction in simulation times. The intake and exhaust systems can be lumped together to be represented as a single volume of the same size as the detailed model, and a single MVC can be used in place of multiple detailed cylinders. However, this simplification may affect the pressure drops and heat transfer within the system, which may need to be artificially increased. Additionally, the trapping ratio can be actuated to account for scavenging blow-through and exhaust composition.

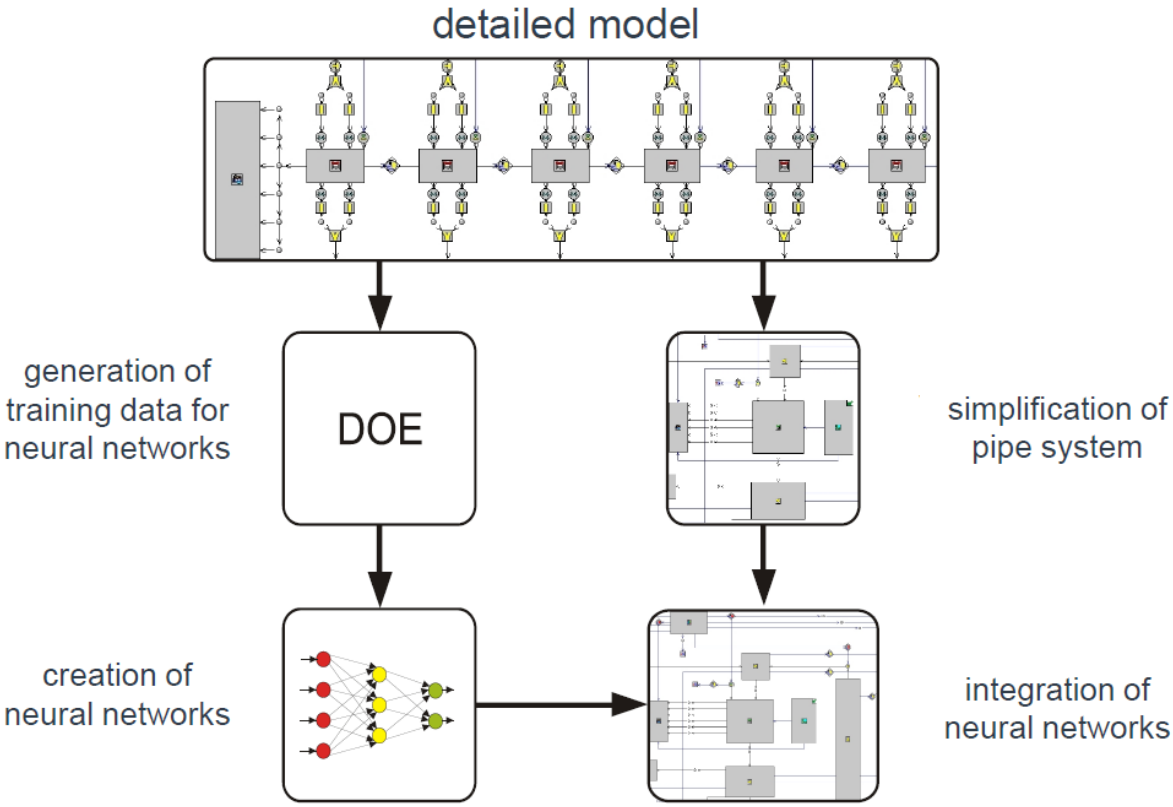


Figure 4: Typical mean value model workflow [12].

2.1.2 Fast running model (FRM)

Fast Running Engine Models (FRMs) are physical engine models optimised for speeds close to real-time while still retaining their predictive capabilities. [8] They are commonly used in system-level simulations where real-time response is needed or where long transient events are being simulated. Unlike high-fidelity models, FRMs can be incorporated into simulations such as Hardware-in-the-Loop (HiL). They can be created by simplifying high-fidelity GT-POWER engine models through a standard conversion process, with the level of simplification determined by the desired balance between accuracy and runtime. FRMs can attain fast run times using two techniques: 1.) Making the simulation time step larger and 2.) reducing the computational requirements per time step. [11] These techniques are often achieved simultaneously by reducing the complexity of a high-fidelity engine model by combining various flow volumes, which reduces the number of sub-volumes and increases the effective sub-volume length, allowing for a larger

time step size (Figure 5). This method is similar to mean value engine modelling; however, in this process, the cylinder components are retained, which allows for the inclusion of combustion and cylinder heat transfer models. Valves and ports are also modelled in the same way as in detailed models. Because the model uses identical solutions as detailed models, it is able to capture wave dynamics that are crucial for studying exhaust gas recirculation (EGR), turbo pulse dynamics, and other related topics. In addition to making pipes and flow volumes simpler, other solver options can also be used to further decrease the number of calculations per time step and reduce run time without changing the time step size.

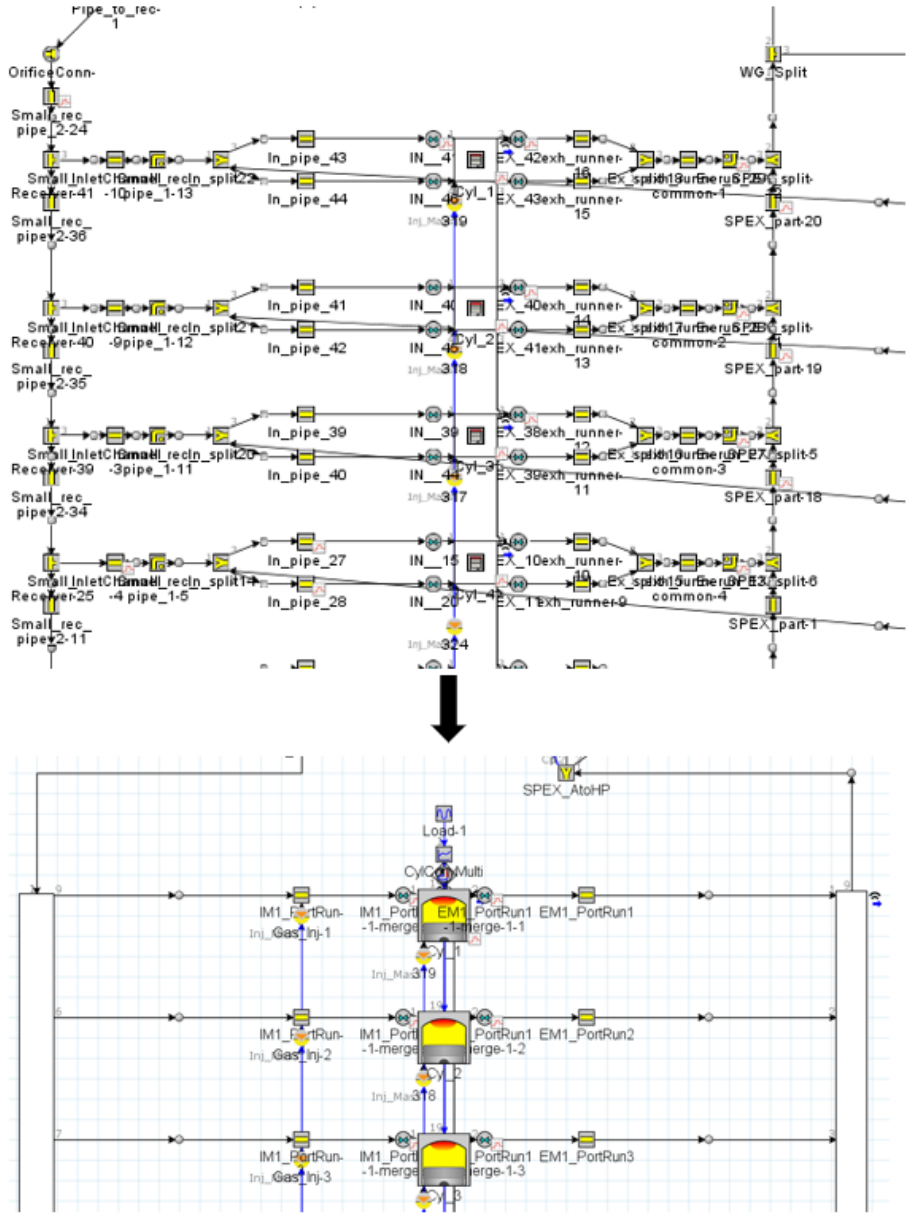


Figure 5: Detailed model conversion into FRM by simplifying volumes and pipes.

When a simplified geometry is used, the number of sub-volumes in the model is greatly reduced, which accelerates the runtime as there are fewer calculations per time step. Furthermore, combined volumes and pipes using a large discretisation permit larger time steps. By increasing the time step of the model cylinder template to a value above 1 deg CA, the default, the model can run with even larger time steps. Together, this enables the possibility for the model to run even faster than in real-time. Although some level of accuracy is sacrificed (such as wave dynamics not being as precise and pressure losses in the system needing to be calibrated), an FRM still has more predictive capabilities compared to a Mean Value model. Additionally, the time-consuming process

of training a neural network using a large set of results from a Design of Experiment (DOE) is not required. [8]

Figure 6 provides advice on the advantages, disadvantages and use cases for both mean value models and FRMs as they can be used for similar applications, in particular in regard to controls and integrated simulation. FRMs clearly allow for higher levels of detail and predictability, but the faster runtime of mean value models may be required for some applications. [8]

	FRM	Mean Value Model
Advantages	- Easy to create from detailed engine model - Predictive Heat rejection from Cylinder - Combustion Modeling - Wave Dynamics - Flexible levels of detail	- Very stable solution - Very fast run time - Neural Nets allow for great accuracy
Disadvantages	- Slower than Mean Value models - Accuracy may be lost	- Longer process to create models (DOE, Neural Net) - Limited predictive capabilities - No Heat Rejection from Cylinder
When To Use	- Thermal Management Models - Integrated System Models - Control Models (SiL, HiL)	- Control Models (SiL, HiL) - Vehicle Models only requiring performance input from engine

Figure 6: Comparison between Mean Value Model and FRM [8].

2.1.3 xRT

GT-POWER-xRT is a real-time solution that is tailored specifically for the needs of vehicle system simulation and ECU calibration. [13] It uses the same physics and Navier-Stokes equations as GT-POWER. As such, it provides exceptional execution speed and, for this reason, allows for higher model fidelity. Control strategies, for e.g., combustion control, can be effectively evaluated using a high-fidelity engine model with increased speed while requiring minimal effort and resulting in maximum accuracy, predictivity, and realism. [11]
It supports integration with Simulink and, in addition, SiL MiL and HiL solutions from various manufacturers (National Instruments, ETAS, dSpace etc.). [13] It also allows for virtual calibration, reducing testing space and costs, and supports current and future ECU sampling rate demands. The original engine model can remain unchanged for SiL and MiL applications. Since HiL applications have a strict real-time requirement and do not allow overruns that exceed real-time for a single time step, they may need simplification to some degree, especially when considering that most HiL hardware is typically slower compared to current desktop or laptop technology. [11]

Main GT-xRT features include [14]:

- Detailed real-time engine models
- MATLAB and Simulink integration
- Predictive models for SI and DI combustion

- Support and developed workflow for several SiL/HiL systems

2.2 In-the-Loop Testing

"In-the-loop" simulation and testing, which refers to testing and evaluating systems in which a simulated system integrates with a real system, is a part of Model-Based Design (MBD). MBD is a process that involves creating a mathematical model of a system, such as a plant or a controller, and using it to simulate and test the system before it is built. [15] [16] Model-in-the-Loop (MiL), Software-in-the-loop (SiL) and Hardware-in-the-loop (HiL) are types of testing that are used to ensure that the function of the system will be correct in the real world. In short, the main focus of MiL simulation is the accuracy of the mathematical models of the system. SiL simulation focuses on the software and its interactions with the hardware, and HiL simulation focuses on testing the control system's response to real-world conditions and inputs. [17] Both simulated and real systems interact in real-time, allowing for more realistic and accurate testing. The simulated system can respond to inputs and changes in the real system, and the real system can then react to outputs and changes in the simulated system. The test results can then be used to improve the model and validate the design before the final product is built. This methodology can be used in many different fields, such as automotive, robotics, and aerospace, to ensure that the system functions correctly in real-world conditions. [15] [18] Figure 7 shows the typical steps in “In-the-loop” testing.

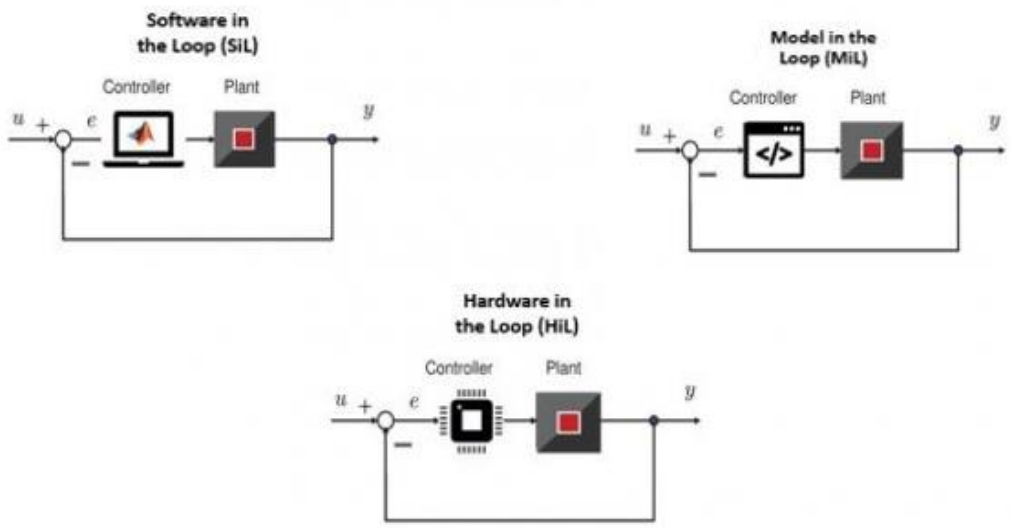


Figure 7: "In-the-loop" testing steps with GT-model as a plant. Adapted from [19].

Model-in-Loop (MIL) (Figure 8) is a technique where modelling the entire system allows for the simulation of a complete environment, e.g., a vehicle or an engine, during the early stages of development. [20] Experimental testing is a standard method used to understand the behaviour of mechanical systems, but in many cases, mathematical models can also be used to describe the system's behaviour. The complete system model enables testing control laws and identifying

mechanical, electronic, or software errors before prototype manufacturing for functional validation. [21]

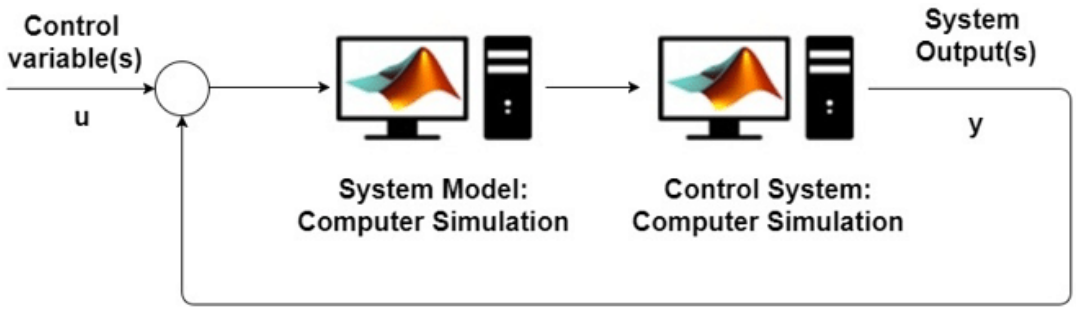


Figure 8: Model-in-the-Loop schematic [22].

Software-in-the-Loop (SiL) testing involves generating code from the controller model and testing it in a virtual environment without hardware to evaluate its performance on the simulated system. The code is tested under various input conditions, functions, and mathematical algorithms to ensure it behaves identically to the model. SiL testing is an effective method for simulating real-time systems that require rapid iterations and validating that the software can meet the requirements. [23] There are several benefits to software-in-the-Loop (SIL) testing, including the fact that it is cost-effective and can run on standard desktop computers without needing special equipment. As it is not limited to hardware, it can be faster than real-time testing. It also enables more adjustability and repeatability and separates software and hardware development. Reusing the SIL simulations can also be done for Hardware-in-the-Loop (HIL) testing, and sharing results across development teams is quickly done [24]

Hardware-in-the-Loop (HIL) simulation (Figure 9) is a method for developing and testing control systems for complicated systems and machines. [18] It involves replacing the physical part of a machine or system with a simulation, allowing for safe and efficient testing. This approach utilises a real-time simulation of a plant that acts as a digital representation of the entire system or specific parts to test the controller's performance. HIL simulation can test the control system's response to various failure modes and beyond the normal range of operation, which increases the safety of the machines and systems. It can also, as earlier stated, be embedded in a model-based design process, allowing for early detection of errors and integration with the control system development process through test automation and script-based testing. With HiL testing, it is possible to confirm the performance of a controller design without the need for the complete system hardware. This

approach allows for comprehensive testing without using the final assembled product in the field, making it cost and time effective.

Figure 9: An example of a HiL simulation setup [25].

One example of a HiL application is AGCO Fendt (Figure 10). They have introduced new HiL test equipment to accommodate the growing complexity of software in modern tractors. The tractor model has been designed in MATLAB/Simulink and contains simulations of the hydraulic valves, the power take-off, the power lift, and the I/O driver blocks. LMS AMESim, a software package for multiphysics simulation, is used to design the engine and drivetrain models. LMS AMESim allows for easy parameterisation for various transmission types, vehicle dynamics simulation, and easy importing into the MATLAB/Simulink environment. [26]

Figure 10: The HiL test bench at AGCO Fendt [26].

2.3 Real-time modelling within the internal combustion engine Industry

The requirements for engine development stated in the introduction are challenging for all ICE manufacturers. Due to cost and time reasons, it is necessary to design and evaluate both combustion and control concepts for engines in the early stage of the development process, especially with strict performance requirements that demand complex combustion systems. For this reason, real-time modelling is increasingly being used for its ability to quickly verify concepts and performance.

There is a significant number of similar efforts related to real-time modelling and simulation that have been reported in the literature, although an identical approach as described in this thesis could not be found. A lot of what has been developed within the ICE industry and within universities in recent years concerns virtual testing, calibration and controls development with much focus on HiL systems [9] [10] [27] - [28]. Some concepts also specifically concern the implementation of GT-POWER xRT models [29] - [30]. In addition, there are some interesting approaches to digital twins that also utilise real-time GT models. [31] - [32].

Ryu et al. [9] describe the development of a virtual simulator for an Electronic Engine Control System (ECS) using hardware-in-the-loop simulation (HiL) as an alternative to traditional testing on engine test benches (Figure 11). The simulator uses a real-time calculation model for the combustion of both gas and diesel fuel based on a detailed 1D engine model that is converted to a mean value engine model. The simulator also includes models for the intake and exhaust systems, turbochargers, engine starting units, and fuel injection systems. It also adds crank-angle resolved math models for cylinder pressure and knock signals. The ECS test was executed using virtual scenarios over the entire engine operation, and some engine control strategies were evaluated to enhance efficiency and reduce emissions. The results of the virtual simulator showed similarities with the actual performance of dual-fuel engines, with calculation times less than 0.75 times in real-time.

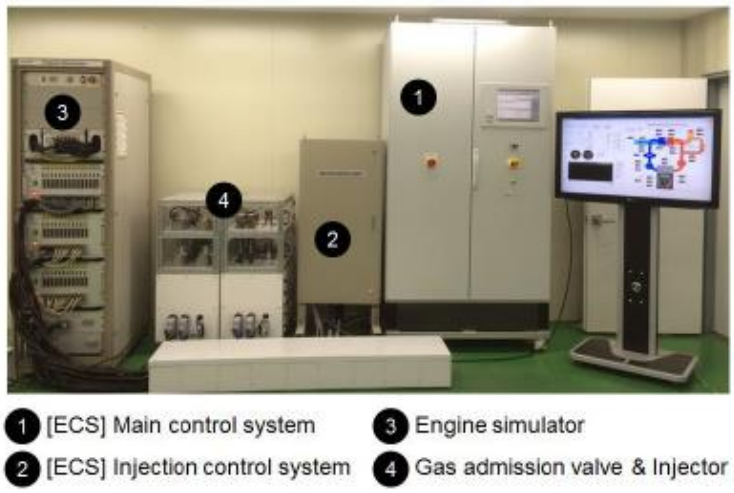


Figure 11: HiMSEN HiL system layout [9].

The detailed 1D model used as a starting point was developed in GT-power and represents all components of the engine, including turbocharging system, wastegate, intake and exhaust manifolds, intercooler, and cylinders. [9] The model is highly accurate in capturing the gas dynamics of the engine system, making it ideal for use in engine design and evaluation of different component geometries and selections on engine performance. However, due to its complexity and high level of accuracy, the model's run time is relatively slow, typically 100-1000 times slower than real-time. The next step in the process involved taking the detailed 1D engine model and converting it into a faster-running version while minimising loss of accuracy. A major part of the challenges with high-accuracy engine simulation is the cylinder itself. To achieve accurate results, the time step used in the simulation must be very small, typically less than one crank angle degree. In addition, the combustion is calculated separately for each cylinder, increasing simulation time as the number of cylinders increases

In the paper of Ryu et al. [9], a couple of different approaches to cylinder modelling were tested. The detailed engine model uses a DIJet model for diesel combustion approximation which uses several sub-zones to achieve high fidelity. This leads to a significant calculation time, and as such, it is not appropriate for real-time applications such as HIL. A simplified combustion model, DIPulse, was subsequently tested and showed a significant, while an inadequate, reduction in simulation time while preserving the high fidelity of the DIJet model. The engine model utilising DIPulse for combustion simulation is on the order of 7 times faster compared to the model using DIJet.

However, in order to achieve the real-time simulation target needed for HIL, the detailed cylinder model was replaced with a mean value cylinder (MVC) model. To test the mean value engine (MVE) model, a scenario involving a series of varying speed and load conditions was tested. The scenario was assumed to be 470 seconds in real-world conditions. The resulting points at steady state conditions showed less than a 1% difference between the detailed model and the MVE, and the MVE was capable of simulations times less than 0.75 times in real-time. It was noted that the MVE overpredicts engine speed during load steps, with the main reason being turbocharger response. In conclusion, the MVE overall showed a good correlation with the detailed model in terms of performance and engine dynamics, and it was found suitable for real-time operation and Hil simulation. The developed models were integrated into a HiL simulator (National Instruments PXI-8119) and connected to a real HiMSEN 8H35/40DF ECS. The primary concern in performing HiL simulation for ECS testing is ensuring that the simulation results align with the behaviour of the real engine. If there are discrepancies in the trends and the control algorithms are not properly designed, it can lead to dangerous situations when used in actual engine applications. [9]

Wu et al. [10] describe the process of simplifying a high-fidelity model into a GT FRM and verifying its performance on a dSPACE Simulator Hil test bench (Figure 12). The authors [10] claim that HiL technology is becoming an increasingly important technology within the automotive industry and is already being used within all aspects of development, such as combustion and engine control, vehicle dynamics etc. For the implementation of HiL simulation, there are a couple of different types of models most commonly in use. Ordinary differential equation models, typically control-oriented, are easily implemented in real-time but lack fidelity. 1D physical partial differential equation models, on the other hand, provide the highest number of details but are comparatively slow. Within this work, the target is to "*simplify the 1D model into fast-run models for real-time HIL while still retaining the accuracy of detailed models".* [10] GT-Power is used to model the engine due to its comprehensiveness and ease of use. It also enables Simulink co-simulation for control simulation and allows for a coherent transfer from engine development to control implementation.

A series of tests were carried out to check the real-time capability and consistency of the results of GT-Power FRM on HIL. [10] A load manipulation test result showed that the rapid controls prototyping (RCP) PID controller and the turbocharger controller in the original GT-Power model reached the same steady state without significant errors. The FRM HIL system can also simulate the pulsing dynamics in the airflow. Some variations were seen in the PID controller during the transients due to inconsistency in the algorithm. Adding a throttle angle step change showed that the load target was reached with minor differences. The spark timings and fuelling pulse width were also investigated, showing inconsistencies in steady-state conditions but good transient behaviour, mainly because of model simplifications and limitations in the sampling rate of the hardware. For real-time simulation, repeatability in pressure and heat release traces are essential. The balance between how complex a model is and its ability to perform in real-time must be determined by

considering the specific needs of the control application. [10]

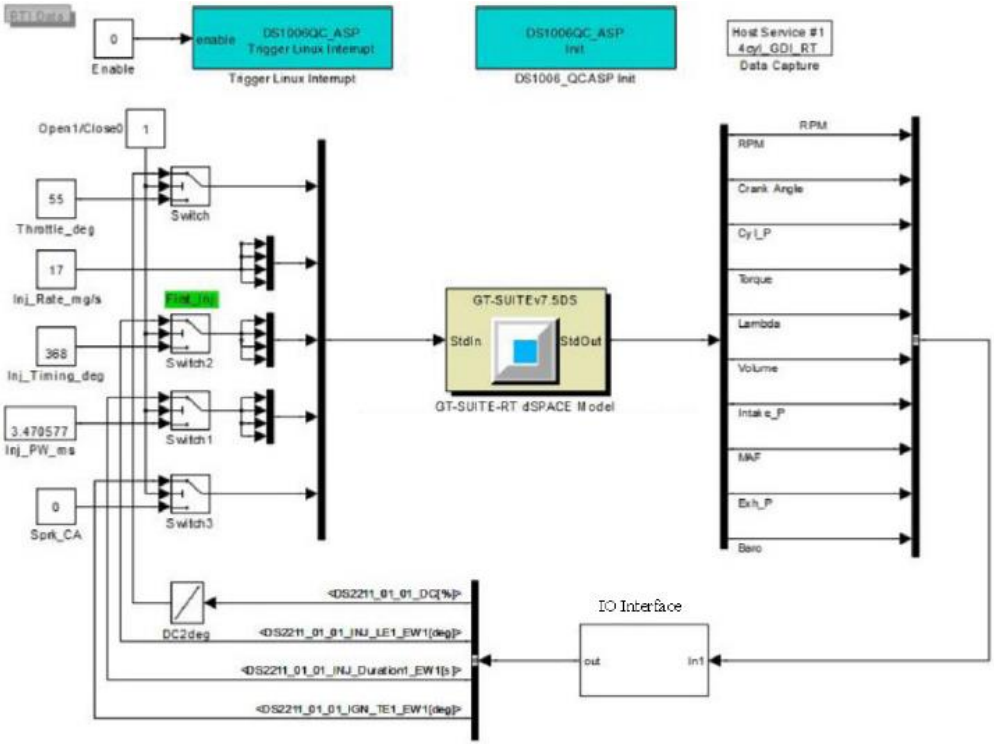


Figure 12: FRM to HiL/RCP interface in Simulink [10].

The thesis of Chandrashekar [27] aimed to convert an existing detailed GT model of a heavy-duty diesel engine from Volvo Trucks into an FRM. The goal was to improve transient performance as the initial model only performs satisfactorily. Volvo Trucks have developed a Global Simulation Platform (GSP) to enable the simulation of whole-vehicle operations by utilising an experimental engine plant model (Figure 13). To improve the GSP performance, the experimental engine model was proposed to be replaced with a real-time GT model. The target was to enhance the evaluation of new concepts for reducing emissions or fuel consumption and cutting the cost of repeated experimental testing. The thesis presented the FRM methodology as a key to improving the transient performance of the model, and its accuracy was verified by comparing it to results from a Volvo engine test cell. In addition, the FRM also included a hybrid system to showcase its capabilities in the GSP. The FRM was integrated into Simulink with inputs from a virtual engine management system and verified by comparing it to input from steady-state mapped data. In general, the results matched the ones from the original model, apart from exhaust gas temperature at low load operation. Transient inputs from a drive cycle test were used in open loop simulations on the FRM and later compared to results from a real engine running the same drive cycle. The majority of the results from the FRM match the ones from the physical engine; however also, in this case, the exhaust gas temperatures were inconsistent. To enable correct simulation of the engine after-treatment system, the implementation of a thermal network would be required to improve

the results of the simulated exhaust gas temperatures.

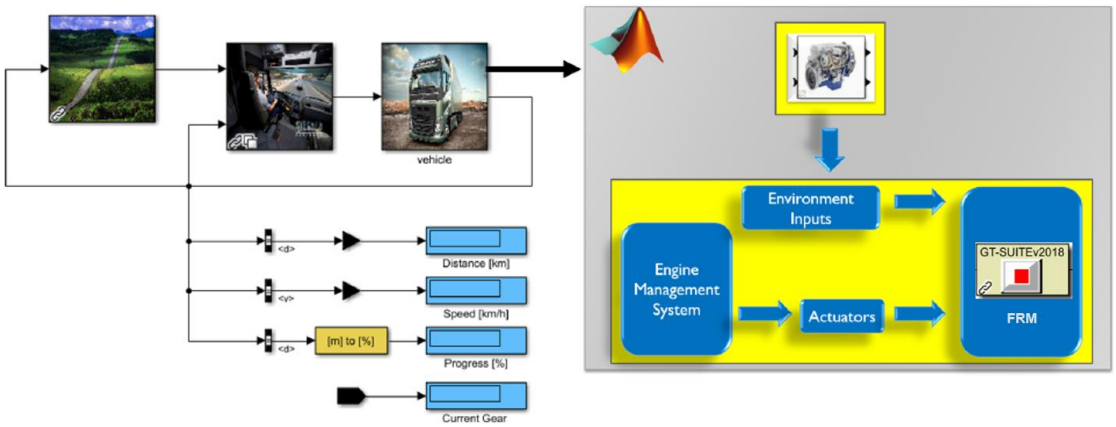


Figure 13: Overview of GSP and Simulink model, adapted from [27].

The work of Xia et al. [33] describes a solution for virtual system integration and testing that utilises HiL simulation, which allows for the early completion of development tasks, offers a safe and dependable environment for testing, and lowers the expense associated with prototype hardware. The idea is to obtain the HiL plant model directly from an existing engine model with a co-simulation approach, resulting in a streamlined and efficient model deployment and verification procedure. According to the authors, there is a major challenge to establish model accuracy that is acceptable to reduce the calibration effort with prototype hardware. The trend shows that model complexity in HiL applications is growing in order to handle the calibration of advanced engine control units (ECU). A model-based design process should be utilised to minimise the effort required for engine modelling in HiL applications.

The starting point is a detailed 1D diesel engine model with exhaust gas recirculation (EGR) made in GT-Power, which is converted to a real-time capable, crank-angle resolved 0D model with predictive combustion capabilities and validated against real engine test bench data. [33] The real-time capability of the 0D model is crucial for HiL systems. Overruns, where the simulation calculation time is longer than the HiL platform trigger frequency, can have very detrimental effects on the simulation and calculation accuracy. The model is then connected to a HiL system (FEV xMOD). The HiL platform simulates a complete powertrain in a closed-loop system (Figure 14). It includes real-time models of various components and a physical ECU. As the computing requirements are high for the engine model, it is run on a separate system, while the ECU and the other powertrain models are run on another system. The signals between the two real-time systems are handled via a UDP interface. Running the HiL simulation on two separate systems introduces challenges with transfer delay, as all HiL models are executed in real-time. In this case, clock skew introduces a jitter of 4-8ms due to the two systems' clocks not being synchronised. Another challenge is the varying latency that the UDP connection causes. The combined effect was noted to be below 10ms which was considered acceptable since the shortest engine cycle is 12ms

and the ECU sample time is more than 10ms. [33]

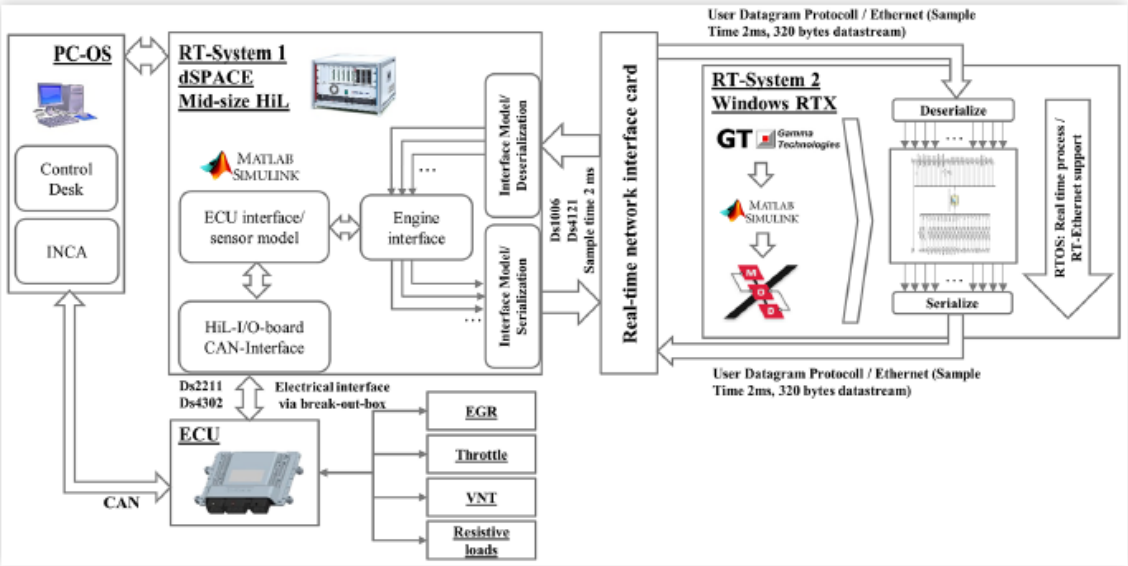


Figure 14: Co-simulation of engine model and ECU on a HiL platform [33].

The results of using the 0D engine model in a closed-loop with the ECU in a steady-state operation during HiL validation were comparable to those obtained during offline validation. The following steps include model integration with the HiL platform for virtual calibration, utilising the 0D model to give input to emission and after-treatment models. This will additionally enable the simulation of complete driving cycles and adverse ambient conditions. [33]

Road-to-Rig-to-Desktop (R2R2D) is an approach evaluated by Andert et al. [34]. In essence, it is a process for powertrain development that uses engine test bench simulations and off-line simulations to estimate a wide range of parameters and testing procedures in the preliminary development stage, resulting in significant cost savings compared to physical testing. According to the authors [34], the main challenge with R2R2D methodology is balancing the accuracy of the simulated models and the computational time required to run them. Models that provide the highest accuracy in terms of representing the physical behaviour, e.g., 3D-CFD, tend to be too computationally intensive for normal vehicle simulations. Alternatively, it is possible to utilise data-based look-up tables to represent physical components, but this requires a significant amount of measurement data and may not be suitable for complex systems. Building and maintaining a full vehicle model with a testing environment can be a significant undertaking. However, component models that are well-maintained are often already available during the development phase. The R2R2D method is to reuse these component models by the use of a co-simulation system. Simulation tools that combine physical functions with phenomenological or empirical models and allow for customisation for the specific application are the most suited for R2R2D.

The R2R2D process was demonstrated using a real-time engine model based on a turbocharged gasoline engine and a vehicle co-simulation that included, among other things, a dual-clutch transmission, chassis, and environment (Figure 15). *"Basic control functions for engine and transmission, together with a driver controller, are implemented to ensure a closed-loop simulation reacting to the reference driving manoeuvres. "* [34] The transient behaviour of the engine model was checked using Real Driving Emission (RDE) profiles that represent real driving conditions. The resulting simulations were compared with vehicle measurement data and showed a discrepancy in fuel consumption of less than 5% and accurate predictions of the transient behaviour and key

engine operation points. The engine model was proven to be able to be used in system testing phases effectively, and the work done to convert and calibrate the model was acceptable. [34]

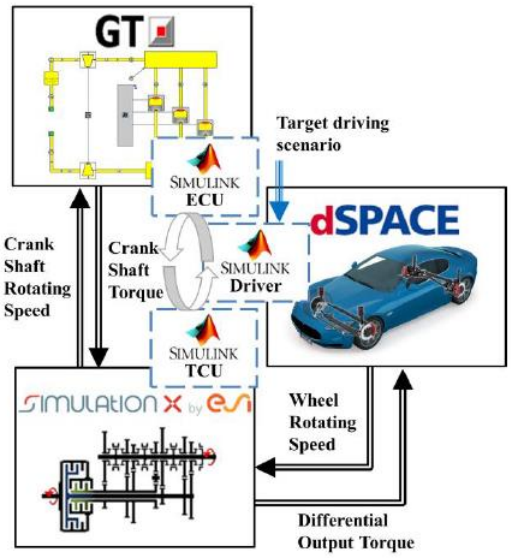


Figure 15: System structure including different models and control units [34].

The papers of Andric et al. titled *"Development and Calibration of One Dimensional Engine Model for Hardware-In-The-Loop Applications"* [28] and "*Calibration Procedure for Measurement-Based Fast Running Model for Hardware-in-the-Loop Powertrain Systems"* [35] outline the development of a real-time capable model of a diesel engine that can be used for HiL simulations utilising GT-SUITE. The papers mention a novel approach where a detailed simulation model has not been utilised in the development of real-time capable models. Both mention the successful use of the developed models at the VIRtual TEst Cell (VIRTEC) at Volvo Penta.

In the first of the two papers, [28] Andric et al. describe the approach as creating the FRM by *"the use of the top-level engine configuration, test cell measurement data, and manufacturer maps"* instead of using a detailed 1D model, which is a more prevalent method (Figure 16). The idea with this approach is to create robust and flexible models that are easier to adjust and calibrate, to enhance the utilisation of HiL simulations. Models employed in a HiL system are crucial for understanding non-linear and other complex behaviour, and the models need to be reliable and precise, able to simulate the transient behaviour of the engine and satisfy the real-time requirements. Andric et al. [28] make the same conclusion as Xia et al. [33] that the main difficulty with HiL simulations is to ensure that the models are accurate enough while also satisfying the requirement of fast enough simulation time steps.

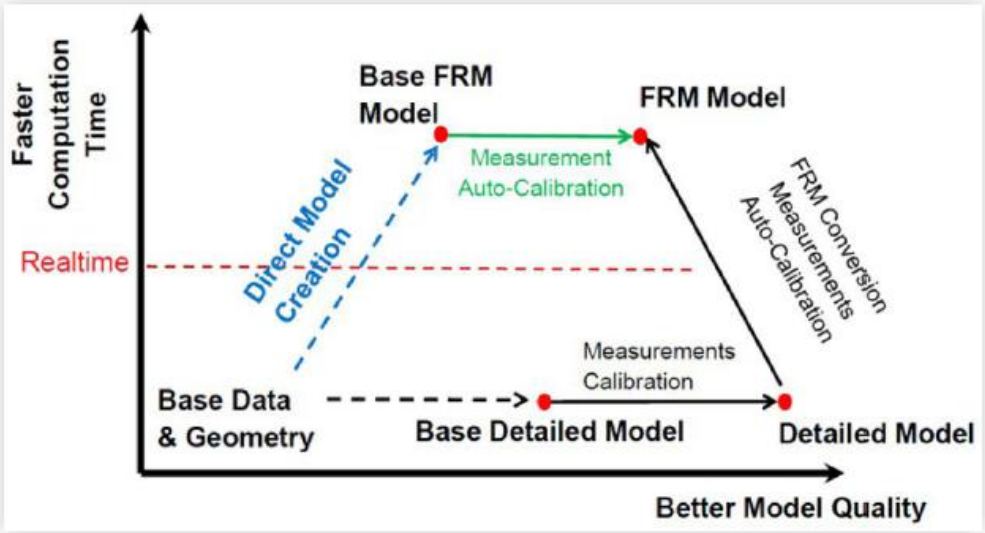


Figure 16: Direct FRM model creation versus traditional approach [28].

For the development of the FRM, a direct model creation approach based on physical engine data and geometry is used. A DI Pulse predictive model was used to estimate the combustion, and gas exchange dynamics were modelled with crank-angle resolution. An auto-calibration of the model was made using part load map operating points. The modelled gas exchange dynamics, such as inlet/exhaust flows and pressures, showed a good match with real part load operating data. NOx emissions, on the other hand, were not estimated with high accuracy and are an important area for future work. Further steps must be taken to guarantee that the model meets all the demands of virtual testing. Future work should incorporate testing the model with various engine configurations to guarantee its reliability. In summary, the approach to FRM model creation is a viable option for use in HiL simulation environments, and the research in the paper illustrates a significant step forward in the creation of reliable predictive models that can be used for enhancing engine operation through HiL-based methods. [28]

In the second paper [35], Andric et al. focus on the calibration procedure from a baseline real-time capable model to a model ready for HiL system implementation. The authors define the method for developing and calibrating an engine model only through engine dynamometer test data. Like the previous paper, [28] This case does not either utilise a detailed model, but instead, a direct baseline real-time capable model is created in GT-SUITE. [35]

The integration of Model-embedded control, model-based calibration and virtual test beds is becoming more common in the development chain. [35] However, there is still room for improvement in optimisation techniques for complicated systems. The implementation of new technologies to achieve emission requirements and improve fuel economy has led to an increase in the number of variables to consider in the calibration process. Additional complications arise from the need to consider non-standard operating conditions, leading to significant demands on the Engine Control Unit (ECU). Acquiring a large amount of data for each operating point through conventional Design of Experiment (DoE) methods is costly and time-consuming for ensuring high-quality calibration. Therefore, reducing the number of measurements to minimise the time and costs of the calibration process is important. The model-based approach for engine development has several advantages, such as achieving excellent performance with less ideally allocated measurements. It can safely and reliably consider various constraints and perform optimisation during transient operation. The method allows calibration engineers to quickly acquire a comprehensive understanding of the interconnected behaviours of the engine. Additionally, it provides a thorough understanding of the powertrain system, including the relationships and interactions between its various components. [35]

The resulting Fast Running Model is more robust and accurate, indicating its suitability as a method for virtual testing applications. [35] The unique features of this methodology include not relying on manufacturers' inputs, efficient model creation, increased model reusability, and user-friendly calibration tools. The improvements in modelling and optimisation can save time and resources, enhance the repeatability of simulations, help engineers with calibration, and emphasise the use of model-based techniques in engine calibration.

A couple of xRT-specific implementations include the presentations by Patel et al. (Mercedes-Benz) [29] and He et al. (Scania CV AB) [36], which both consider the challenges of using different models for different types of testing and the possibility to utilise GT-xRT models for both SiL and Hil testing.

In *"Implementation of a Real-Time GT-POWER-xRT Engine Model in Virtual Test Bench (VTB)"*, Patel et al. consider the utilisation of a real-time engine plant model for use in virtual test benches (VTBs) for hardware and software testing, reduction in control unit development time, and checking the functionality of the model. The challenge with VTBs is that different plant models may be required for different VTBs, leading to differing results. The goal was to develop a real-time gasoline engine plant model that exhibits the same behaviour in different VTBs. The process involved converting an existing detailed engine model into a fast-running model (FRM) and calibrating and validating it for transient behaviour operation with ±5% accuracy. The model was then developed using an xRT license and integrated into both a SiL and HiL simulation environment. Validation was conducted in both transient and steady-state conditions. The final model showed the same behaviour in both SiL and HiL environments, fulfilling the requirements for both runtime and model accuracy, meaning no different plant models are required for different VTBs. [29]

The presentation by He et al. discusses the development of a GT-xRT gas exchange model for use in software-in-the-loop (SiL) and hardware-in-the-loop (HiL) testing by Scania. Three different business units within the company currently handle gas exchange system simulations using different models: a group that specialises in detailed but slow models, a SiL testing team that uses GT Fast Running Models (FRM), and a HiL testing team that uses map-based ETAS models. The goal is to create a tool that can minimise double work, model development where competency is, and reliable test environments for function development and initial software calibration. The xRT license is presented as a solution which is specifically designed for ECU calibration and testing and supports HiL testing, providing the capability for greater model complexity. It is compared to other alternative tools such as FRM with GT-suite license, Mean value engine models with GT-suite license, and RT License. The xRT model is evaluated for its performance, conversion time, real-time speed, and accuracy in both SiL and HiL testing. A Virtual Truck and Bus (VTAB) simulation platform is used for SiL testing, and the evaluation goal for HiL testing is to check if the xRT model compiles and performs well in terms of accuracy and real-time performance. The summary states that compared with an FRM, the xRT model is better in accuracy and conversion time but has a lower real-time factor and some limitations. It allows for the possibility of using the same model in both SiL and HiL testing, reducing unnecessary double work. [36]

Another approach to xRT modelling is shown in "Development of a Real-Time GT-POWER xRT Model for Virtual Calibration" by Boccardo et al. [37]. The authors explain that the control system development within the automotive industry is evolving rapidly due to regulatory pressure, shorter time-to-market, and control system complexity. VTBs are used for reducing the need for physical experimentation, quicker implementation of the design of experiments, improve scalability and advancement of control system development at an earlier stage in the development process. The GT-POWER xRT engine FRMs are fast, accurate and predictive, making them ideal for virtual calibration (Figure 17). The aim of this work is to identify a process for the development of GT-

POWER xRT engine models for virtual calibration activities. The Engine Models in this study represent the Jaguar Land Rover Ingenium 2.0Lt Diesel Engine. Multiple xRT models were checked to determine real-time capability, stability, and robustness, to identify the best xRT model for virtual calibration. The xRT model proved to be quite accurate despite the computational efficiency and was the basis for a virtual calibration feasibility check. [37]

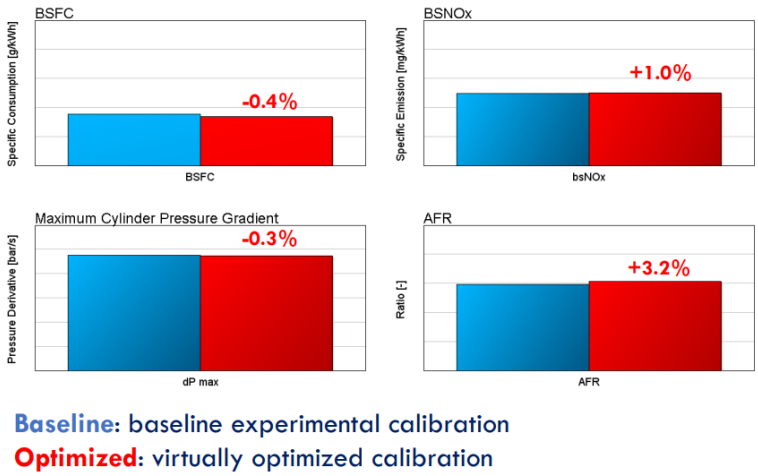


Figure 17: Virtual calibration results [37].

The presentation by Padmavathi et al. [30] shows an end-to-end simulation process that can generate inputs for RDE/NEDC cycles, generate outputs from the ECU, predict engine-out emissions, and perform after-treatment simulations (Figure 18). This is being done to reduce development time and cost by reducing the need for engine test bench and vehicle calibration, exploring and optimise various engine inputs and strategies to meet emissions standards, and integrating ECU, engine and EATS to determine the feasibility of meeting emissions standards, and investigate extreme scenarios without risking hardware failure. The process involves using different tools and modelling approaches such as GT-DRIVE+ for on-road cycle data and vehicle model, MATLAB SIMULINK for ECU modelling, GT-POWER xRT for engine simulations and GT after treatment for EATS simulations. The model has been validated for both RDE and NEDC cycles, and future plans include the potential for soot emission modelling and validation for extreme ambient conditions.

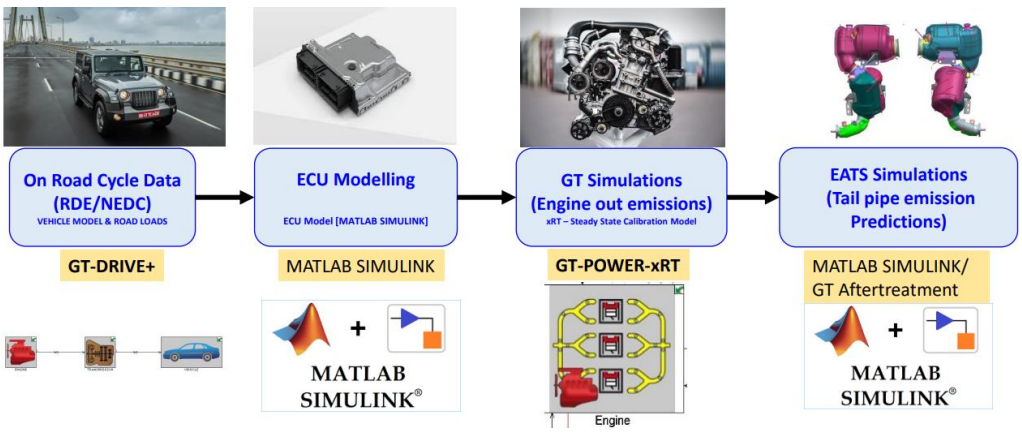


Figure 18: A modelling approach for the end-to-end simulation process [30].

The works of Hautala and Söderäng et al. [31] [38] [39] [40] concern developing and implementing a digital twin of a research engine in various applications. Digital twins are virtual representations of physical objects or systems. It can be used to simulate and analyse the performance of the physical object or system and can be used to optimise its design, operation, and maintenance. Digital twins are often used in manufacturing, transportation, engineering, and other industries to improve efficiency and reduce costs. They can be based on sensor-, historical-, or other forms of data and can be utilised to predict future performance and detect and diagnose issues. [41]

In *"Digital-twinning the engine research platform in VEBIC"* [31] and *"Real-time simulation of the combined engine and electrical equipment model in Simulink RT"* [39], the approach is to implement a digital twin of a hybrid power system in a laboratory environment (Figure 19). The target is to investigate the potential benefits and challenges of using digital twins. The research involves developing existing laboratory equipment component models, modelling the hybrid power generation system, and converting the model into a real-time application. MATLAB/Simulink is used to model the electrical components, and GT-POWER is used to model the engine. Real-time simulations are run using Simulink Real-Time and a Speedgoat performance real-time target machine. The authors target optimising the engine design and operation by running the actual engine in parallel with the real-time application, enabling the integration of technologies such as energy storage, waste heat recovery, and emissions reduction.

In the first work [31], the engine model was converted to an MVM to ensure real-time capability. However, the MVM was limited to only a few operating points due to a lack of test and validation data from the physical engine. To run the MVM on the Speedgoat real-time target machine, it needs to be converted and utilise a special GT-SUITE RT license. The real-time target machine acts also act as a communication interface between the digital twin and the physical engine.

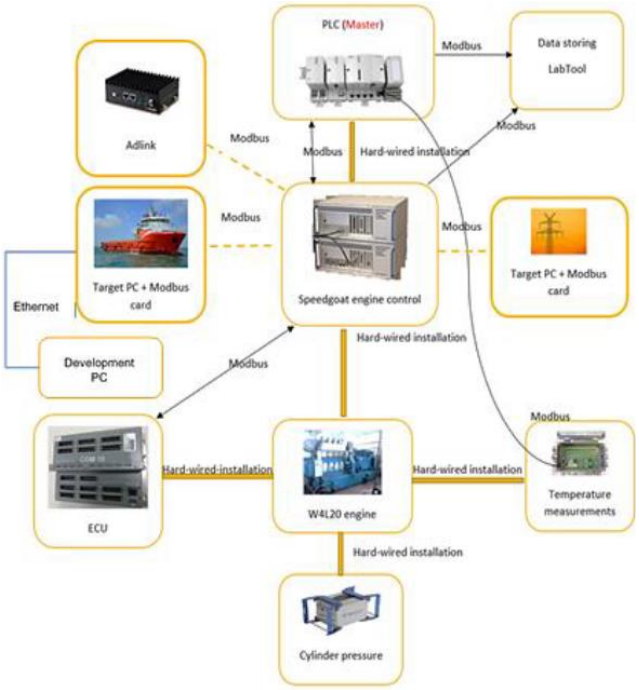


Figure 19: Digital twin setup in VEBIC [31].

In [38] - [40], which is a continuation of [31], the focus lies on achieving combined real-time modelling of both engine and electrical equipment. The initial MVM process was not used due to a lack of measurement data, and instead, the FRM method was selected. An FRM was deemed to be relatively straightforward, and it also allows for some details to be kept in the model, e.g., combustion modelling, while still achieving real-time performance. After the FRM conversion process, simulation results were compared to the detailed model with all parameters remaining

within the expected 5% error tolerance, except for exhaust gas temperature for a single operating point. This was, however, accepted due to the intended use. The FRM model was capable of reaching a real-time factor of 0.35 [39]. After the model was confirmed to be real-time capable, it was linked to Simulink for co-simulation with the electrical equipment models (Figure 20). A 200-second simulation of the models was performed to test their functionality. The results were compared to measured data and matched those of the FRM run in GT-SUITE. The conversion of the models into a combined real-time application proved to be challenging but was eventually solved. To reach real-time capability, the FRM needed a long sample time, causing poor accuracy. The combined model could not reach real-time speeds due to insufficient computational power on the target computer. The following steps involve running the combined model on the Speedgoat target machine in order to reach real-time capabilities for the combined system [38] - [40]

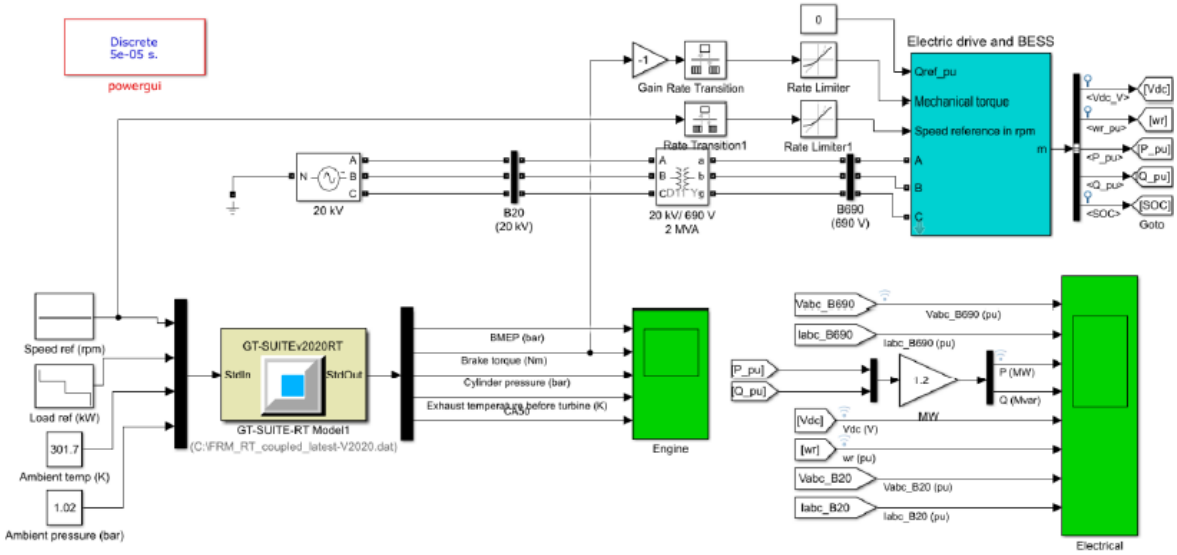


Figure 20: Simulink model of engine and electrical equipment [39].

In *"Development of a diesel engine's digital twin for predicting propulsion system dynamics"*, [42] Bondarenko et al. describe that making a digital twin, a virtual copy of a real ship or system that can track and predict its behaviour in real-time, is a solution to understanding and optimising the dynamic process of the propulsion system. Operating at the most favourable efficiency is the aim of the propulsion system, and a digital twin is an appropriate solution to achieving this. The digital twin is a collection of physics-based models that reflect the characteristics of the propulsion system, and diesel engines are a core part of the digital twin. [42] However, as others have also noted [33] [28], a challenge in creating digital twin and real-time engine models is achieving a balance between gaining detailed insight into the dynamic behaviour and meeting real-time execution requirements.

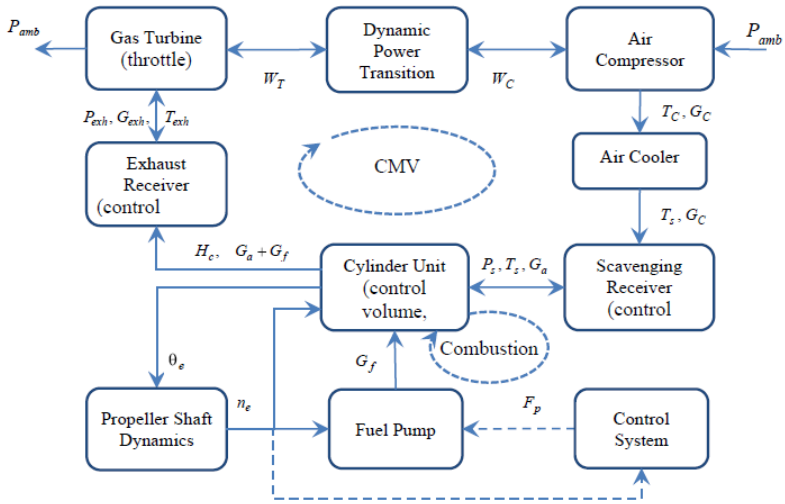


Figure 21: Cycle-mean value model [42].

The paper introduces a method of modelling that utilises a combination of a cycle-based MVM (Figure 21) with a Wiebe-based combustion model that is crank-angle resolved to meet these requirements and provide a faster and more accurate solution than previous methods (Figure 21). The developed model that calculates in-cycle pressure variation outperforms traditional methods and has acceptable performance when compared to actual engine results from an engine testbed. The developed digital twin can be used for several applications, such as anomaly detection, follow-up of thermal efficiency and emissions, and evaluating transient response. The authors note that there is still room for improvement, specifically in low-load predictability. Additionally, the basis for the model's performance is a limited amount of test data and fixed environmental conditions. For this reason, the model needs continuous adaptation in order to stay up to date with the actual operating conditions of the real ship. [42]

Another approach to digital twin concepts has been studied by Saurabh et al. at Cummins Inc. [32]. Within this work, the focus has been on digital twin usage within the automotive industry. It was specifically related to tracking vehicle health and diagnosing issues with the goal of improving commercial automotive engine service and repair by implementing a new simulation-based method using a digital twin of the engine (Figure 22). According to Saurabh et al., failure mode isolation and diagnostics when an issue with a vehicle or engine is detected is a less studied aspect of digital twin concepts within the automotive industry. The use of digital twins in diagnostics can improve service by identifying the specific failed component with a high level of confidence before the vehicle with the issue arrives for maintenance, reducing unnecessary testing of components, and decreasing overall service costs by shortening troubleshooting time and increasing the likelihood of replacing the correct part. The digital twin simulation uses a 1D GT-POWER engine model. If a diagnostic trouble code (DTC) is activated on a car or other vehicle, the simulation model (digital twin) performs an analysis of the related failure modes using a couple of minutes of transient engine data as input. This is expected to be a valuable tool for reducing vehicle downtime and costs related to maintenance.

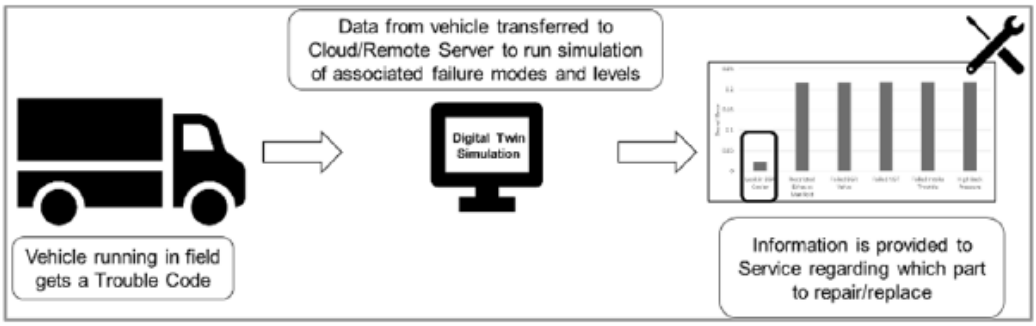


Figure 22: Digital twin diagnostic process [32].

By utilising a real-time 1D GT-POWER engine model, it was able to examine real-world failure cases where DTCs were triggered on a running vehicle. The model was modified to include potential failure modes causing DTCs. Simulation predictions and reference data from the truck were compared using an Overall Error metric. The approach was successful in identifying the correct failure and could aid service technicians in troubleshooting and repair planning. Future plans include expanding the approach to other engine components and implementing it in a real-time digital twin environment. [32]

The amount of literature related to both real-time modelling and single-cylinder testing is quite limited. However, some interesting works could be found. Mayr, P. et al. describe in the paper titled "Simulation-Based Control of Transient SCE operation" [43] a methodology for simulating the transient behaviour of an MCE on an SCE test bed and operating it through a Hardware-in-the-Loop system. Their goal is to improve both gas and dual-fuel engine transient characteristics, which can help them compete with diesel engines. The methodology includes both simulation and measurement methods and is designed for use with a large-bore gas engine with two-stage turbocharging. It utilises fast and simple physics-based models that can provide real-time boundary conditions for the MCE in transient operation, including boost pressure and exhaust back pressure. Measured values of the SCE test bed are then fed back into the real-time models to calculate the transient MCE behaviour for the next time step. In this setup, there is no combustion feedback to the MCE models. Only single-value cylinder pressure at the exhaust valve opening (EVO) is used as input for the gas exchange model. The gas exchange of the MCE is modelled by a simple 0D method to ensure real-time capability. The methodology also involves the implementation of the models on a real-time system and reaching an accuracy of the model that is good enough to avoid calculation errors. This method enables the transferability of boundary conditions from the MCE to the SCE during transient operation. The models are validated on an SCE test bed through a suitable interface between the MCE model and the hardware. In this specific case, 1D GT-power models are deemed to be too slow for real-time operation and are only utilised to validate the 0D algorithms that are used to describe the transient gas exchange. The HIL real-time controller not only controls the SCE test bed but also acts as a platform for the different models and the MCE control unit. This setup allows for the development of MCE control strategies on the SCE test bed. Figure 23 shows the layout of the system.

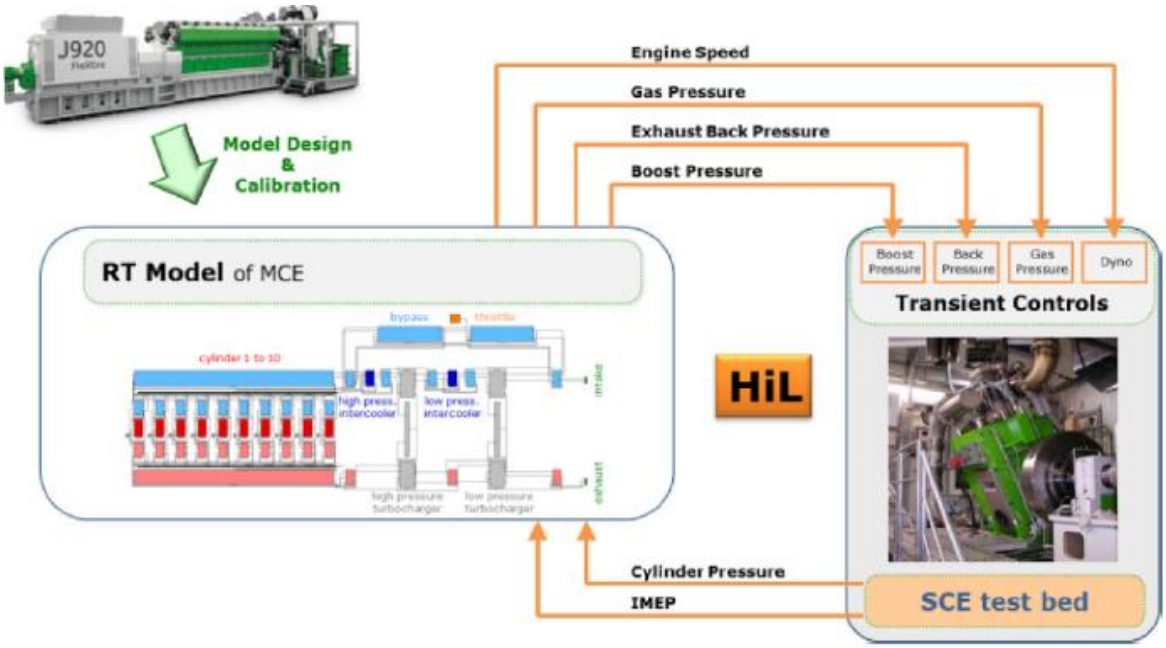


Figure 23: Layout of the transient SCE HIL system [43].

In keeping to transient SCE operation, certain interesting features were also found in other works such as: “A Transient Test System for Single Cylinder Research Engines With Real-Time Simulation of Multi-Cylinder Crankshaft and Intake Manifold Dynamics” and “Simulating Transient Multi Cylinder Engine Gas Exchange Dynamics on a Single-Cylinder Research Engine” [44] [45] by Lahti, L. et al. The research in both papers demonstrates the development of a transient test bed that enables the testing of an SCE that replicates the individual cylinder conditions of an MCE. However, these are not focused on real-time modelling as such. The basis is instead hardware design such as an intake airflow simulator and a dynamometer to mimic MCE behaviour. The models applied in these works are primarily built on look-up tables.

Concluding the literature survey regarding real-time modelling within the ICE industry, it can be observed that there is a significant amount of research and development ongoing, and not only limited to engines. The main focus is regarding virtual testing and calibration using a model-based design approach with various MiL, SiL, and HiL applications. A majority of the different modelling applications use either predictive combustion models or simple mean value models. In this survey, no implementation mentioned the use of combustion feedback from a real engine to the model application. Only a small part of the literature that concerns real-time modelling mentions SCE, and the few that do are in relation to transient testing, where real engine dynamics are modelled and imposed on an SCE to mimic MCE behaviour.

2.4 Simulink Real-Time and Speedgoat

MathWorks and Speedgoat have developed a plug-and-play real-time solution for developing and testing embedded controllers using MATLAB and Simulink (Figure 24). The system consists of two main components: Simulink Real-Time, a solution for real-time simulation and testing, and Speedgoat's real-time target machines with I/O. [46] Simulink Real-Time allows for creating real-

time applications from Simulink models, and it is possible to run these real-time applications on Speedgoat's hardware, which links to the machine being tested via physical I/O lines and communication links. [47] These two components are specifically suited for real-time use cases such as Rapid Control Prototyping (RCP) and hardware-in-the-loop simulation. RCP allows for early testing and iteration of control designs and shortens the time to market of embedded controllers, while Hardware-in-the-loop simulation allows for safe, cost-effective and reproducible testing of embedded controllers in simulated scenarios. The solution enables easy monitoring, locking, and fine-tuning of data during real-time execution and offers integration with a wide range of MathWorks products, such as Simulink test, HDL Coder, Motor Control Blockset, Automotive products, for various application areas. [46] Speedgoat and Simulink Real-Time are engineered to perform seamlessly with each other and provide superior compatibility with Simulink and other MathWorks tools. [47]

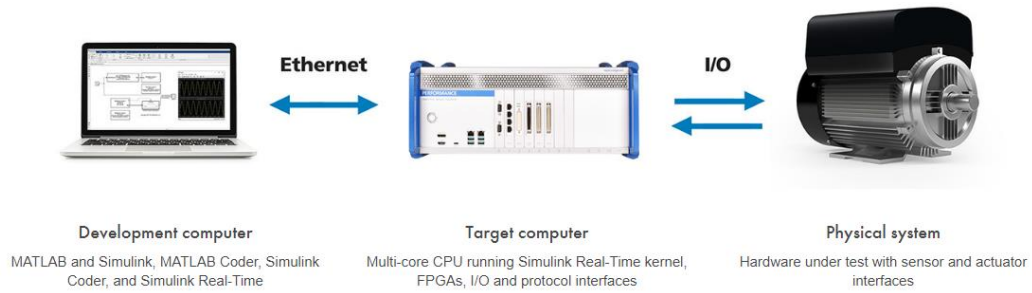


Figure 24: Example of Simulink Real-Time and Speedgoat testing environment [46].

In Wärtsilä, the process of developing control applications for engines also utilises RCP (Figure 25). [48] Simulink Real-Time and Speedgoat are implemented since it is the fastest way of testing prototype software on engines. The engine control software is built up in Simulink, meaning that it can all be simulated on a laptop or desktop PC. It enables the simulation of individual or multiple control systems in conjunction with dynamic engine models. It also allows for control hardware (FPGA) testing before synthesis and, in addition, enables the simulation of all control applications and engine dynamics together using actual engine configuration parameters. [48] The control systems are developed and maintained in Simulink and HDL coder, and the systems run on Speedgoat real-time target PC's with I/O-, communication- and FPGA cards. SiL testing is performed with controller and plant model built in Simulink and can be handled on any computer. MiL is done with both the control software and the plant model compiled on a real-time target PC. With HiL testing, the control models are compiled on one real-time PC and used as an ECU model, while the plant model (engine model used to generate signals) is run on another real-time PC. Once the new engine controls or functionality have been developed and tested, the compiled software is sent to the laboratory engine for implementation [49]

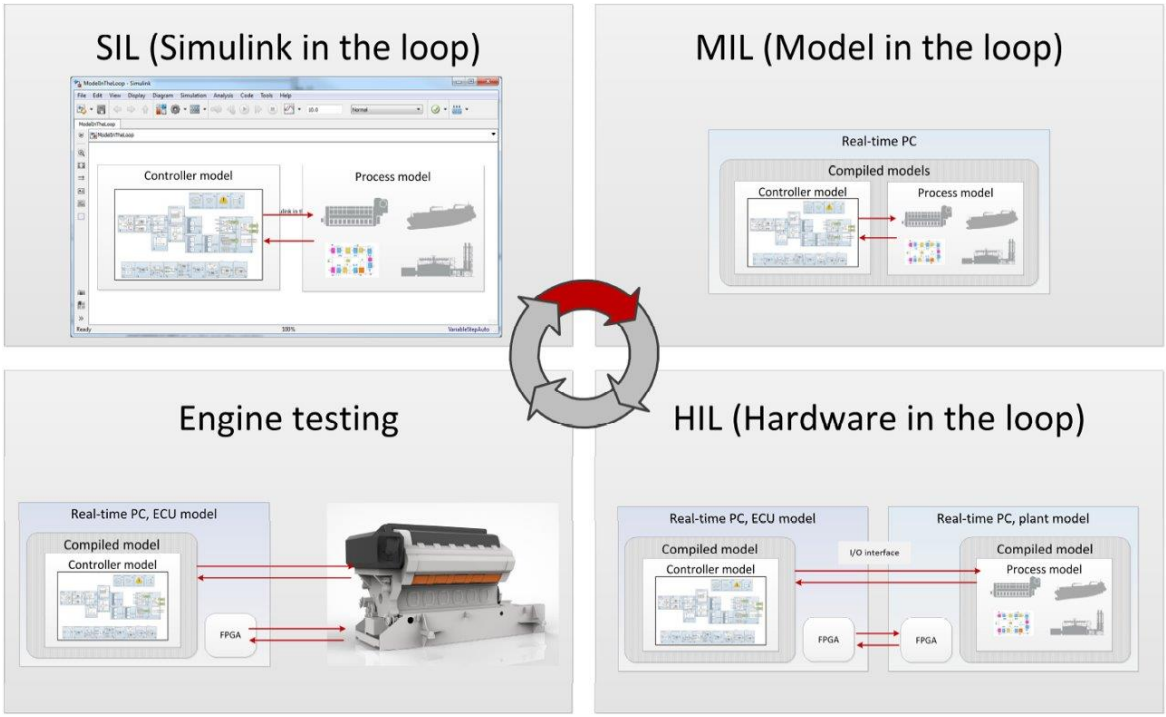


Figure 25: Example of RCP workflow [50].

2.5 User Datagram Protocol (UDP)

This chapter aims to give a short introduction to UDP and its use cases, advantages, and disadvantages (Figure 26). Simulink and Speedgoat support a real-time version of UDP that can be used to connect a target machine to different nodes by either using a dedicated ethernet card or by sharing the ethernet port used for host-target communication. [51] When utilised within Simulink Real-Time, UDP increases the likelihood of a working real-time execution for the application. [52]

User Datagram Protocol (UDP) is a transport layer protocol within the Internet Protocol (IP) that enables communication without needing a dedicated end-to-end connection. Its minimalistic and lightweight nature makes it ideal for real-time applications with high performance and low latency needs. One of the critical advantages of UDP is its ease of use. Unlike TCP, which establishes a full duplex connection and offers error checking and flow control, UDP does not ensure delivery or maintain the order of messages. This reduces the need for state information or lost packet retransmission, leading to higher efficiency and lower overhead. However, the absence of error checking and flow control also increases the likelihood of packet loss and may not be suitable for applications that demand reliable data transfer. To overcome this, UDP applications can implement error-checking and flow-control mechanisms. [53] [54]

UDP allows for finer control over what data is sent and when; as soon as data is passed from the application process, it is packaged and immediately sent to the network layer. On the other hand, TCP has a control mechanism for congestion that may slow down or resend data, making it less suitable for real-time implementations with a minimum sending rate requirement and which can

tolerate some data loss. These real-time applications can use UDP and implement additional functionality beyond UDP's basic segment-delivery service. UDP does not require a connection establishment, meaning it does not introduce any delay before transferring data. For this reason, DNS applications run over UDP as it would be slower with TCP. However, if reliability is critical, e.g., for web pages with text, it is better to use TCP instead of UDP. The connection-establishment delay in TCP is a significant reason for delays when for example, loading web pages. [55]

Characteristic / Description	UDP	TCP
General Description	Simple, high-speed, low-functionality "wrapper" that interfaces applications to the network layer and does little else.	Full-featured protocol that allows applications to send data reliably without worrying about network layer issues.
Protocol Connection Setup	Connectionless; data is sent without setup.	Connection-oriented; connection must be established prior to transmission.
Data Interface To Application	Message-based; data is sent in discrete packages by the application.	Stream-based; data is sent by the application with no particular structure.
Reliability and Acknowledgments	Unreliable, best-effort delivery without acknowledgments.	Reliable delivery of messages; all data is acknowledged.
Retransmissions	Not performed. Application must detect lost data and retransmit if needed.	Delivery of all data is managed, and lost data is retransmitted automatically.
Features Provided to Manage Flow of Data	None	Flow control using sliding windows; window size adjustment heuristics; congestion avoidance algorithms.
Overhead	Very low	Low, but higher than UDP
Transmission Speed	Very high	High, but not as high as UDP
Data Quantity Suitability	Small to moderate amounts of data (up to a few hundred bytes)	Small to very large amounts of data (up to gigabytes)
Types of Applications That Use The Protocol	Applications where data delivery speed matters more than completeness, where small amounts of data are sent; or where multicast/broadcast are used.	Most protocols and applications sending data that must be received reliably, including most file and message transfer protocols.
Well-Known Applications and Protocols	Multimedia applications, DNS, BOOTP, DHCP, TFTP, SNMP, RIP, NFS (early versions)	FTP, Telnet, SMTP, DNS, HTTP, POP, NNTP, IMAP, BGP, IRC, NFS (later versions)

Figure 26: Comparison between UDP and TCP [53].

3 Method

This chapter will present the implementation of the thesis. The first subchapter concerns combustion feedback and the multiple possibilities there are to implement a combustion profile within the MCE GT-POWER model. The various methods and their advantages and disadvantages will be discussed in detail. The following chapter concerns the GT-model conversion to an xRT-compatible model. The chapter discusses the different challenges and peculiarities that xRT presents and how to handle them. It also demonstrates the real-time capability of the finalised xRT model. The final subchapter shows the Simulink Real-Time & Speedgoat + GT-power environment and explains how the real-time combustion feedback from the SCE to the MCE model was achieved. The cylinder pressure vector processing, the heat release calculation, and the method for sending the combustion profile to GT will be presented in detail.

3.1 Combustion feedback and modelling

It was noted that none of the literature surveyed in this thesis mentioned the use of real-time combustion feedback from a physical engine for combustion modelling. Depending on the application, in most cases, either predictive combustion models or even simpler mean value models, where combustion is not modelled at all but instead mapped, are utilised. It is likely that, in many cases, there is no real engine combustion feedback available or that it is not practical to achieve the feedback from a real engine. In other cases, it is not necessary for the goals and objectives that have been established.

In GT-POWER, there are two types of combustion models: non-predictive and predictive [8]. Non-predictive models impose a burn rate and do not depend on in-cylinder conditions for the characterisation of combustion and emission parameters. They are helpful in evaluating concepts that do not impact burn rate characteristics but may need to be more accurate for studying phenomena such as EGR and injection timing. Predictive models calculate the burn rate for each cycle based on in-cylinder conditions, leading to longer simulation times, but they are generally more accurate when used to study parameters that impact burn rate, e.g., variable injection timings, EGR, and injection profiles.

Predictive models require calibration against known reference data to produce accurate results [8], and the process may be extensive. This is not an issue if the engine concept and model are kept relatively constant, e.g., virtual ECU calibration for an automotive production engine. However, on the flexible SCE research platform, this becomes a limitation when hardware changes can occur daily, and different combustion concepts and fuels are tested with a high frequency. For this reason, it is not practical to implement a predictive combustion model. Additionally, as the plan is to run the SCE in parallel with the modelled MCE, the actual engine combustion is already available for all data points and cycles. In essence, the basis is a non-predictive model, but each simulation is based on real combustion rates taken from the physical SCE, giving the highest possible accuracy. The following sub-chapters will go through different methods of imposing combustion feedback from the SCE into the MCE model in order to simulate combustion correctly. Several options were considered and tested before the final selection was made.

3.1.1 Impose reference burn rates

The first alternative that was considered was to use already calculated burn rates from reference measurements on the SCE. This is generally how stand-alone simulation models that don't utilise predictive modelling are operated within Wärtsilä. A reference cylinder pressure- or heat-release curve, measured from a real engine, is imposed as a profile array within GT-POWER "EngCylinder" combustion object and used for the in-cylinder combustion model (Figure 27).

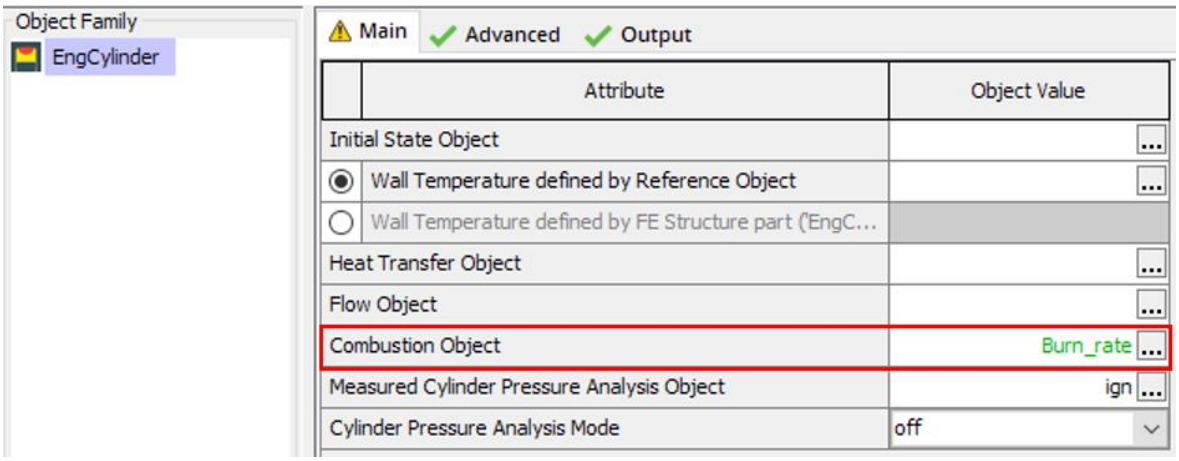


Figure 27: A burn rate added as a combustion object in the cylinder template.

The burn rate is crank angle based and imposed as either **rate,** which indicates that the burn rate profile is entered as an instantaneous profile or **cumulative,** indicating that it is an integrated array (Figure 28). [8]

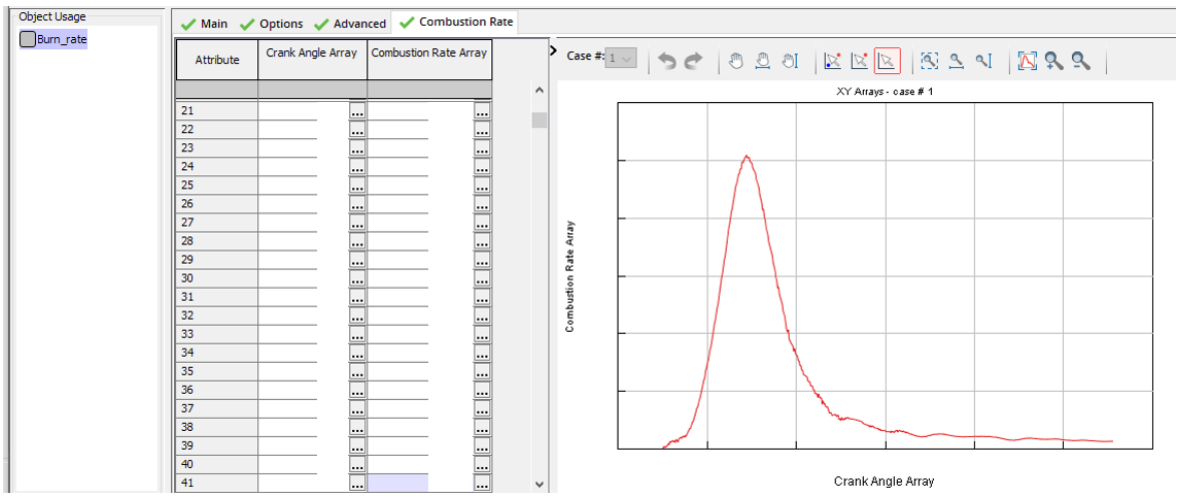


Figure 28: Example of a combustion rate vs crank angle.

The benefit of using burn rates from reference measurements is that it is a very straightforward and quick way to get started. As it is a non-predictive model that uses "pre-made" burn rates, the model runs very fast and reaching real-time capability is quite simple. The drawback, on the other hand, is that the reference combustion is only valid for a given point with a particular engine setup, load, and tuning. As soon as changes are made to the engine hardware setup or load or tuning parameters, it is no longer correct. Within GT-POWER, it is possible to tabulate several burn rates to make, e.g., load, speed, or fuel-dependent burn rates that it can interpolate between. However, this quickly becomes an impossible task when one must consider several hundreds of different combinations of hardware, settings, and operating points.

3.1.2 Impose Spark-Ignition (SI) Wiebe parameters

The second option for imposing the combustion to the MCE model involves the SI Wiebe combustion model. The basis is a Wiebe function that approximates the burn rate of a typical SI engine as a function of crank angle. The Wiebe function is a mathematical expression defined by a set of parameters that are calibrated to match the burn rate data from an engine during the combustion process. [8] Using three parameters, Anchor Angle, Duration and Wiebe Exponent, it is possible to create a burn rate that will match a measured burn rate (Figure 29, Figure 30).

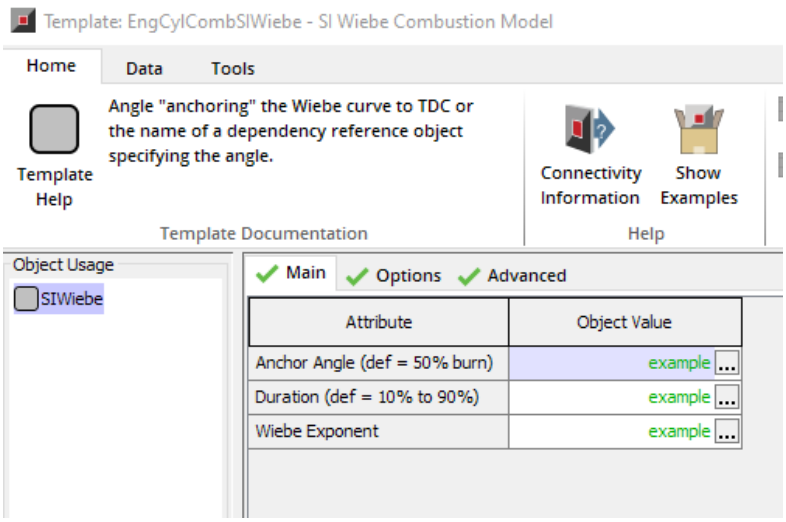


Figure 29: SI Wiebe combustion model template.

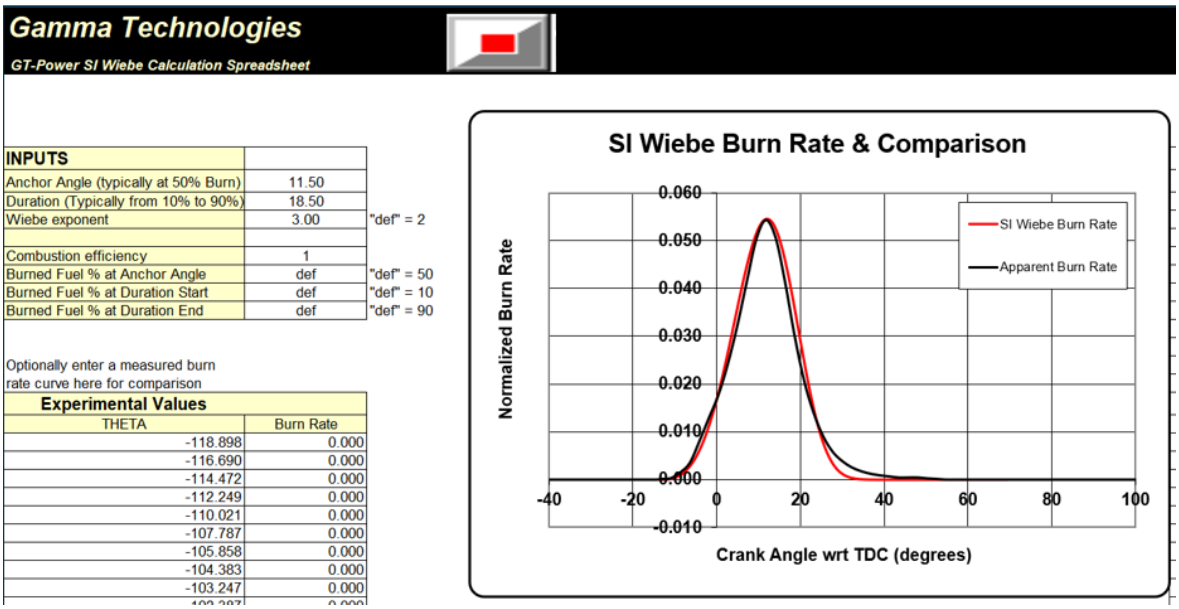


Figure 30: SI Wiebe example, estimated burn rate in red and measured burn rate in black.

The SI Wiebe combustion model is non-predictive and can execute very fast due to its simplicity, although the accuracy is not as good as using a calculated burn rate. It is also possible to impose the various controlling parameters during a simulation, making it suitable for transient modelling. Two of the parameters, Anchor Angle and Duration, can be easily imposed directly from measured values on the SCE. The Wiebe exponent is not a measured value and needs to be estimated with good precision, which is challenging and can lead to poor accuracy. One possibility that has yet to be explored is the use of neural networks (NN) to estimate the Wiebe exponent, in a sense making a predictive Wiebe exponent based on various inputs. This would require large amounts of test data to validate the NNs for a wide variety of operating points.

3.1.3 Impose measured heat release rate for each point / Steady-state

The third option considers the possibility of imposing a measured heat release rate (HRR) profile for each operating point on the SCE before the GT MCE model is started. The heat release rate is calculated in real-time on three separate data acquisition systems in the SCE testing environment. The idea is to calculate a 100-cycle averaged HRR profile during steady-state operation from any one of the acquisition systems and to insert it into GT-POWER through an external file such as ASCII, Excel, or MAT (Figure 31).

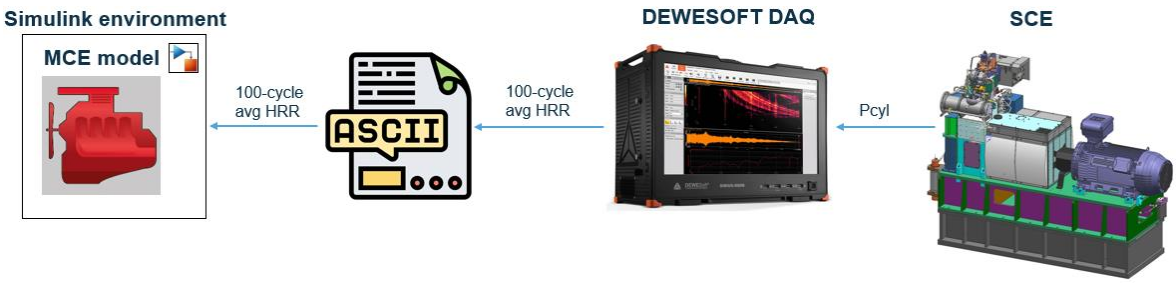


Figure 31: Steady-state method (start/stop).

With this solution, the accuracy will be acceptable at all steady-state operating points. On the downside, this solution cannot be used for transient simulation since only one HRR profile from a steady state point can be used in each simulation case, as the profile cannot be modified once the simulation is running. This means that the model must be stopped after each steady state point and have a new HRR profile imposed for the next case before it can be started again. This Start/Stop method may be suitable for steady-state operation if there are no real-time criteria. The model itself will still be real-time capable, but the start-stop process of the simulation process can range from one to two minutes.

3.1.4 Sequential cylinder pressure analysis model

This alternative describes the creation of an additional GT-POWER model that would be run sequentially with the SCE and existing MCE model. Cylinder pressure data would be fed from the SCE to a “Cylinder Pressure Only” (CPOA) model or a “Three Pressure Analysis” (TPA) model that runs until it converges and calculates a burn rate. The calculated burn rate would then be imposed in the MCE model, which in turn creates the output data of interest (Figure 32).

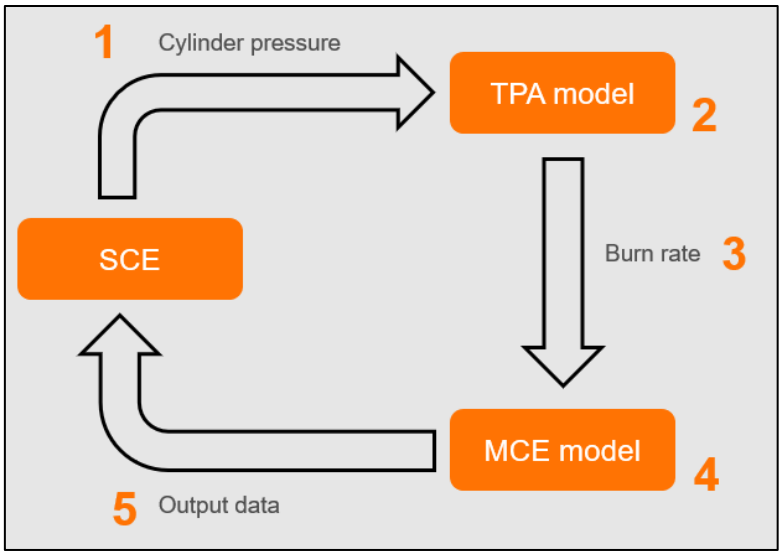


Figure 32: Sequential cylinder pressure analysis model.

This is likely to be the most accurate model for steady-state points since the profile that is imposed into the MCE model is a burn rate rather than a heat release rate. The burn rate and heat release are somewhat different. Burn rate refers to the initiation of chemical reactions between air and fuel, but since the fuel does not immediately fully combust, the full heat is not immediately released. Heat release rate, on the other hand, deals with the rate at which fuel is converted into thermal energy, including all conversions. [8] It is delayed compared to the burn rate. If heat release is used as input, GT-Power will treat it as burn rate, resulting in a slower final heat release than intended.

To determine the burn rate of an engine, “reverse run” calculations are done by inputting the measured cylinder pressure and iterating the amount of fuel transferred from the unburned to the burned zone until the cylinder pressure matches the input. These calculations use the same equations as "forward run" calculations, which use the burn rate as the input and the cylinder pressure as a result. Estimating a burn rate from cylinder pressure can be achieved in two ways within GT-POWER, either with a CPOA model or a TPA model. [8]

There is currently no way of imposing cylinder pressure into a running model within GT-POWER. Meaning that the model needs to be stopped and started again each time a new burn rate is imposed. Since this alternative incorporates two different GT models that both need to start, converge and stop before the loop can proceed, the run time will be quite long. A workflow proposal from Gamma Technologies has been made to enable cylinder pressure feedback to be imposed within a running model, but this still requires software development.

3.1.5 Impose instant cumulative burned fuel fraction

The last alternative considered in this thesis is the possibility of imposing an instant "cumulative burned fuel fraction" into the cylinder object within the engine model. A combustion profile can be imposed using signals from, e.g., Simulink via an "ActuatorConn" object to actuate the combustion in a cylinder (Figure 33). This will override any combustion attributes already set up in the cylinder object. A crank angle degree signal is taken from the GT model and sent to a dynamic look-up table within Simulink. The heat release rate is calculated in Simulink from measured cylinder pressure and sent to the same dynamic look-up table. Depending on the crank angle signal from GT, a corresponding point from the calculated HRR is sent back to the GT model and imposed into the cylinder object for each time step of the simulation (Figure 34).

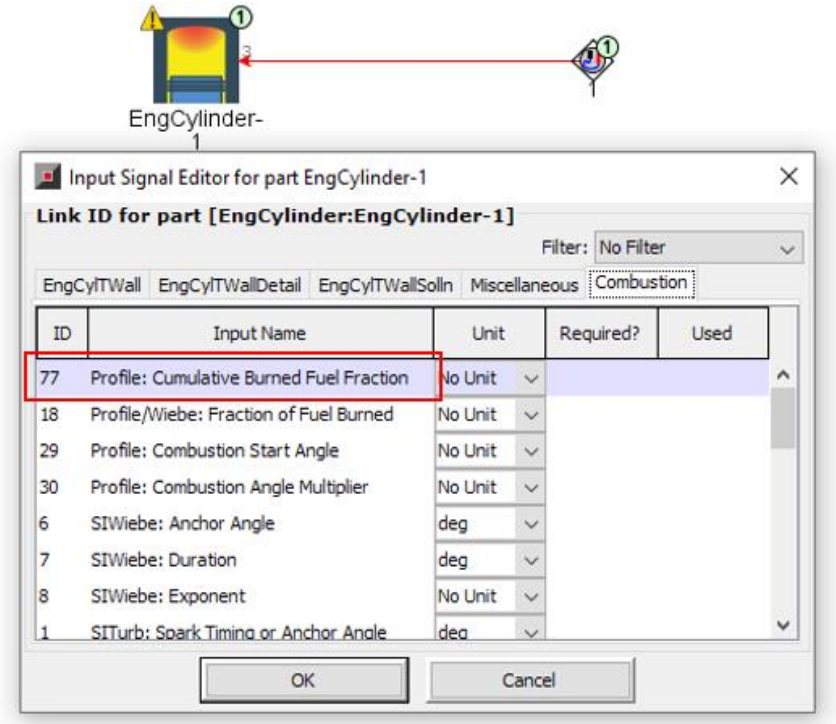


Figure 33: Combustion profile actuation.

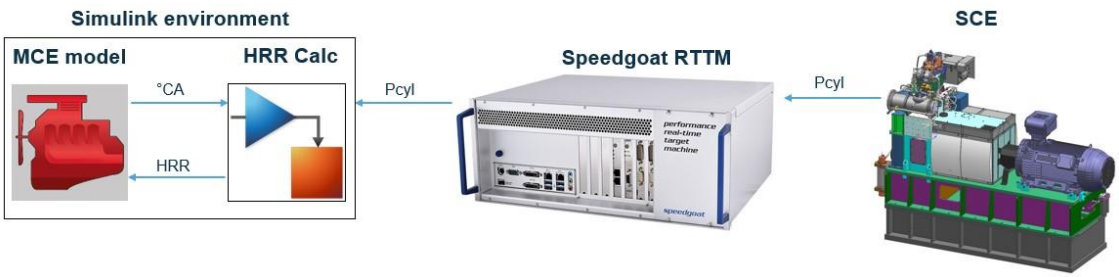


Figure 34: Measured cylinder pressure to imposed HRR workflow.

The main advantage of this approach is the possibility to impose a combustion profile while the simulation is running and that the combustion profile is based on actual SCE measurements, making

it the most accurate option for real-time operation and transient modelling. Hence better than tabulated reference combustion profiles or SI Wiebe models. It is also as accurate as the start/stops steady-state model with measured HRR.

One challenge with this setup is the requirement that the combustion profile is imposed into the model point–by–point as a cumulative and normalised profile with a minimum value of 0 and a maximum value of 1 and always monotonically increasing. This means that the Simulink model needs to be run with the same time-step as the GT model in order to avoid stability issues and achieve an acceptable accuracy of the combustion profile. Good enough accuracy would mean a combustion profile with a resolution no larger than 1°CA, preferably in the range of 0.2-0.5°CA. This corresponds to timesteps in the order of 30-150µs at the nominal SCE speed of 1000rpm. The fast-running GT MCE model in itself is capable of real-time execution speeds with these small time steps. However, running the Simulink model with the same time steps makes the combined model very slow. To reach real-time capability with a very small Simulink time-step, the MCE model needs to be converted to an xRT-model, and an xRT-license needs to be used.

3.2 GT xRT model conversion and comparison to baseline

In order to reach the real-time capability of the MCE model even when running Simulink with very small time steps, it was decided to convert the existing fast-running model into an xRT model and utilise an xRT-license. As the MCE model had already been converted and simplified from a detailed model into an FRM, it was not necessary to further reduce the fidelity of the model to reach real-time with the xRT version. Instead, the focus could be put on handling all the peculiarities and special requirements when it comes to xRT conversion. The first part concerns making the model compatible with xRT by changing, removing, or adding certain features to the model. As xRT is dedicated to real-time HiL applications, there are a few options that differ from the original FRM model and some functions that xRT does not support.

One example is “RLTDependence”, which is a template for imposing the value of an attribute as a function of a wireless signal or result (RLT) variable rather than using a constant value (Figure 35). The input signal triggers an update to the dependent variable. In periodic simulations, RLT variables are updated every cycle, while in continuous simulations, they are updated at a time increment that the user can specify. Wireless input signals are updated at every time step. [56] Within xRT, the template can only be used for a select quantity of attributes. It is, however, possible to achieve RLTDependence for almost any attribute that can be actuated using a ”SignalGenerator” object, which is supported. The SignalGenerator object creates an output signal that the user has imposed. The signal may be created through several options, such as a constant or equation etc. (Figure 36). In the existing FRM model, some examples of attributes utilising “RLTDependence” that were not allowed were the cylinder convection multiplier and the valve-to-seat heat transfer coefficient. The attributes were either set to constant values or actuated through the SignalGenerator object.

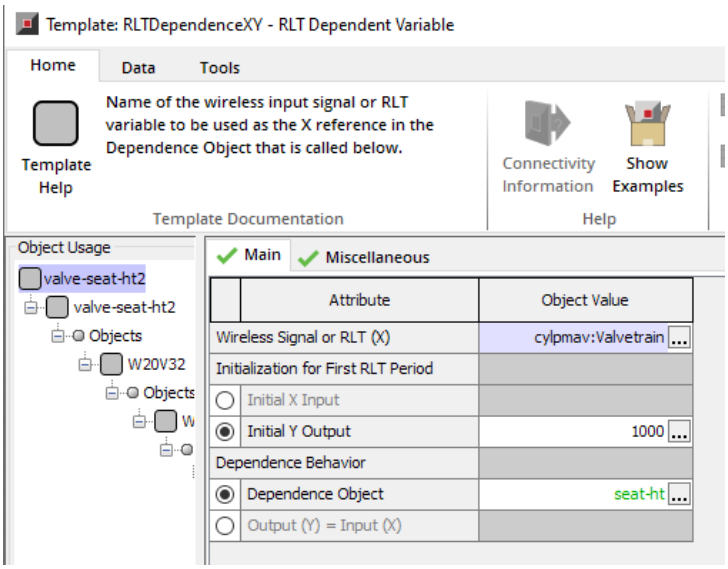


Figure 35: Example of RLTDependence object: Valve to seat heat transfer coefficient as a function of cylinder pressure.

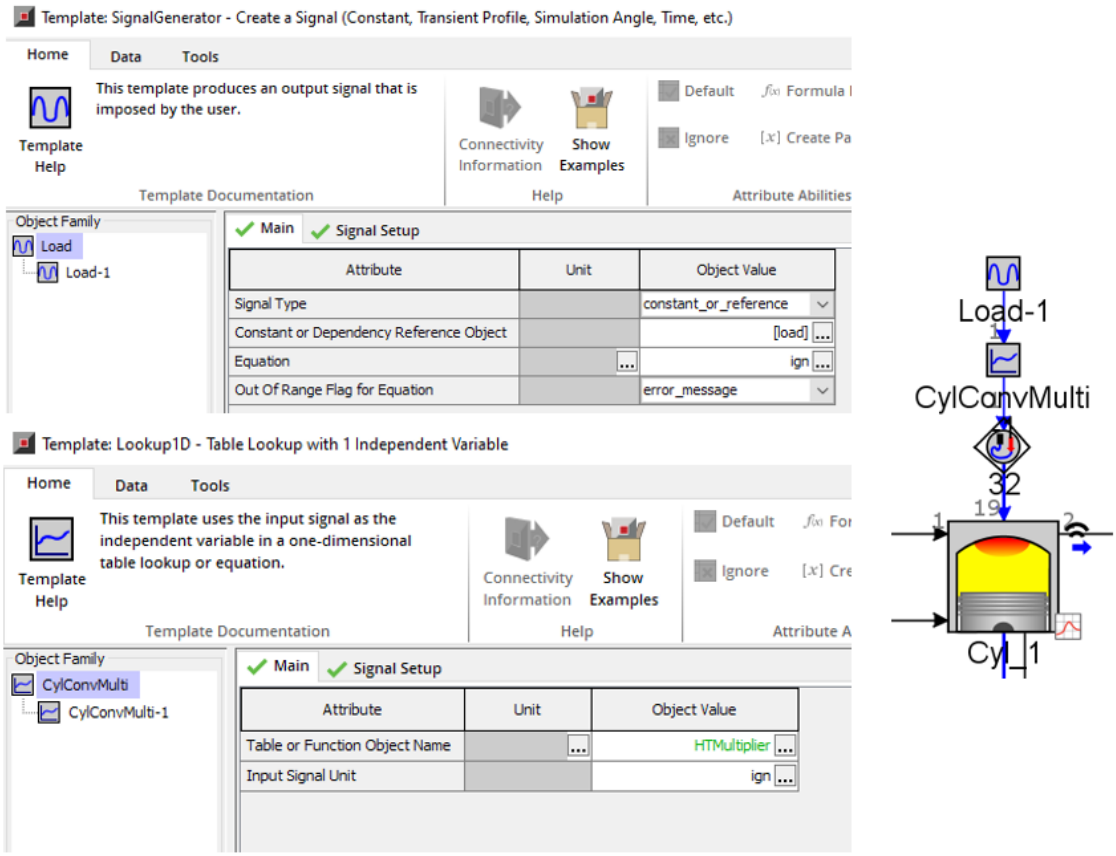


Figure 36: Example of actuated heat transfer multiplier using signals instead of RLTDependence.

Another difference within the xRT model is the "Cylinder Copying Option" (Figure 37). This capability allows replicating the cylinder pressure from one cylinder (the lead) to one or more other cylinders (the copies) by either computing the required burn rate based on trapped conditions and applying it to the copy cylinders or by directly applying the cylinder pressure. [56] It shortens the simulation time as the predictive combustion model calculations are only performed in the lead

cylinder. This lead/copy- feature is intended for use with FRM and real-time models, not detailed engine models where cylinder variations are crucial. The initial FRM model uses an option called “copy-RT-full-v2017”, which means that both the combustion and valve mass flow rates from the lead cylinder will be replicated on the copies. xRT does, however, not support the “copy-RT-fullv2017” option but only the “copy-RT-partial”. This should, anyway, not be needed since the xRT model is much faster than the FRM.

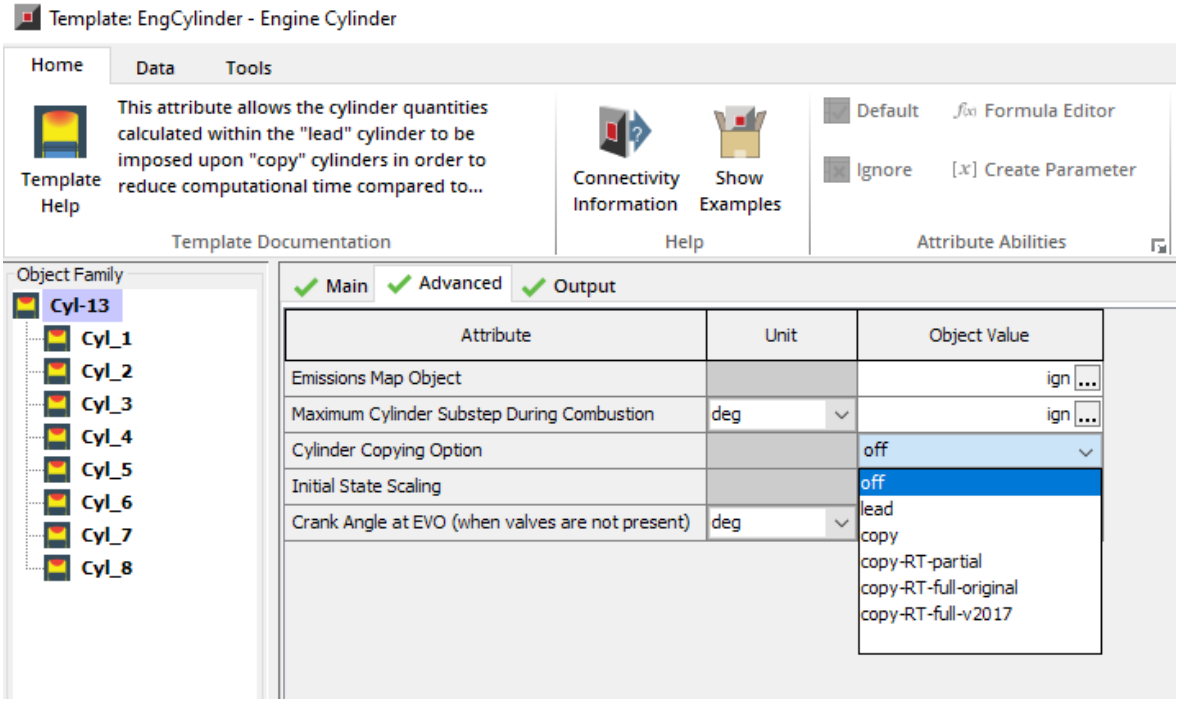


Figure 37: Cylinder copy.

The “MathEquation” template allows the user to input mathematical expressions to perform calculations on one or more input signals and generate an output signal (Figure 38) [56]. When using an xRT-license, it is possible to choose between two calculation methods: “Real-Time” and “Interpreted (slow)”. The real-time method uses compiled equations to achieve quick execution. To enable the Real-Time method, a C compiler is required to be specified within GT-POWER (Figure 39). It is possible to automatically compile the MathEquation templates every time the simulation is started or to use a pre-compiled library (e.g., gtmathequations.dll) to save time when starting the model.

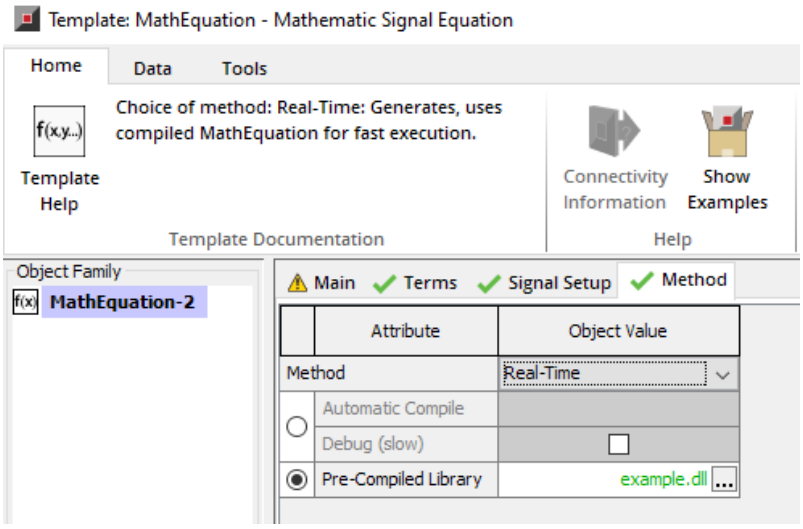


Figure 38: MathEquation template setup for real-time modelling.

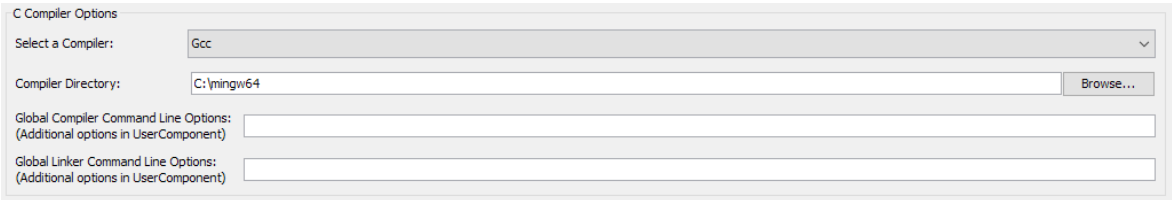


Figure 39: C compiler options for MathEquation.

Other major differences and factors to consider include:

- A fixed time step is required for GT-POWER xRT. As xRT is a dedicated HiL solution, the model is always required to run with a fixed time step. Additionally, in order to impose an instant cumulative heat release rate, the time step needs to be fixed and set to the same values in both GT and Simulink. The dynamic time-step logic has been removed from the xRT solver since it would only contribute to extra overhead. The time step can be determined by executing the model with a normal GT-POWER solver and applying the resulting time step to the xRT model.

- GT-POWER xRT does not support any monitor output from the running model, and it is one of the reasons why it is so fast. Any signal that needs to be monitored while the simulation is running (i.e., life) needs to be sent as a signal via the SimulinkHarness so that it can be monitored through Simulink instead. The drawback is that Simulink generally causes more overhead compared to GT when running at very small time steps.

- Array templates for coefficients in various GT Objects, e.g., Valves and valve flow coefficients, are not supported in GT-POWER xRT. Instead, it is possible to utilise parameterised objects to allow the flexibility of changing coefficients.

- "IfThenElse" object is not supporting the use of parameters from the Case Setup. The parameters that are to be used can be specified in "SignalGenerators" instead.

- "Air" Object for reference stoichiometric air-to-fuel ratio must be specified since RLT quantities for air-to-fuel ratio and lambda from the cylinder are replaced with air-fuel ratio and lambda signal from the exhaust manifold components.

The second part of the xRT conversion is customising the output from the model. The largest difference between GT-POWER and GT-POWER xRT is the fact that the only location where RLTs are calculated is in the SimulinkHarness. xRT provides all standard engine-related performance measures as signals instead. [11] Meaning that specific RLTs used in a normal GT model need to be converted to signal or may require to be recalculated. In this case, as the turbocharging system uses mass flow averaged quantities, new calculations are needed in the xRT-model to replicate the RLT output from the FRM (Figure 40, Figure 41).

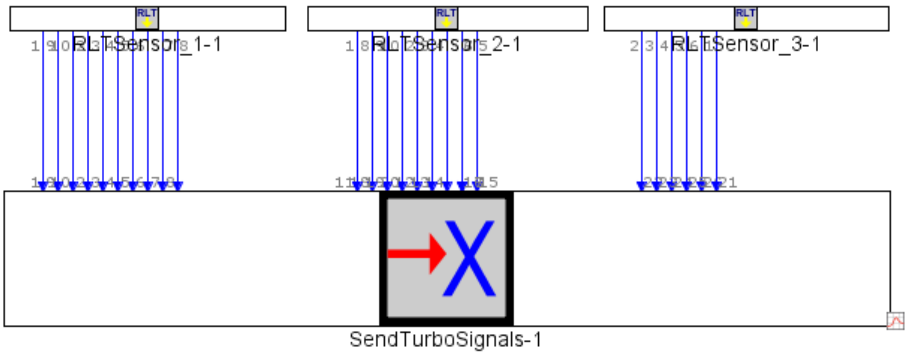


Figure 40: Turbocharging system-related RLTs in the FRM.

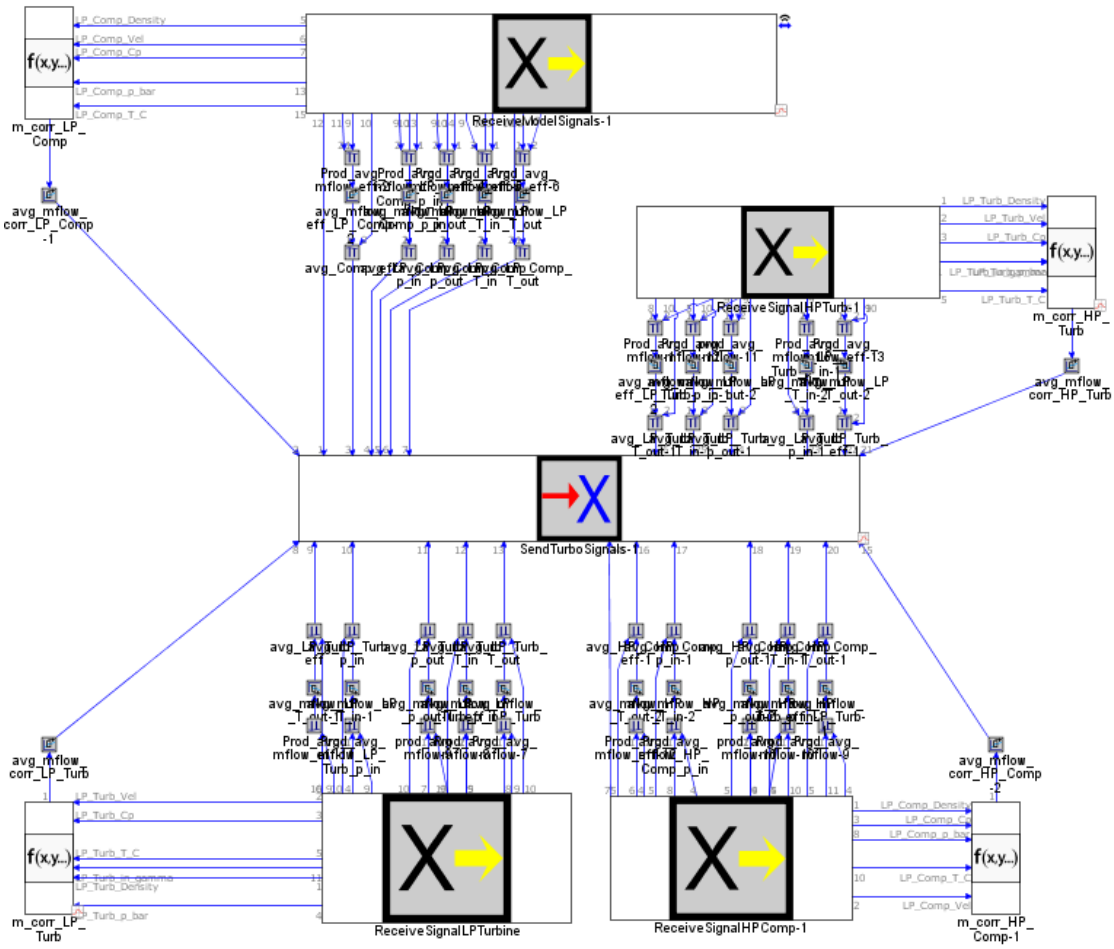


Figure 41: Turbocharging system-related signals and calculations in the xRT-model.

After the conversion was done, the resulting xRT-model was capable of a real-time factor of 0.1. This is close to ten times faster than the baseline FRM. The resulting model generally shows good agreement with the FRM, although some small-scale differences can be observed, mainly in the TC-related mass flow values. A calibration of the model might be required to mitigate these differences. During the first run of the xRT-model, it was noted that some attributes are not converging, e.g., IMEP (load). In Figure 42, it can be observed that the initial oscillation disappears after a few hundred simulation cycles but reappears later. The noted reason is the large time steps of the xRT model that causes flow instability. A "Stable Flow Through Valves and Throttles Multiplier" can be adjusted to dampen the flow, primarily over the exhaust valves, and make the simulation more stable (Figure 43, Figure 44). The multiplier is likely not needed when running small time steps for real-time combustion feedback.

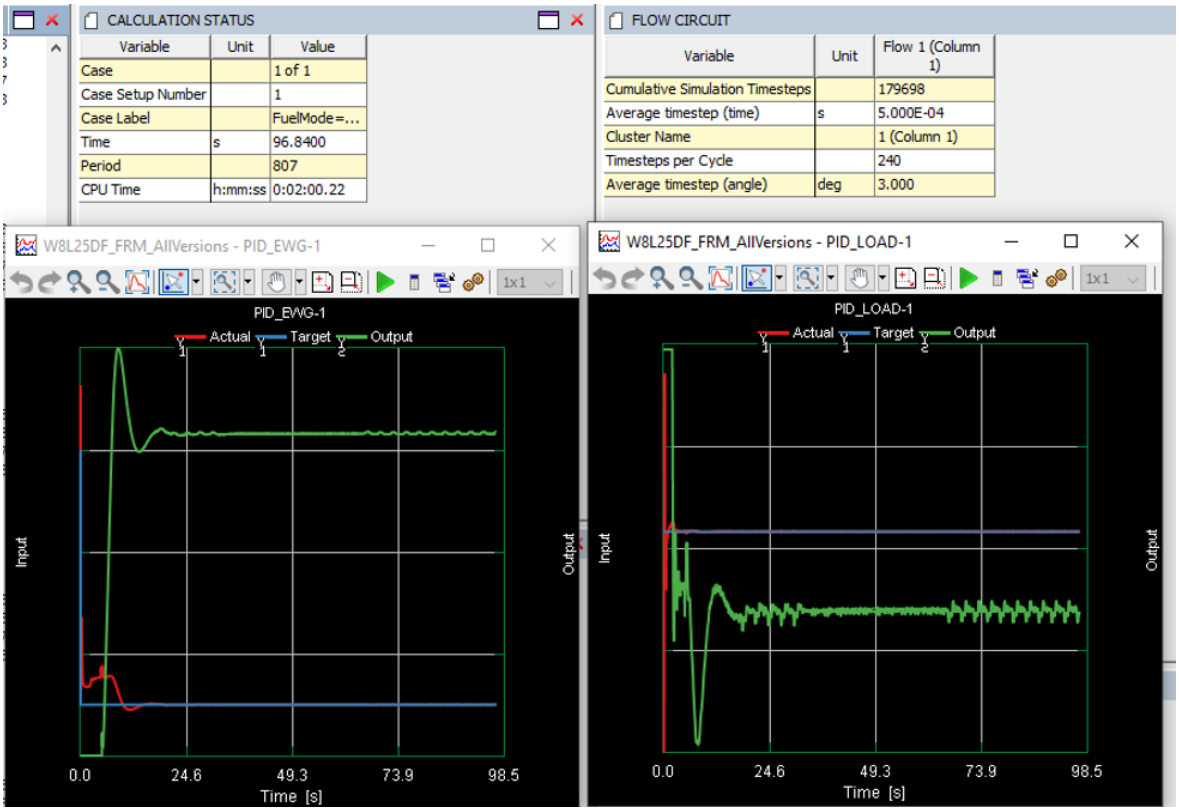


Figure 42: xRT model instability with large time steps (0.5 ms).

Figure 43: Flow stability multiplier.

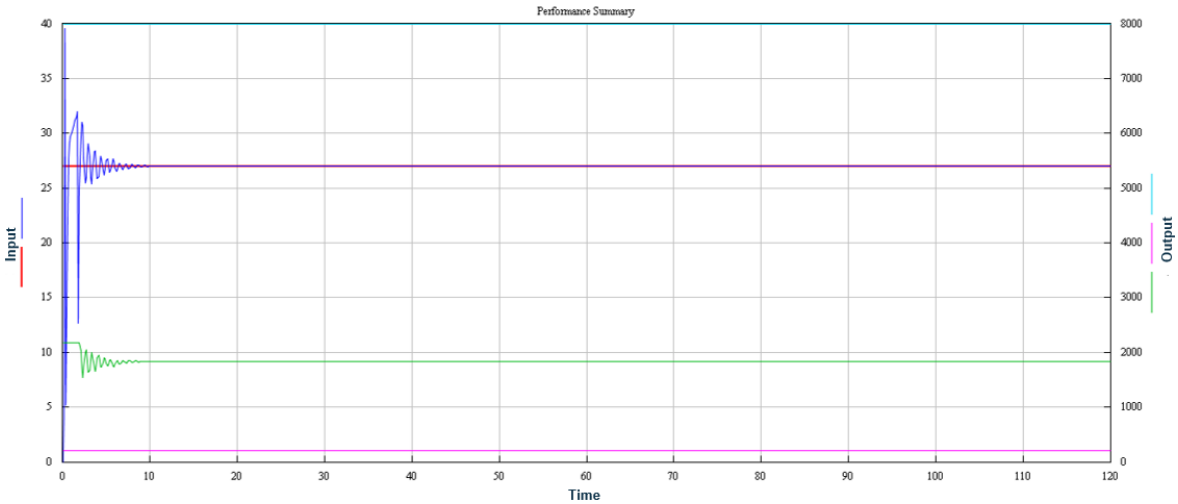


Figure 44: xRT model stability improved with the flow stability multiplier.

3.3 Simulink Real-Time & Speedgoat + Simulink & GT-POWER co-simulation

The following subchapters go through the process of sending a measured combustion profile from the SCE to the MCE model in real time. The GT-POWER MCE model is incorporated into an existing Simulink environment through co-simulation. The GT model is compiled as an s-function and runs within Simulink. Signals to and from the GT model are handled through a “SimulinkHarness” component within GT-POWER. The input signals to the GT model consist of, among other things, ambient conditions, SCE speed and load etc. Output signals from the GT model back to Simulink and the SCE contain TC system parameters such as flows and temperatures, simulated cylinder pressures and more.

The SCE also incorporates a Simulink Real-Time and Speedgoat environment (Real-Time Target Machine) for rapid control prototyping and HiL testing. It allows for the creation, control and monitoring of real-time applications that run on the real-time target computer via Simulink models. To achieve below micro-second control rates for high-frequency signal acquisition and processing, the Speedgoat target computer includes Field Programmable Gate Array (FPGA) I/O modules that are used to measure, analyse and calculate cylinder pressure, engine speed and phase. The FPGA is also programmed using Simulink.

For this specific application, a Simulink module built up of several subroutines is created. Among others, a method to handle the UDP communication between Speedgoat and the Simulink – GT-POWER simulation laptop for cylinder pressure measurements from the SCE. Further models were made to build up the cylinder pressure vector, calculate the heat release and send the heat release profile to the GT model.

3.3.1 Real-time cylinder pressure vector via UDP

The application running on the Speedgoat target computer includes a log function with the possibility to send data over real-time UDP. The logged data that is sent consists of a vector with 1x1444 elements that is split into four packets as single precision with 1x361 elements for transmission over UDP. As the UDP protocol sends data in the uint8 (8-bit unsigned integer) format, the vector needs to be reformatted in a “Byte Pack” block that converts the input signal for UDP transmission before connecting to a “UDP Send block”. [57] The Speedgoat target computer is connected to the Simulink – GT-POWER simulation laptop through a dedicated point-to-point ethernet connection (Figure 45).

The logged data consists of a cylinder pressure vector with 1x1430 elements, corresponding to 715 crank angle degrees with 0.5-degree resolution. The final 5 degrees of the 720-degree cycle is extrapolated as the variation in-cylinder pressure at the scavenging top dead centre (TDC) is small. For the heat release calculation, only the pressure from combustion, TDC +-100°CA, is used. Further, there is a cycle counter, amounting to one element and another element, that notes the packet number since the vector is split into four packets for sending over UDP. The last 12 elements of the vector are utilised for valve timings and various other measurements.

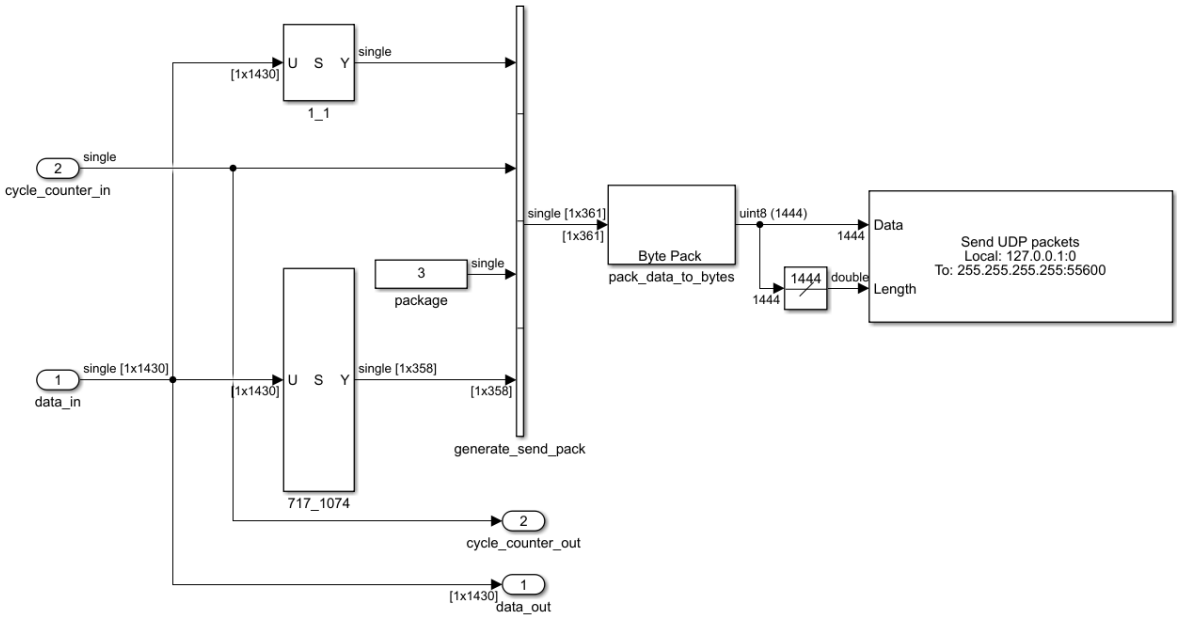


Figure 45: "UDP send" example.

On the Simulink - GT-POWER simulation laptop side, there is a corresponding "UDP receive" and "Byte Unpack" block to convert the data back into its four single precision 1x361 packets. The cylinder pressure is later put together and used for heat release calculation (Figure 46).

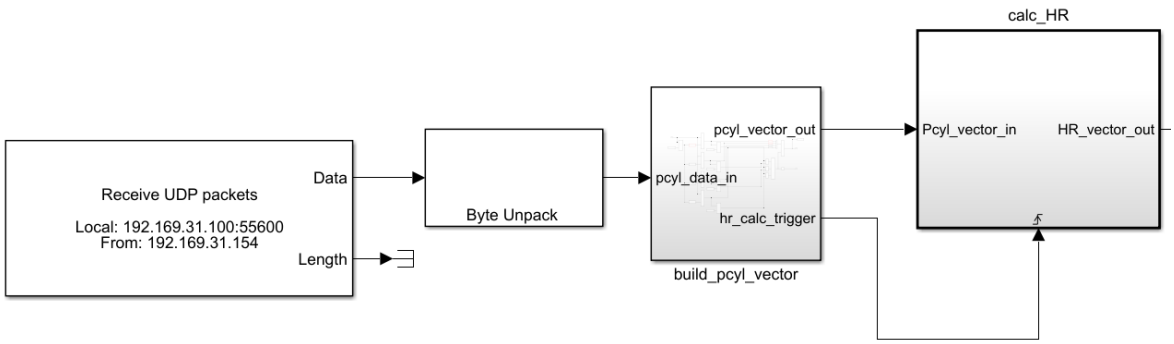


Figure 46: "UDP Receive", vector build up and heat release calculation.

A "build_pcyl_vector" function was constructed to put together the four-vector packets into one continuous cylinder pressure vector. See Figure 47, Figure 48 and Figure 49. The first "Selector" block looks at the packet element of the four packets to ensure that they are built in the correct order. A second "Selector" block looks at the cycle counter and only sends the complete cylinder pressure vector for HRR calculation if all packets are coming from the same cycle. It also acts as a trigger signal for the HRR calculation subroutine. In addition, all elements in the vector that are not cylinder pressure are removed, utilising a "Selector" block and an "Assignment" block.

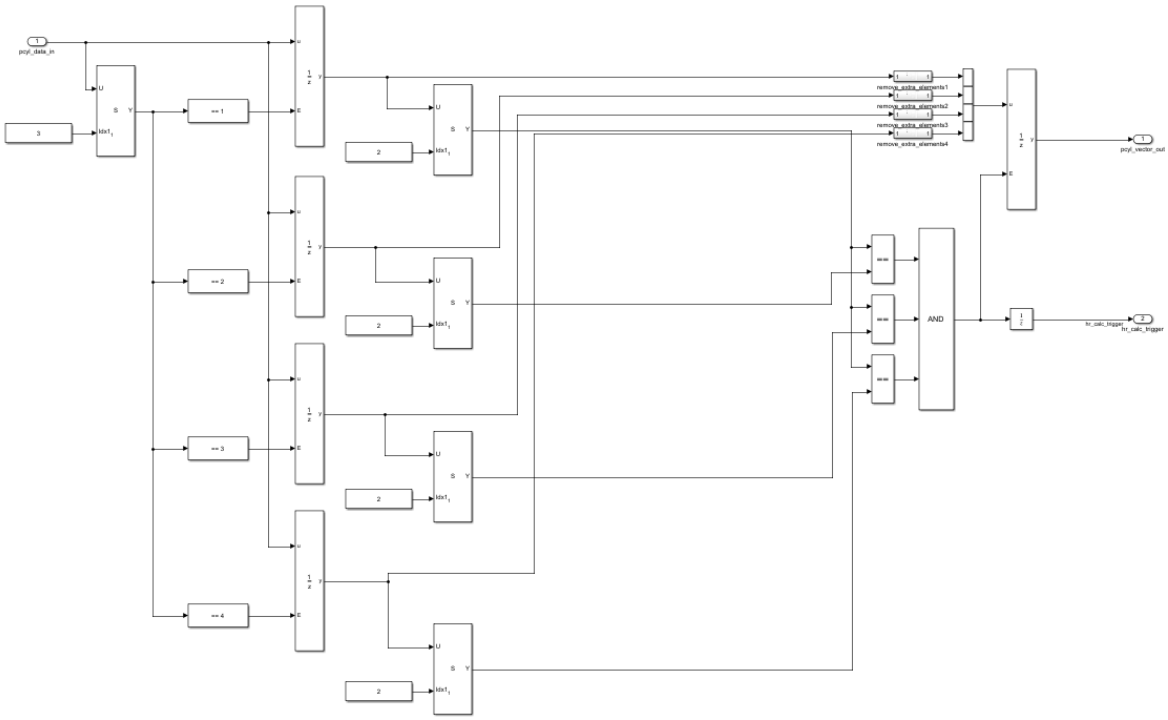


Figure 47: "Build_pcyl_vector": Function overview.

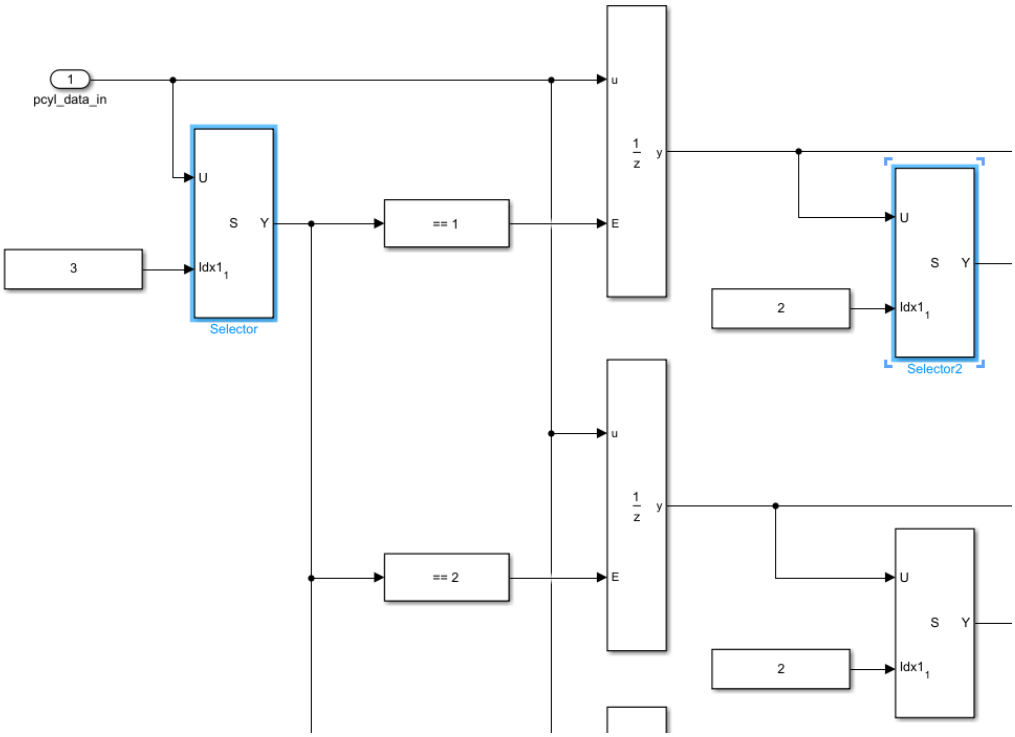


Figure 48: "Build_pcyl_vector": Vector packet and cycle selectors.

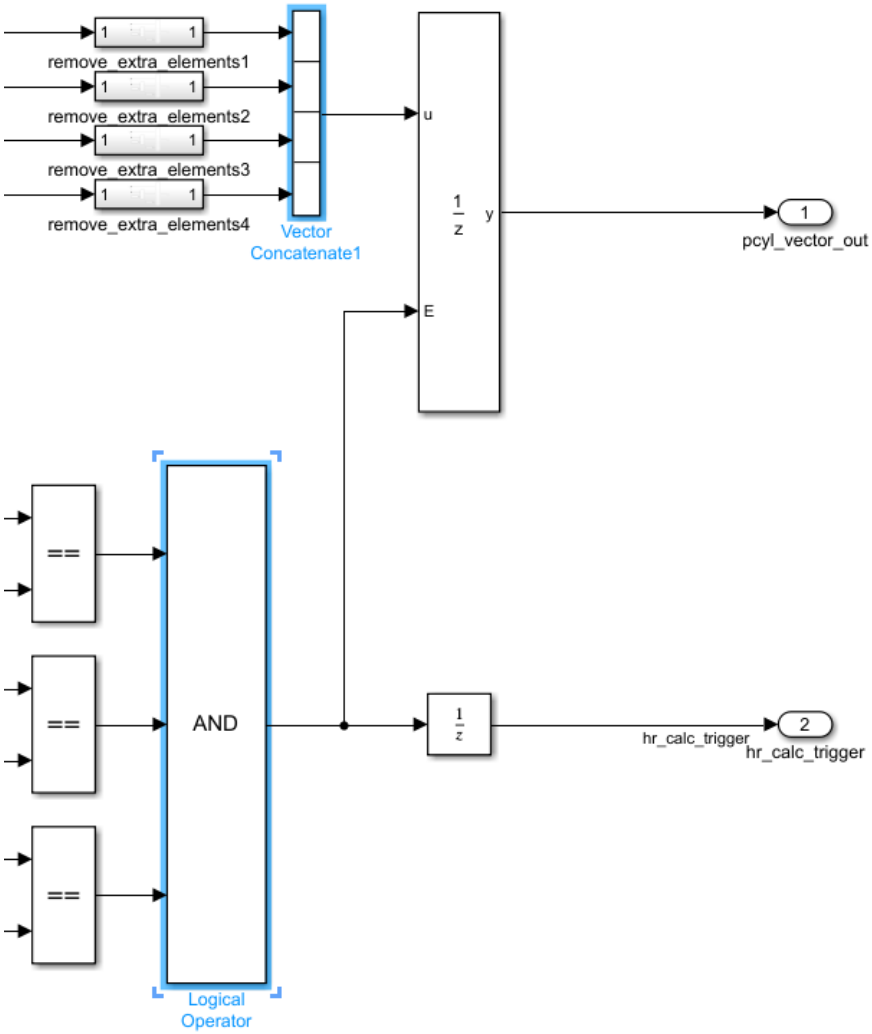


Figure 49: "Build_pcyl_vector": Removing extra elements, sending vector to and triggering HRR calc once per cycle.

3.3.2 Heat release calculation

The Simulink "heat release calculation" model used in this thesis has since earlier been developed by Kaas [58]. In this case, it has been adapted for use with the Simulink-GT-POWER and SCE applications. It is based on a formula by Hohenberg [59]:

The heat release rate (dQ) is calculated as follows:

$$\Delta Q_{1-2} = \frac{1}{\kappa - 1} \cdot V_2 \cdot \left[p_2 - p_1 \cdot \left(\frac{V_1}{V_2} \right)^{\kappa} \right] \quad (1)$$

In which:

- ΔQ_{1-2} : Heat release in crank angle increments.
- κ : Polytropic coefficient, generally between 1.3-1.4.
- p_1, p_2 : Cylinder pressure in PA
- : Volume in m³

Cumulative heat release (IntQ) is determined using the following method:

$$Q = \int \Delta Q \cdot dx \qquad (2)$$

The calculation excludes the heat transfer to the wall and utilises a constant polytropic index. [59]

As can be noted from the formula and the Simulink model in Figure 50, the heat release calculation utilises two vectors. One is the cylinder pressure vector that is sent via Speedgoat from measured pressure on the SCE. The other vector is the cylinder volume, calculated based on cylinder geometry, connecting rod length and compression ratio. The geometrical parameters and the polytropic coefficient that are used for HR calculation are stored in a MATLAB script and can be modified before the simulation is started. Heat release only needs to be calculated during the part of the cycle where combustion occurs. For this reason, the MATLAB script also contains crank angle parameters for heat release calculation start and stop. In this case, HR is calculated for 100° CA between 25° before TDC and 74° after TDC, but the range can be adjusted according to need (Figure 51).

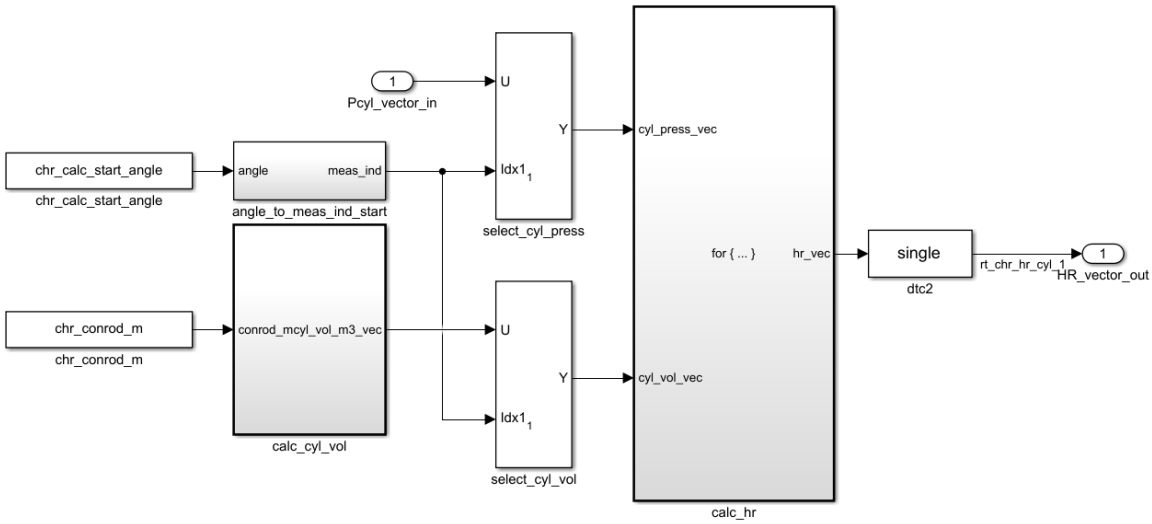


Figure 50: Simulink heat release calculation overview.

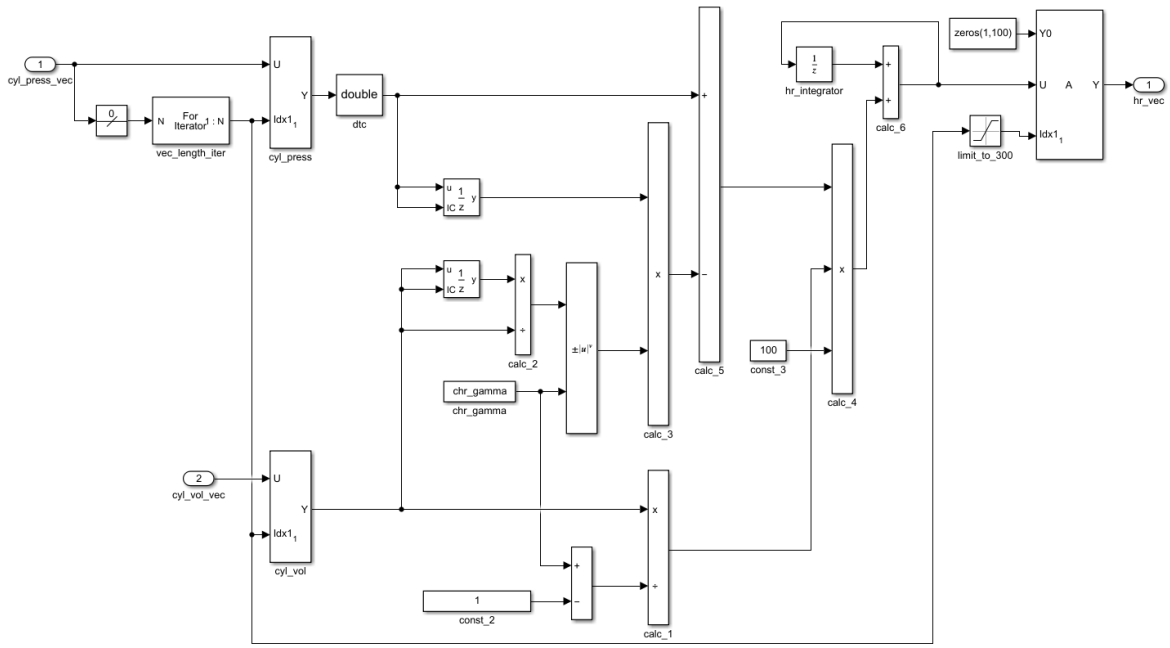


Figure 51: Heat release calculation subroutine.

3.3.3 Instant cumulative heat release to GT

The calculated instant cumulative heat release is sent to the GT-POWER model via a dynamic lookup table. Figure 52 shows an overview of the heat release to the GT function. Figure 53 contains the dynamic lookup table, and a reset function is shown in Figure. In addition, the procedure requires taking a “local cylinder crank angle signal” from the GT model via the “SimulinkHarness” component into the Simulink model and feeding it to the lookup table. The calculated HR is normalised, and a “Saturation” block is used to limit the heat release signal between 0-1, which are the limits of the signal that GT-POWER accepts. A position vector is also fed to the look-up table to limit the heat release that is sent to the GT model to the correct crank angle range. To avoid issues with “non-monotonically increasing”, HR, a function that compares the last HR value to the new one and chooses the larger of the two, is added. This ensures that the instant HR value that is sent to GT can never be smaller than the previous value. A reset feature that triggers when a new simulation cycle starts in GT-POWER is built to set the instant HR value to zero at the beginning of the cycle. The reset functionality also sends a new/updated HR vector to the lookup table for each simulation cycle.

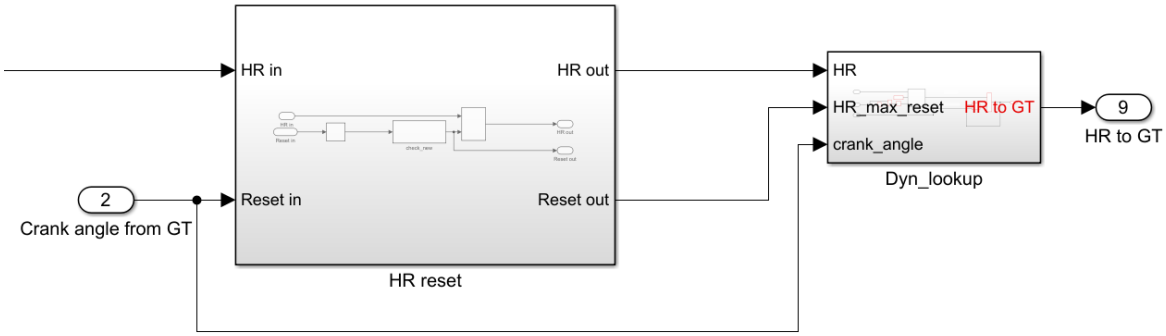


Figure 52: Heat release to GT: an overview.

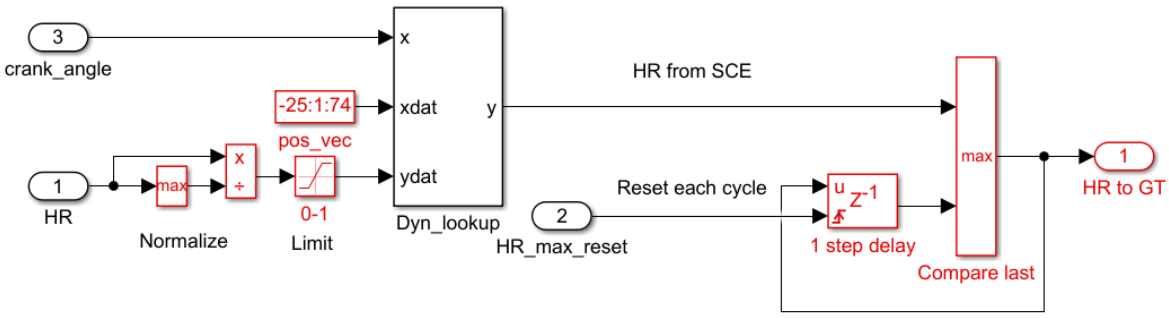


Figure 53: Dynamic lookup table.

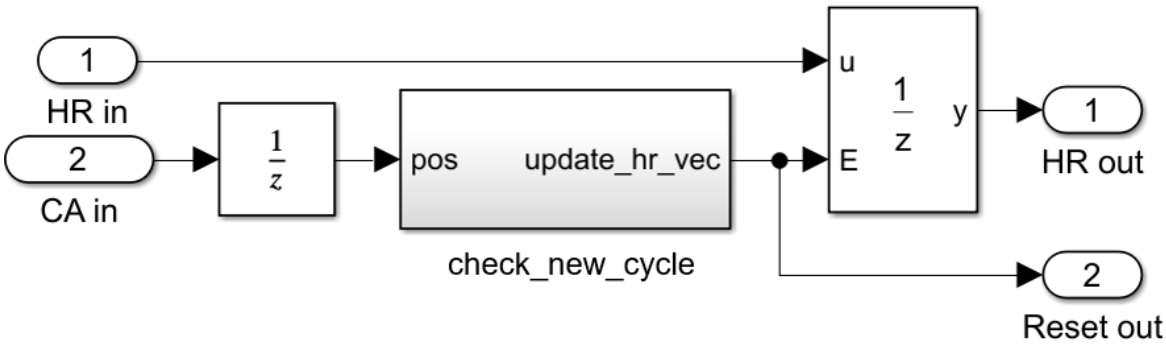


Figure 54: Reset subroutine.

Within GT-POWER, the received HR signal from Simulink is imposed to a cylinder through an "ActuatorConn" to connection ID 77: Profile: Cumulative Burned Fuel Fraction. The "Local Cylinder Angle" is taken out through connection ID 23 from the same cylinder and sent to Simulink. A "Switch" and a "Limiter" component make sure that the imposed combustion profile starts from 0 at the beginning of each simulation cycle and that the profile does not exceed 1 (Figure 55).

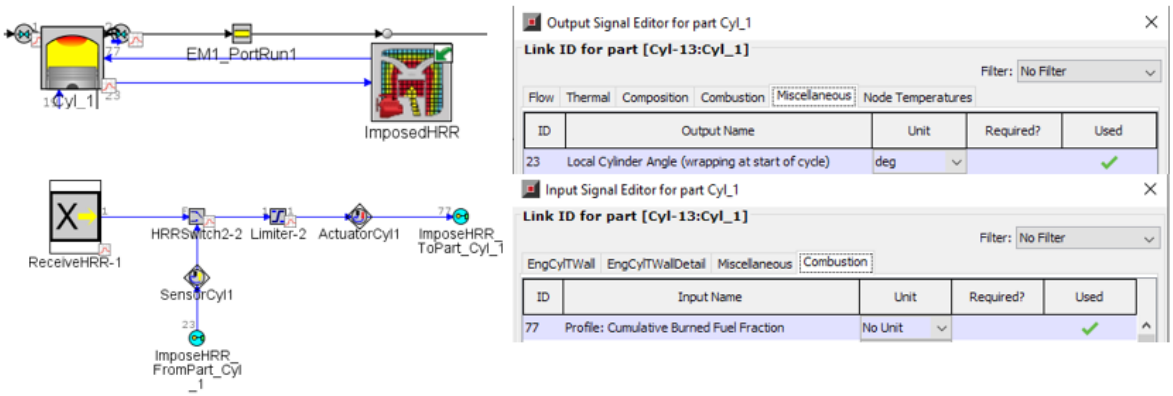


Figure 55: Real-time HR GT model overview.

4 Results

The following chapter will present and assess the results of the Simulink and GT-POWER models that have been built. An overview of the whole SCE – to – MCE model loop will be presented, and aspects such as real-time capability and accuracy will be addressed. A comparison between using measured heat release versus simulated burn rate will be presented.

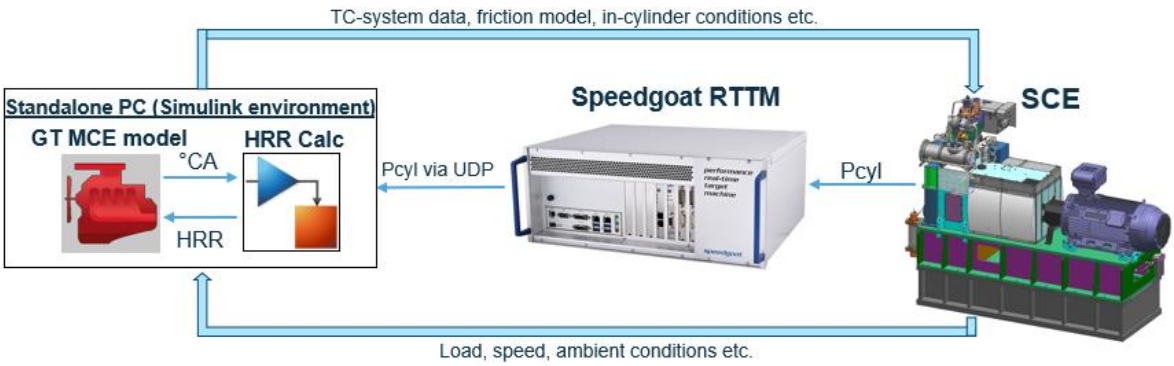


Figure 56: Overview of the SCE - to - MCE model system.

Figure 56 shows an overview of the setup that concerns this thesis. To enable real-time modelling of an MCE with combustion feedback from an SCE, the cylinder pressure processing and heat release calculation models have been built up in Simulink. In addition, the modelled MCE was converted from an FRM to an xRT-model to achieve a factor of real-time lower than one while still allowing crank angle resolved combustion modelling.

4.1 Real-time capability

The first version of the real-time combustion profile model in Simulink had separate dynamic lookup tables and reset functions for each cylinder of the MCE model in order to impose the real

combustion profile in all cylinders (8 in total). Correspondingly the GT MCE model also needed signal and actuation components for all cylinders. It was noted that actuating all cylinders, with all the features it included, created much overhead and made the simulation time slower than real-time in some cases, mainly with time steps smaller than 0.15ms. As the real combustion feedback is coming only from one cylinder, meaning that all eight cylinders in the MCE model will get the same combustion profile imposed, it was deemed unnecessary to impose the combustion on all cylinders via Simulink. Instead, the decision was made to only impose HR on one cylinder in the MCE model and use the cylinder copying option "copy-RT-partial" instead, which copies the combustion from the lead cylinder to the rest (Figure 57).

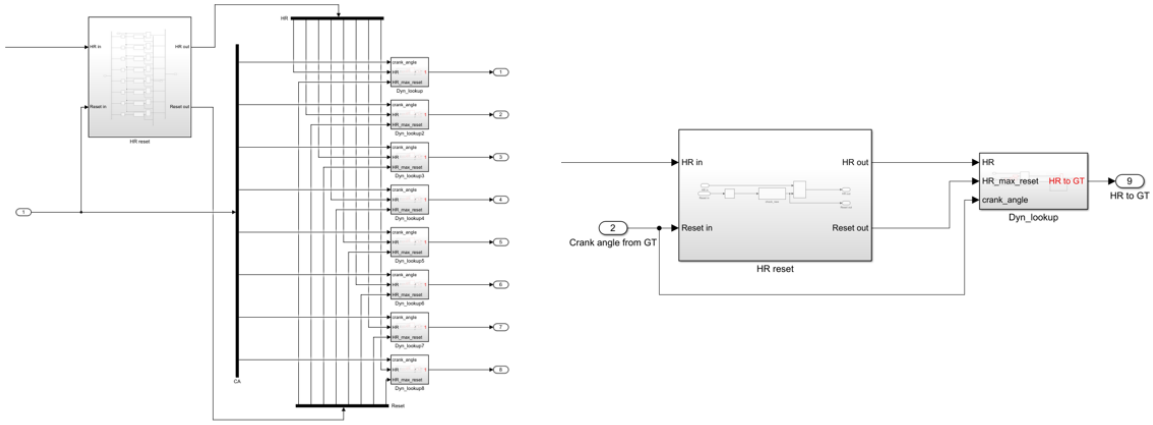


Figure 57: Left: Sending HR profile to all cylinders separately. Right: Sending HR profile only to lead cylinder.

To further minimise overhead and increase simulation speed, "Rate Transition" blocks were added to the Simulink models (Figure 58). "Rate Transition" blocks allow different functions and subroutines to run with different time steps. As only the part which sends the HR profile point-by-point needs to run at the same time step as the GT model, most other parts could run with significantly larger time steps. The cylinder pressure vector changes only once per engine cycle, meaning that it is not necessary to receive the UDP packets, build the cylinder pressure vector and calculate the HR every 0.2 ms when the cycle time is over 100 ms. Another example is the signals sent to the Simulink environment over Modbus, such as engine speed, load and ambient pressure and temperature, as these are slow-changing parameters relative to engine cycle times.

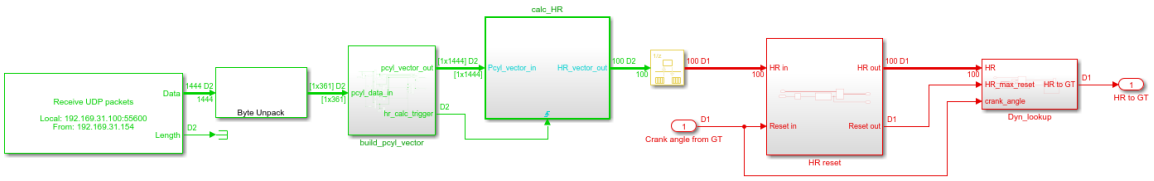
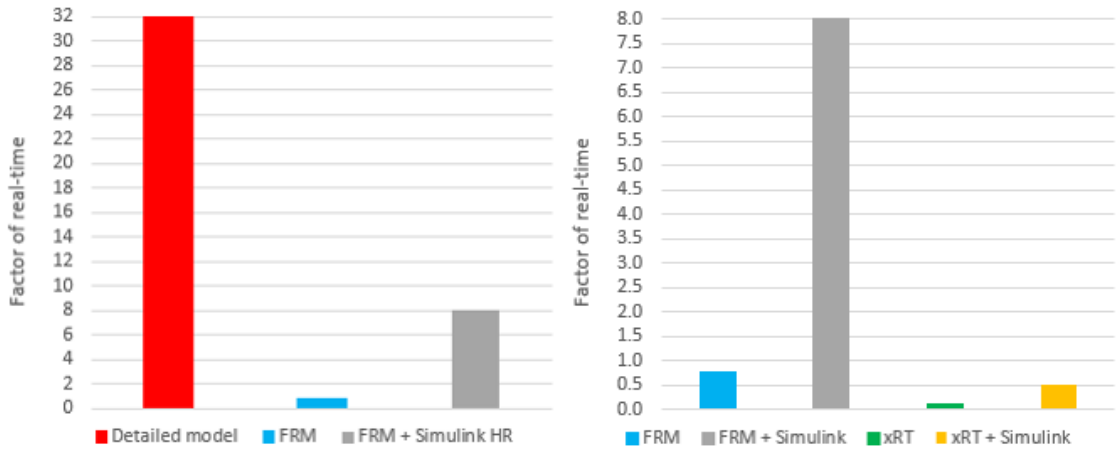


Figure 58: Different colours indicate different time steps. Green elements are 0.1 s (~once per cycle), and red elements are the same as GT 0.1- 0.2 ms (0.6-1.2° CA).

The first test with the "optimised" Simulink-GTP-Speedgoat model loop utilising the 8-cylinder xRT model and xRT-license is real-time capable down to 0.1 ms time step, corresponding to 0.6° CA at 1000 rpm engine speed. Simulink is still responsible for the bulk of the overhead, as the GT-Model on its own is about five times faster compared to the co-simulation model. Figure 59 shows an overview of the different models and their corresponding factors in real-time.



Model	Detailed model	FRM	FRM + Simulink HR	xRT	xRT + Simulink HR
Factor of real-time	32	0,75	8	0,1	0,5

Figure 59: Factor of real-time for various models.

4.2 Accuracy vs timestep

Running the xRT + Simulink HR model with a time-step of 0.2 ms gives a crank angle resolution of ~1° CA and a real-time factor of 0.5 (twice as fast as real-time). From a crank angle resolved combustion point of view, a resolution of 1° CA or less should be good enough for accurate combustion modelling. However, it remains to be validated. With 0.1 ms time-step, the crank angle resolution is ~0.5° CA, and the model runs with a real-time factor of 1, giving very little room for adding features in the future without further optimisation. Nevertheless, in this case, going lower than 0.1 ms time steps (0.5°CA) gives no benefit since the combustion feedback from the SCE is provided with a 0.5° CA resolution.

According to the GT manual [11], already at 0.2 ms or larger time steps, models with simplified geometries such as this one, performance variations and instabilities can occur. This is a result of high mass flow rates through the exhaust valves and can become especially apparent in TC systems. A “Stable Flow Through Valves and Throttles Multiplier” can be helpful for solving this issue.

Figure 60 shows an example of a heat release profile sent to the GT model using different timesteps. The first case is run with a very small timestep and shows a smooth and continuous combustion profile. The second case is run with a large time step and exhibits a very different curve with sharp steps and a completely different profile that would lead to a very different combustion. The example is meant to showcase how timestep and crank angle resolution can affect combustion modelling.

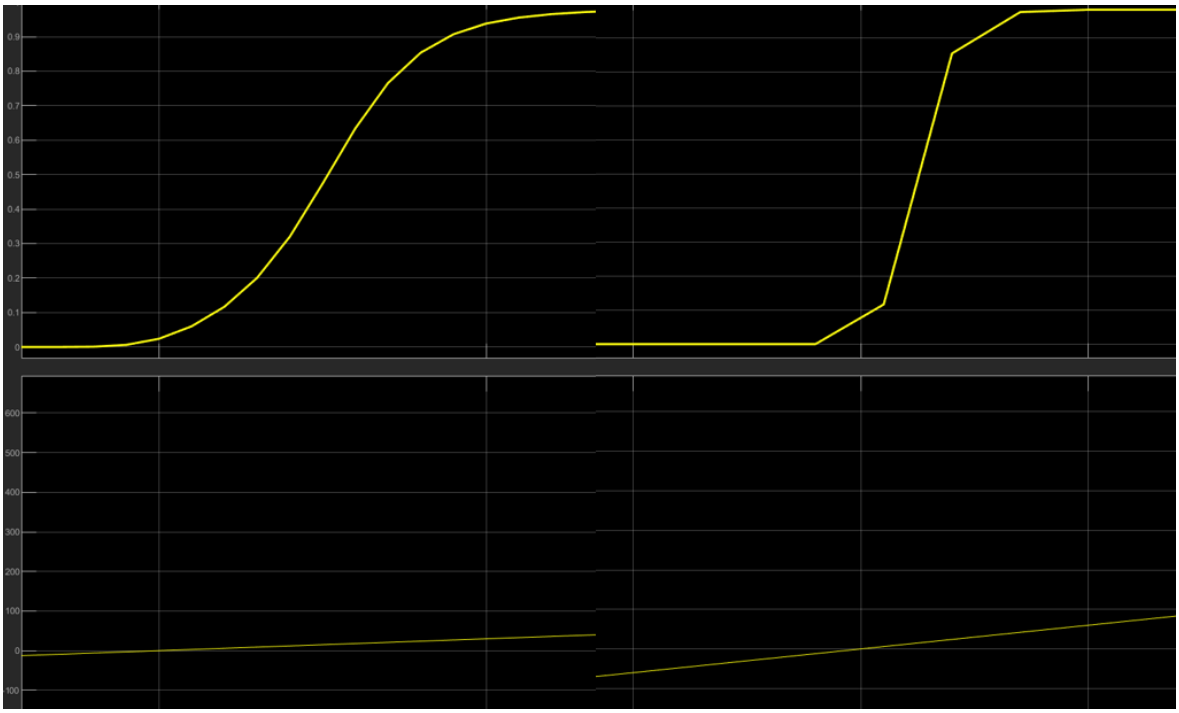


Figure 60: Combustion profile vs. timestep. Top: Cumulative Heat Release from SCE. Bottom: Crank angle from GT.

4.3 Heat release rate vs Burn rate

As noted in the Engine Performance Manual for GT-POWER, heat release rate and burn rate are not the same thing since burn rate takes into account the chemical reactions between air and fuel, while heat release only considers the rate at which fuel is converted into thermal energy. The implications are that if heat release is used as a combustion profile, it will lead to differences in the modelled cylinder pressure compared to a burn rate. On the other hand, the only way to correctly calculate a burn rate is to do a “reverse” run with either a CPOA or TPA model to analyse the cylinder pressure. Since it is not possible to impose cylinder pressure in real-time in a running model, the only option for real-time combustion feedback is to use heat release instead.

A comparison was made in GT-POWER to check the difference between using a combustion rate and a heat release rate for combustion modelling and cylinder pressure calculation. A cylinder pressure curve was imposed in the detailed model, and a reverse run was carried out to obtain the burn rate. Both the burn rate and the calculated heat release rate from the cylinder pressure curve were, in turn, imposed as the combustion profile for the model, and a forward run was performed to calculate the corresponding cylinder pressure. In conclusion, there is clearly a difference between using a burn rate and a heat release rate. However, the difference was surprisingly small (Figure 61). A minor correction to the imposed heat release vector or a multiplier might be enough to compensate for the variance. Nevertheless, this still needs further validation as it was only tested for one case at one operating point.

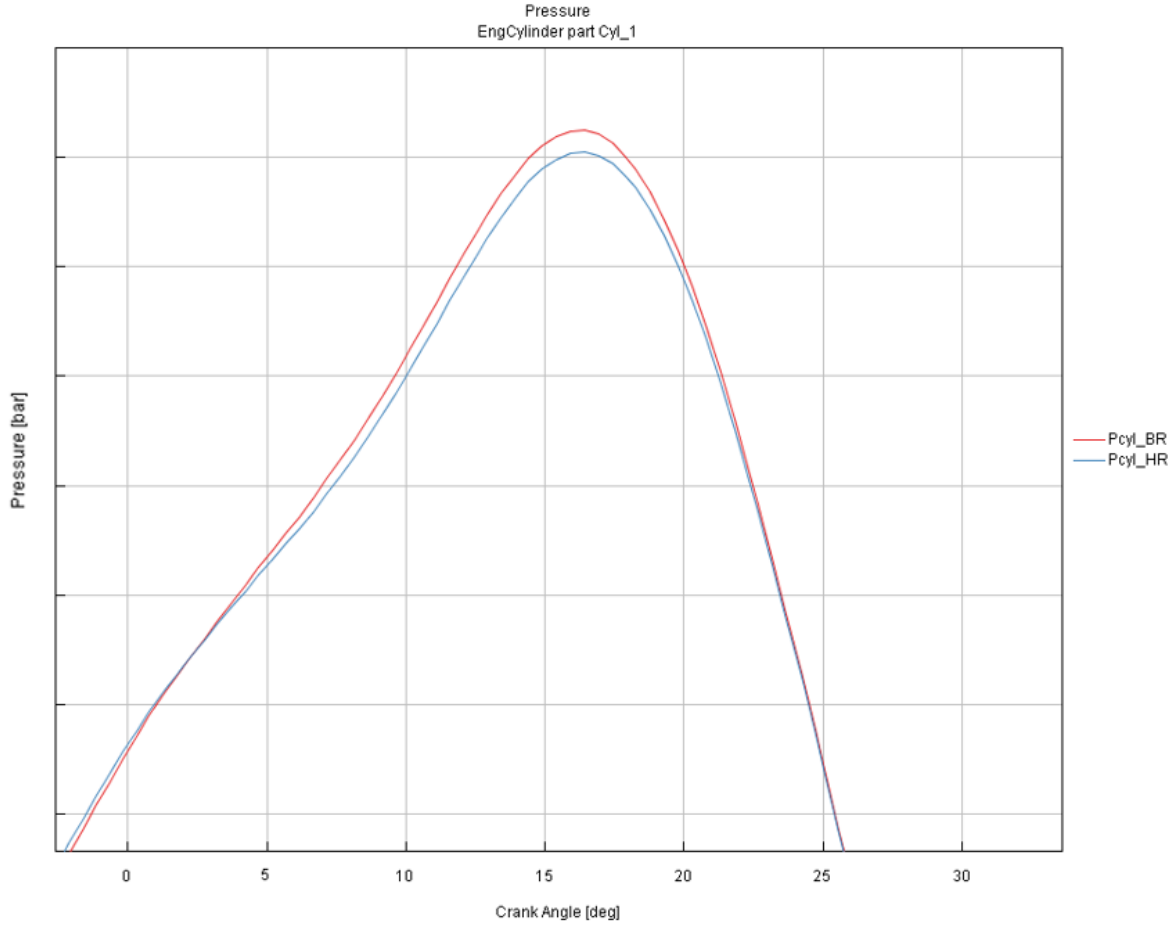


Figure 61: Burn rate- and Heat release rate-based cylinder pressure.

5 Discussion and conclusions

The thesis aimed to allow the SCE research platform to be run more realistically and according to actual MCE conditions by combining it with a real-time engine model built up in GT-POWER simulation software. Modelling certain aspects of an MCE that the SCE lacks, such as TC system, gas exchange dynamics etc. and applying the results for use in the control and operation of the SCE enhances the output from the research platform greatly. In addition, real-time combustion feedback from the SCE to the MCE improves the model accuracy and enables the possibility of calculating in-cylinder conditions such as compression end temperature, EGR and residual gas fraction, which are critical to understanding complex phenomena within modern and advanced combustion concepts.

A literature survey was performed, looking at real-time modelling applications in the ICE industry, focusing on GT-POWER. The outcome showed that real-time modelling is a widespread and essential practice in the industry and is used for a variety of applications, including model-based design and “in-the-loop” simulations for virtual testing and calibration, end-to-end simulations for entire vehicle simulations, including emission prediction, and various digital twin concepts. A few applications also included modelling real engine dynamics for implementation on an SCE test bench to improve transient testing.

The implementation consisted of a detailed review of different methods for imposing a combustion profile into the GT-simulation software and the MCE model within and selecting the most appropriate method considering real-time criteria and simulation accuracy. A Simulink model for processing cylinder pressure data from the SCE, calculating a combustion profile, and sending said combustion profile to the simulated MCE was constructed. In addition, the existing fast-running MCE model was converted into a GT-POWER xRT version in order to meet real-time requirements when running through Simulink co-simulation.

The result of realising the thesis is a Simulink-based environment with an 8-cylinder Wärtsilä 25DF GT-POWER xRT model with real-time combustion feedback from the SCE, capable of running at a real-time factor of 0.5 (twice as fast as real-time). The MCE model gets input such as combustion profile, engine load and speed, and ambient conditions from the SCE via Simulink and in return, the output from the model, including TC system parameters and in-cylinder conditions, is fed back to the SCE. The xRT model performance results are within +-2% of the initial FRM model and show good agreement with experimental TC system data.

A significant improvement was the TC system modelling compared to using the simpler MATLAB-based model. The setup provides a real-time accurate TC model that can easily be modified for different TC setups and layouts. One- or two-stage turbocharging, variable geometry turbines and sequential systems can be tested with little effort by simply changing the TC subsystem in the MCE model.

Real-time modelling of the MCE and the feedback it sends to the SCE can, in theory, be used for transient testing, such as loading and start/stop tests, as the dynamics of the MCE are captured quite well. However, the SCE control system would need significant modification to enable this in practice. Charge air, exhaust and gaseous fuel systems need to be faster than they currently are.

Regarding the accuracy of the MCE model, as heat release rate is used instead of burn rate, there is room for improvement. The calculation of cylinder pressure and other in-cylinder conditions may need to be more accurate for a wide variety of operating conditions, as a comparison between heat release rate and burn rate has only been made for one operating point. In addition, as the way of

implementing the combustion profile into the MCE model only allows monotonically increasing profile values, this may affect accuracy. For example, it may not be able to catch or simulate all types of combustion phenomena (e.g., knocking).

Further development of the model includes:

- Calibration and comparison of the xRT model over a wide variety of operating points.
- An extensive comparison between using heat release and burn rates and developing a method or calibration factor to compensate for the difference if needed.
- Data handling of inputs and outputs. The current model uses inputs from various acquisition systems such as Speedgoat, DEWESOFT and D2T Morphee. A future version would use inputs from only one system for simplification. This would also likely remove some overhead from the Simulink model and might result in a faster model speed.
- Using the MCE model in-cylinder conditions to control trapped air mass and residuals in the SCE instead of the traditional way of controlling charge air and back pressures. This would result in a better match between SCE and MCE since gas exchange dynamics could be ignored.
- Workflow development together with Gamma Technologies: A future version of GT-POWER may include a CPOA/TPA model capable of receiving cylinder pressure directly in real-time. This would enable the use of burn rate instead of heat release rate and generally give more accurate combustion modelling.

7 References

[1] IPCC, “The Intergovernmental Panel on Climate Change,” 2021. [Online]. Available: https://www.ipcc.ch/report/ar6/wg1/. [Accessed 11 October 2022].

[2] UNFCCC, “Global GHG Emissions Trends,” 2021. [Online]. Available: https://www.unfccc.int/sites/default/files/resource/Global%20GHG%20Emissions%20Trends%202021.pdf. [Accessed 11 October 2022].

[3] European Council, “Fit for 55,” 29 June 2022. [Online]. Available: https://www.consilium.europa.eu/en/policies/green-deal/fit-for-55-the-eu-plan-for-a-green-transition/. [Accessed 12 August 2022].

[4] IMO, “Initial IMO Strategy on reduction of GHG emissions from ships,” 2018. [Online]. [Accessed 5 May 2022].

[5] Wärtsilä, “Wärtsilä commits to carbon neutrality by 2030,” October 2021. [Online]. Available: https://www.wartsila.com/media/news/26-10-2021-wartsila-commits-to-carbon-neutrality-by-2030-2995912. [Accessed 5 May 2022].

[6] K. G.-R. A. J. Wojciech Tutak, “Combustion and Emission Characteristics of a Biodiesel-Hydrogen Dual-Fuel Engine,” *MDPI,* 2020.

[7] Gamma Technologies, “What is GT-Power?” 2022. [Online]. Available: https://www.gtisoft.com/gt-power/. [Accessed 8 April 2022].

[8] *Engine Performance Application Manual,* Gamma Technologies, 2022.

[9] S. Ryu, H. Yoon, H. Choe and G. Choi, “Development of HiLS system for HiMSEN Dual Fuel Engine,” in *CIMAC Proceedings, No. 195*, Helsinki, 2016.

[10] H. Wu and M.-F. Li, “A Hardware-in-the-Loop (HIL) Bench Test of a GT-Power Fast Running Model for Rapid Control Prototyping (RCP) Verification,” SAE International, 2016.

[11] *Engine Performance Tutorials,* Gamma Technologies, 2022.

[12] B. Lumpp, M. Tanimou, E. Trapel, M. Münzenmay, M. McMackin, E. Boullion and K. Zimmermann, “Desktop Simulation and Calibration of Diesel Engine ECU Software using Software-In-The-Loop Methodology,” in *GT-SUITE European Conference*, Frankfurt / Main, 2013.

[13] Gamma Technologies, “What is GT-Power-xRT,” 2022. [Online]. Available: https://www.gtisoft.com/gt-power-xrt/. [Accessed 8 April 2022].

[14] Gamma Technologies, “GT-POWER-xRT,” April 2019. [Online]. Available: https://www.gtisoft.com/wp-content/uploads/2019/04/GT-POWER-xRT_Brochure.pdf. [Accessed May 2022].

[15] A. Kumar, “Model-Based Design and MIL to HIL,” 19 April 2020. [Online]. Available: https://algo.volganga.com/model-based-design-and-mil-to-hil/. [Accessed March 2022].

[16] J. Nibert, M. E. Herniter and Z. Chambers, “Model-Based Design for MIL. SIL, and HIL,” *World Electric Vehicle Journal,* vol. 5, pp. 1121-1130, 2012.

[17] N. Srinivas, N. Panditi, S. Schmidt and R. Garrelfs, “MIL/SIL/PIL Approach - A new paradigm in Model-Based Development,” in *MATLAB EXPO*, 2014.

[18] C. Kleijn, “Introduction to Hardware-in-the-Loop Simulation,” Controllab, 2017.

[19] Gamma Technologies, “Control, MiL, SiL, and HiL Simulation,” October 2021. [Online]. Available: https://www.gtisoft.com/control-mil-sil-and-hil/. [Accessed 6 May 2022].

[20] A. R. Plummer, “Model-in-the-Loop Testing,” *Institution of Mechanical Engineers,* vol. 220, no. 3, 2006.

[21] Acsystème, “Model-Based Design,” 2022. [Online]. Available: https://www.acsysteme.com/en/customised-service/model-based-design/. [Accessed 11 Oct 2022].

[22] A. B. Castanheira de Farias, R. S. Rodrigues, A. Murilo, R. Lopes and S. Avila, “Low-cost Hardware-in-the-Loop platform for embedded control strategies simulation,” *IEEE Access,* vol. 7, pp. 111499 - 111512, 2019.

[23] D. Bergström and R. Göransson, “Model- and Hardware-in-the-Loop Testing in a Model-Based Design Workflow,” Lund University - Department of Automatic Control, 2016.

[24] Mobility Insider, “What is software-in-the-loop-testing?,” APTIV, 17 March 2022. [Online]. Available: https://www.aptiv.com/en/insights/article/what-is-software-in-the-loop-testing. [Accessed 11 May 2022].

[25] Speedgoat, “Plant Simulation HIL,” 21 May 2017. [Online]. Available: https://www.speedgoat.com/applications-industries/applications/plant-simulation-hil. [Accessed 5 May 2022].

[26] AGCO Fendt, “Automated testing of tractor controllers using Hardware-in-the-Loop test benches,” Speedgoat, 2015.

[27] S. Chandrashekar, “Real Time Fast Running Engine Modelling in GT Power - Development of the Virtual Drivetrain Simulation Environment,” Chalmers University of Technology; Volvo Group Trucks Technology, 2019.

[28] J. Andric, D. Schimmel, S. A. D, J. Sjoblom and E. Faghani, “Development and Calibration of One Dimensional Engine Model for Hardware-In-The-Loop Applications,” SAE International, 2018.

[29] H. Patel and V. Bansal, “Implementation of a Real-Time GT-POWER-xRT Engine Model in Virtual Test Bench (VTB),” in *GT Suite Conference*, 2021.

[30] R. Padmavathi, S. Asthana, M. VikramRaj, K. Chaitanya and K. Chandrashekhar, “Model-in-Loop (MIL) Toolchain development for Emissions Prediction using GT-DRIVE+, GT-POWER-xRT and MATLAB SIMULINK,” in *GT Suite Conference*, Frankfurt / Main, 2021.

[31] S. Hautala, E. Söderäng and S. Niemi, “Digital-twinning the engine research platform in VEBIC,” *Integrated Energy Solutions to Smart And Green Shipping,* pp. 120-128, 2020.

[32] S. Saurabh and B. Sureka, “An_Evaluation_of_an_Unhealthy_Part_Identification_Using_a_0D-1D_Diesel_Engine_Simulation_Based_Digital_Twin,” SAE International, 2022.

[33] F. Xia, S.-Y. Lee, J. Andert, A. Kampmeier, T. Scheel, M. Ehrly, R. Tharmakulasingam, Y. Takashi and T. Kumagai, “Crank-Angle Resolved Real-Time Engine Modelling: A Seamless Transfer from Concept Design to HiL Testing,” SAE International, 2018.

[34] J. Andert, F. Xia, S. Klein, D. Guse, R. Savelsberg, R. Tharmakulasingam, M. Thewes and J. Scharf, “Road-to-Rig-to-Desktop - Virtual Development Using Real-Time Engine Modelling and Powertrain Co-simulation,” *International Journal of Engine Research,* vol. 20, no. 7, 2018.

[35] J. Andric, D. Schimmel and J. Heide, “Calibration Procedure for Measurement-Based Fast Running Model for Hardware-in-the-Loop Powertrain Systems,” SAE International, 2020.

[36] H. He, “GT-xRT gas exchange model for SiL and HiL testing,” in *GT Suite conference*, Frankfurt / Main, 2022.

[37] G. Boccardo, E. Servetto, A. Seebooa, G. Sammut and J. Primitivo, “Development of a Real-Time GT-POWER xRT Model for Virtual Calibration,” in *GT Suite Conference*, 2020.

[38] S. Hautala, M. Mikulski, E. Söderäng, X. Storm and S. Niemi, “Toward a digital twin of a mid-speed marine engine: From detailed 1D engine model to real-time implementation on a target-platform,” *International Journal of Engine Research,* no. Special Issue: 9th International Congress on Combustion Engines (PTNSS 2021), 2022.

[39] S. S. E. Hautala and S. Niemi, “Real-time simulation of combined engine and electrical equipment model in Simulink RT,” *Integrated Energy Solutions to Smart and Green Shipping: 2021 Edition,* pp. 72-81, 2021.

[40] E. Söderäng, S. Hautala, M. Mikulski, X. Storm and S. Niemi, “Development of a digital twin for real-time simulation of a combustion engine-based power plant with battery storage and grid coupling,” *Energy Conversion and Management,* 15 August 2022.

[41] “What is a digital twin?” IBM, 12 April 2021. [Online]. Available: https://www.ibm.com/topics/what-is-a-digital-twin. [Accessed 1 October 2022].

[42] O. Bondarenko and T. Fukuda, “Development of a diesel engine’s digital twin for predicting propulsion system dynamics,” Elsevier, 2020.

[43] P. Mayr, G. Pirker, A. Wimmer and M. Krenn, “Simulation-Based Control of Transient SCE Operation,” SAE International, 2017.

[44] J. Lahti and J. Moskwa, “A Transient Test System for Single Cylinder Research Engines With Real-Time Simulation of Multi-Cylinder Crankshaft and Intake Manifold Dynamics,” SAE International, 2004.

[45] J. Lahti, J. Moskwa and M. Snyder, “Simulating Transient Multi-Cylinder Engine Gas Exchange Dynamics on a Single-Cylinder Research Engine,” SAE International, 2006.

[46] Speedgoat, “Simulink Real-Time Workflow,” 1 March 2017. [Online]. Available: https://www.speedgoat.com/solutions/simulink-real-time-workflow. [Accessed 2 June 2022].

[47] C. Hahn, “What is Simulink Real-Time,” Speedgoat, 2017. [Online]. Available: https://se.mathworks.com/videos/what-is-simulink-real-time-1619099403861.html.

[48] *Rapid Controls Prototyping: Tech Training,* Wärtsilä, 2022.

[49] T. Troberg, *Advanced Concepts: Rapid Controls Prototyping Way of Working,* Wärtsilä, 2020.

[50] *RCP Tools and Environment,* Wärtsilä, 2022.

[51] Speedgoat, “Real-Time UDP,” Speedgoat, 8 May 2017. [Online]. Available: https://www.speedgoat.com/products-services/i-o-connectivity/protocols/real-time-udp. [Accessed 5 April 2022].

[52] MathWorks, “UDP Transport Protocol,” MathWorks, 2021. [Online]. Available: https://se.mathworks.com/help/slrealtime/io_ref/udp-transport-protocol.html. [Accessed 5 April 2022].

[53] C. M. Kozierok, “Transmission Control Protocol (TCP) and User Datagram Protocol (UDP),” in *The TCP/IP Guide*, San Francisco, No Starch Press, 2005, pp. 806-835, 945.

[54] M. Cook, “TCP vs UDP: What's the Difference?,” Lifesize, 24 October 2017. [Online]. Available: https://www.lifesize.com/blog/tcp-vs-udp/. [Accessed 6 May 2022].

[55] J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, 7th Edition, Hoboken, New Jersey: Pearson, 2017, pp. 238-243.

[56] *Reference Manual,* Gamma Technologies, 2022.

[57] MathWorks, *MATLAB 2021b Documentation,* 2021.

[58] T. Kaas, “Method for calculating heat release within Simulink,” Wärtsilä Internal Report, 2015.

[59] “Heat Release Calculation (Hohenberg),” FEV Software and Testing Solutions GmbH, 7 Jan 2013. [Online]. Available: https://www.uniplot.de/documents/en/src/addins/ifile.html. [Accessed 2 June 2022].

[60] Wärtsilä, “Wärtsilä launches major test programme towards carbon-free solutions with hydrogen and ammonia,” July 2021. [Online]. Available: https://www.wartsila.com/media/news/14-07-2021-wartsila-launches-major-test-

programme-towards-carbon-free-solutions-with-hydrogen-and-ammonia-2953362. [Accessed 5 May 2022].

[61] J. Shepard, “How do MIL, SIL, PIL and HIL simulation and testing relate to MBSE,” 14 September 2022. [Online]. Available: https://www.analogictips.com/how-do-mil-sil-pil-and-hil-simulation-and-testing-relate-to-mbse-faq/. [Accessed November 2022].

[62] E. Johansson, “Real Time GT-POWER Simulation for Emission Prediction for Engine Verification,” in *GT Suite Conference*, 2012.

[63] J. Heywood, Internal Combustion Engine Fundamentals, ISBN 0-07-028637-X: McGraw Hill, 1998.