Jaber McBreen

Matching 1-dimensional engine flow solution with digital twin requirements for system-level simulations

A model-in-the-loop study of an engine-based stationary power plant

School of technology and innovations
Master's thesis in energy technology

Vaasa 2024

VAASAN YLIOPISTO
Tekniikan ja innovaatiojohtamisen yksikkö
Tekijä: Jaber McBreen
Tutkielman nimi: Matching 1-dimensional engine flow solution with digital twin requirements for system-level simulations : A model-in-the-loop study of an engine-based stationary power plant
Tutkinto: Diplomi-insinööri
Koulutusohjelma: Energiatekniikka
Ohjaajat: Professori Maciej Mikulski, professori Marcelo Godoy Simões
Vuosi: 2024 **Sivuja:** 129

TIIVISTELMÄ:
Polttomoottorit puolustavat paikkaansa merisovellusten pääasiallisena voimanlähteenä, mutta pakottava tarve vähentää kokonaispäästöjä vaatii niiden kehittämistä taloudellisemmaksi. Tämä opinnäytetyö tutkii model-in-the-loop -lähestymistavan tarjoamia mahdollisuuksia dieselmoottoripohjaisten hybridivoimalaitosten integroinnissa ja simuloinnissa. Työ keskittyy ensisijaisesti yksiulotteisten virtausratkaisujen käyttökelpoisuuden arviointiin tarkkojen ja laskennallisesti tehokkaiden digitaalisten kaksosten luomiseksi järjestelmätason simulointeja varten.

Lähestymistavan uutuusarvo piilee sen kyvyssä toteuttaa korkean tarkkuuden fysiikkapohjainen mallinnus reaaliaikaisesti. Mallinnustekniikka tarjoaa merkittävän parannuksen verrattuna perinteisiin black-box -malleihin yhdistämällä korkean tarkkuuden ennustavan mallinnustavan matalaan laskennalliseen kustannukseen.

Speedgoat Mobile -kohdekoneen osoitetaan kykenevän ajamaan hybridivoimalaitoksen mallia reaaliaikaisesti. Näin ollen päivitetyn järjestelmän voidaan osoittaa soveltuvan hardware-in-the-loop -simulaatioihin. Mallia voitiin ajaa näytteistystaajuuksilla jotka vastasivat moottorimallin tapauksessa 14:ää kammenkulma-astetta ja voimalaitosmallin tapauksessa 18:aa kammenkulma-astetta. Nopeuden todettiin riittävän sylinterin palotapahtuman aiheuttaman paineenlisäyksen mallintamiseen riittävällä tarkkuudella. Lisäksi näytteistystaajuuden vaikutuksia mallin tarkkuuteen arvioitiin laajasti.

AVAINSANAT: Model-in-the-loop, digitaalinen kaksonen, hybridivoimalaitos, reaaliaikainen simulointi, dieselmoottorin mallinnus

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Jaber McBreen
Title of the Thesis: Matching 1-dimensional engine flow solution with digital twin requirements for system-level simulations : A model-in-the-loop study of an engine-based stationary power plant
Degree: Master of science
Programme: Energy technology
Supervisors: Professor Maciej Mikulski, Professor Marcelo Godoy Simões
Year: 2024 **Pages:** 129

ABSTRACT:
Internal combustion engines hold their position as the main source of power in maritime applications, but the inescapable requirement of reducing overall emissions calls for more advanced solutions. This thesis explores the integration and simulation capabilities of a model-in-the-loop approach for a diesel engine-based hybrid power plant, with the primary focus being in evaluating the feasibility of using one-dimensional flow solutions in creating accurate and computationally efficient digital twins for system-level simulations.

The novelty of this approach lies in its ability to integrate high-fidelity physical modelling with real-time simulation capabilities, providing an unprecedented level of detail and accuracy. The modelling technique offers significant improvements over traditional black-box models by retaining the physical realism required for predictive simulations while ensuring computational efficiency.

The use of a Speedgoat Mobile target machine is demonstrated to handle the computational demands of the hybrid power plant model, ensuring real-time performance. Key results show that the upgraded system successfully meets real-time requirements, proving its potential for hardware-in-the-loop simulations. Explicitly, the model could be run at sample times corresponding to 14 crank angle degrees for the isolated engine model and 18 crank angle degrees for the combined hybrid power plant model. This speed matches the phenomenological timescale of in-cylinder combustion, enabling the future digital twin with state-of-the art fidelity. Additionally, sample time effect on model accuracy was assessed extensively.

KEYWORDS: Model-in-the-loop, digital twin, hybrid power plant, real-time simulation, diesel engine modelling

Contents

Figures

Tables

Abbreviations

0D	Zero-dimensional
1D	One-dimensional
3D	Three-dimensional
ADC	Analog to digital converter
BESS	Battery energy storage system
BMEP	Brake mean effective pressure
CAD	Crank angle degree
CFD	Computational fluid dynamics
DAC	Digital to analog converter
DDM	Data-driven model
ECU	Electronic control unit
FC	Frequency converter
FRM	Fast-running model
GPOS	General-purpose operating system
HiL	Hardware-in-the-loop
MiL	Model-in-the-loop

MBD	Model-based design
MBSE	Model-based systems engineering
MVEM	Mean-value engine model
ODE	Ordinary differential equation
PDE	Partial differential equation
PLM	Product lifecycle management
RCCI	Reactivity-controlled compression ignition
RCP	Rapid control prototyping
RPM	Revolutions per minute
RTF	Real-time factor
RTOS	Real-time operating system
RTS	Real-time simulation
SCIG	Squirrel cage induction generator
SiL	Software-in-the-loop
TET	Task execution time

Other Symbols

P_{max}	Maximum average cylinder pressure
T_{exh}	Average exhaust gas temperature

1 Introduction

Being an efficient tool for converting fuel chemical energy into mechanical energy, diesel engines still defend their place as an energy source for ship propulsion systems (Bondarenko & Fukuda, 2020). Hybrid powertrains represent a critical advancement in the quest for more efficient and environmentally friendly energy solutions. These systems combine traditional internal combustion engines with electric propulsion, aiming to reduce fuel consumption and emissions during transients, while maintaining performance with peak-shaving scenarios. This load-stabilising feature ultimately allows to depart from conventional compression ignition principle towards ultra-clean, fuel-flexible low temperature combustion concepts. The development and optimisation of such powertrains require sophisticated modelling and simulation techniques to accurately represent the complex interactions within them (Laurén et al., 2022). Real-time simulation and the concept of digital twins have emerged as vital tools in this context, enabling engineers to test and refine control strategies under realistic operating conditions.

The significance of digital twins in this regard lie in their ability to mirror the behaviour of physical systems in a virtual environment, allowing for extensive testing without the risks and costs associated with physical prototypes. Furthermore, as noted by Smith et al. (2007): “[– –] engineers often just have more fun doing their job when they have the right set of tools.”

Previous research has laid a solid foundation for this field. Wurzenberger et al. (2013) demonstrated the feasibility of real-time engine simulations, highlighting key challenges such as task execution time (TET) spikes during engine cycles. Similarly, Söderäng et al. (2022) explored the integration of hybrid powertrain components in real-time simulations. Despite these advancements, there remain gaps in the ability to achieve consistent real-time performance with models including one-dimensional (1D) flow solution, particularly when integrating them into complex subsystems.

In the realm of engine modelling, model fidelity comes at the cost of increased computational demands. Modelling approaches varying from simplified zero-dimensional (0D)

thermodynamic models to extremely accurate three-dimensional (3D) models and variants between them have all found their application. Isermann, for one, gives a brief overview of the best-suited purposes of these approaches in Table 1.

Table 1. Comparison of air path model approaches (Isermann, 2014b)

	Zero-dimensional	**One-dimensional**	**3D**
modelling method	quasi-steady flow models, filling and emptying method	method of characteristics, method of differences	CFD methods
flow	(quasi-)steady	unsteady	unsteady
model complexity	low	high	very high
computing time	short	medium	large
real-time capability	yes	no	no
applications	parameter studies, control system design	engine design	engine design

Isermann states that the only approach capable of real-time operation, which is a prerequisite of e.g. control system design, is the zero-dimensional one. However, in the advent of advanced combustion concepts such as reactivity-controlled compression ignition (RCCI), it becomes ever more disadvantageous to have to disregard engine air path and combustion dynamics in control-oriented simulations. For this reason, Hautala et al. (2022) set out to strategically simplify a detailed 1D model of a Wärtsilä 4L20 engine in order to combine the added functionality of a 1D approach with the real-time potential of a 0D approach.

Söderäng et al. picked up on this work, using the resulting fast-running model (FRM) to represent the engine as a part of a hybrid power plant situated in Energy Lab. Coupling the engine model with a Simscape model of a grid-connected generator and battery energy storage system (BESS), Söderäng et al. found that the computational demands are mostly set by the chosen sample time of the engine model and that real-time simulations of the electrical model caused no issues at a sample time of 50 µs. As the sample time of the engine model had to be set to 1 s to avoid overruns, the model could not reliably reproduce in-cylinder pressure, which could be observed in the unrealistic oscillations in exhaust temperature.

1.1 Problem formation and scope of the work

This thesis aims to address these challenges by evaluating the computational limits of a hybrid power plant model being run on a Speedgoat real-time target machine. Following the work of Söderäng et al., the University of Vaasa has acquired a dedicated Speedgoat Mobile real-time target machine to replace the legacy machine. The target machine is expected to be substantially better suited for real-time simulations both in terms of performance and ease of use. Considering that the engine model is the main computational burden and that the hybrid power plant model is to be evolved into a digital twin, the main research questions stand as follows:

RQ1: What computational restrictions does a Speedgoat Mobile target machine set for running the fast-running engine model and hybrid power plant model?

RQ2: What are the necessary steps for evolving the model into a complete digital twin?

RQ3: What is the trade-off between simulation speed and accuracy while meeting real-time requirements and maximising sub-model fidelity?

RQ4: What benefits does the improved tool provide for system-level simulations?

The main contribution of this research is validating the level of fidelity reachable by the particular combination of software and hardware without sacrificing model accuracy. This work both advances the state of real-time simulation for hybrid powertrains and continues the groundwork for future digital twin development, with significant implications for both academic research and practical applications.

The structure of this thesis is as follows:

Chapter 1 provides an overview of hybrid powertrain systems, discussing their importance in the context of energy efficiency and environmental sustainability. The objectives of the research are outlined and the specific goals for this study are set.

Chapter 2 presents a review of the existing literature on real-time simulation and digital twins. Key studies are discussed, highlighting their contributions and the current gaps in the field. This review sets the stage for understanding the context and significance of the present research.

Chapter 3 explores the methodology and experimental setup used to test the hybrid power plant model. This includes a description of the software and hardware configurations, the approach to model reduction and the procedures for validating real-time performance and model accuracy.

Chapter 4 discusses the results of the simulations. The performance of both the isolated engine model and the combined hybrid power plant model are presented, focusing on task execution times and the impact of varying sample times on real-time capabilities. Extensive results are presented to support careful reviewing of the evolvement of model accuracy as a function of sample time. Additionally, system-level simulations are used to showcase model capabilities outside of this particular model-in-the-loop setting.

Chapter 5 provides an in-depth discussion of the findings. Revolving around the research questions presented before, the discussion condenses the results and their analysis into digestible outcomes.

Finally, **chapter 6** wraps up the thesis by providing an overview of the main conclusions drawn from the research. It condenses the information presented in chapter 5 into even more compact takeaways. Furthermore, it highlights the practical applications of the research findings and offers recommendations for future work.

2 Background

2.1 State of the art in engine simulations

In simulating a four-stroke engine a few key considerations must be made. The tools available for professional use allow intricate simulations of engine workings, but ultimately a choice between the level of detail and the simulation times has to be made (Coraddu, Kalikatzarakis, et al., 2022, p. 147). According to Grimmelius (Grimmelius, 2003), from a researcher's point of view the transparency of a model is of utmost importance and its envisioned use and goal should be considered in both model development and when evaluating model suitability. This is further supported by Heywood (Heywood, 2018), who emphasises clearly defined objectives along with well-fitted model structure and detailed content being a good practice.

Engine modelling and our understanding of the chemical and physical processes of an internal combustion engine are constantly evolving. Nevertheless, creating models that aim to recreate all engine processes at a detailed level is not yet feasible due to the exceedingly high computational costs (Mirfendreski et al., 2016). Thus, compromises must be made and modelling endeavours have to be carefully tailored to the task at hand, with the main focus being on processes that are most vital to achieving the set objective.

Coraddu et al. (2022) introduce three main approaches for modelling diesel engines; physical models, data-driven models and hybrid models. This chapter introduces these different approaches to the extent that is necessary for justifying the choices made in creating a digital twin of the power plant in question. Other ways of categorising engine models exist, of which one is described by Grimmelius (2003). There, rather than just separating model types by their fundamental mechanism, they are judged by their qualities in model level, model time domain, application time domain and model data requirements. However, in the scope of this work, it is sufficient to only consider the core

mechanism of engine simulations and to move on the continuum between white-box (physical) and black-box (data-driven) models.

Bozza et al. agree on this continuum, using two main modelling approaches (Bozza et al., 2020). One is to start from experimental data and identify the mathematical relations of measured values. Another, quite opposite to the first, is to begin from detailed 3D CFD simulations of physical phenomena taking place within the engine. As the phenomena are both complex and numerous, the computational complexity of this approach exceeds that of the first by multiple orders of magnitude. Bozza et al. also mention different methods being described in literature, varying between single-zone and quasi-dimensional approaches.

As the ultimate goal is of this work is to build a digital twin of not just an engine but a whole hybrid powertrain, the overall complexity of simulations far exceeds that of plain engine simulations. Simulations that include an engine model without it being the sole centre of attention, such as the one depicted in this thesis, particularly call for computationally efficient solutions (Laurén et al., 2022).

2.1.1 Physical models

When building a physical model of a diesel engine, the main goal is to create a model that imitates the engine's processes as accurately as possible within the constraints set by other factors. Since there are various complex physical phenomena happening within a cylinder during fuel oxidation, modelling them becomes challenging especially when computational resources are limited (Onorati & Montenegro, 2020). Examples of such phenomena are thermodynamic, fluid-flow, heat-transfer, combustion and pollutant-formation processes (Heywood, 2018).

An engine model or its sub-model being physical, in the sense of categorising models into physical and data-driven models, is not yet a guarantee of it being as accurate as current understanding of the phenomenon in question allows. Rather, the fidelity of

each sub-system must be fitted to the task in the same manner as whole engine models must be fitted to the ultimate objective. The combination and individual properties of these sub-systems then determine the overall computational efficiency of the complete model. As an engine model can ultimately only be considered as accurate as its weakest link (Heywood, 2018), the sub-models should generally be on par with each other to not waste computing power on actions that do not increase overall model accuracy.

A key characteristic of a physical engine model is the level of individual simulations. One way of categorising models is to consider the number of dimensions in which the simulations take place. These levels of simulation are zero-dimensional or 0D, one-dimensional (1D) and three-dimensional (3D). These dimensionalities refer to the number of independent space variables considered in the calculations. It is evident that an engine exists in a three-dimensional space, and thus its internal operations must happen in three dimensions as well. Therefore, anything other than a 3D model will represent a notably simplified version of the real world to varying degrees of fidelity.

2.1.1.1 0D models

Zero-dimensional (0D) models are a simplified approach to engine simulation where the spatial dimensions of the system are not considered. Instead of simulating the complex flow and distribution of gases within the engine, 0D models focus on the average properties of the gas within a control volume. This simplification makes them computationally efficient and useful for certain types of analysis (Heywood, 2018).

In the context of internal combustion engines, 0D models are often used to simulate the thermodynamic processes occurring within the engine cylinders. These models treat each cylinder as a single point where the mass and energy balance equations are applied to determine parameters such as pressure, temperature, and composition of the gas mixture. The key advantage of 0D models is their ability to quickly provide insights into the overall performance of the engine without the need for detailed spatial resolution. For example, in a 0D model, the combustion process might be represented by a heat

release rate equation, which describes how the energy released by the fuel combustion is transferred to the gas mixture in the cylinder.

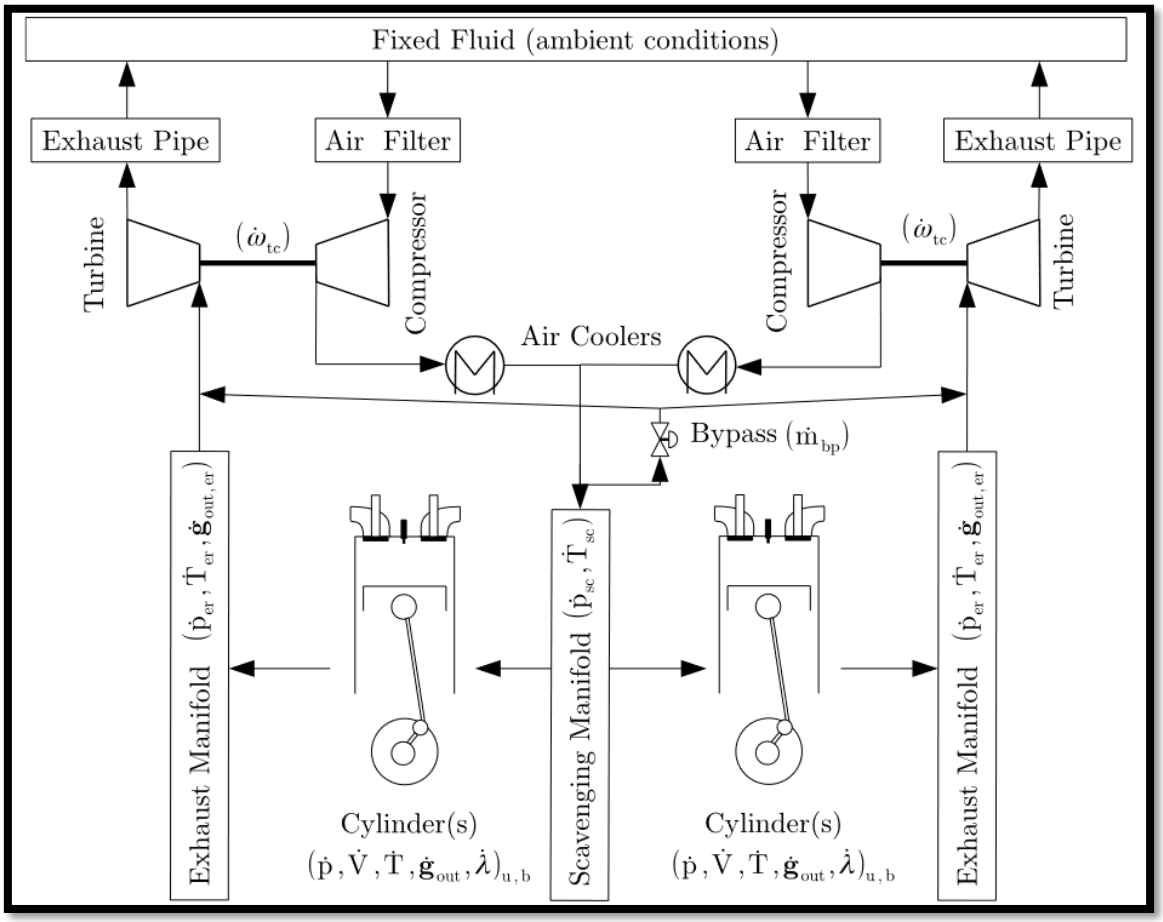

Figure 1. A layout of a 0D diesel engine model (Coraddu, Kalikatzarakis, et al., 2022)

At the core of 0D engine simulations is the concept of control volumes, which are used to model the engine or parts of it. These simulations calculate key parameters such as pressure, temperature, and composition of gases, either for the entire engine in single-zone models or for each control volume individually in multi-zonal models, such as the one described in Figure 1. This approach allows for the analysis of energy flows and transformations within the engine system without the computational complexity associated with models of more dimensions. The simplification to a zero-dimensional perspective significantly reduces computational requirements, making these models particularly

fit for real-time applications and preliminary design stages where quick iterations are necessary (Keskin et al., 2020).

In summary, 0D engine models consider the engine or its parts to be homogenous volumes which have a certain effect on the input. This simplification renders them computationally effective, which in turn favours their use in real-time and control applications as well as system-level analysis and optimisation. The main disadvantage of 0D models is their inability to predict spatial or transient effects.

2.1.1.2 1D models

Comparing to 0D and 3D models, 1D engine models include one-dimensional flow in the calculations and thus provide an essential middle ground between the two approaches. By doing so, 1D models can produce information on pressure fluctuations along a single axis. Thus, of the three spatial dimensions, two are simplified to a point where only their influence on a one-dimensional flow is considered. The remaining dimension then represents the flow of fuel mixture through the engine airpath.

As 1D models are still simplifications of three-dimensional processes, the choices made during the modelling process will have great impact on the model's accuracy and therefore ability to predict engine behaviour. One such choice is the selection of discretisation length. The higher the amount of infinitesimally thin planes perpendicular to the length of the airpath being approximated at once, the better the accuracy of the model will generally be. Reducing the discretisation length will then lead to models of better accuracy, but with a caveat of increased computational cost (Wurzenberger et al., 2013). Again comparing to 0D models, 1D models are not restricted into steady-state situations. Rather, as 1D models are capable of solving gas pressure, temperature and composition for each measure of a discretisation length along the airpath independently, they can be used for simulating transient behaviour as well. In simulation software, the 1D flow is represented as individual flow elements leading from the intake to the exhaust, as is illustrated in Figure 2.

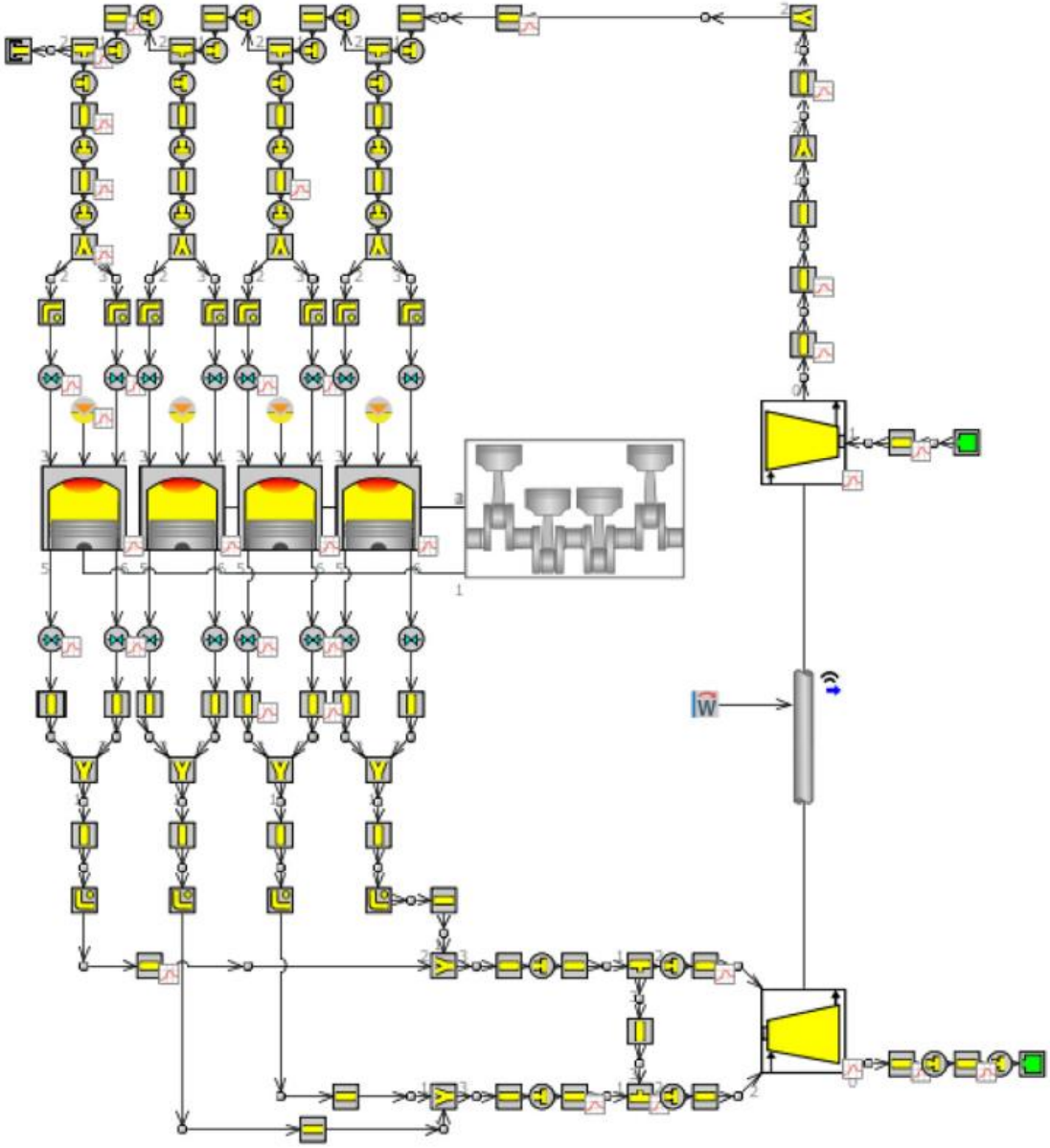

Figure 2. Overview of detailed 1D model representing the Wärtsilä 4L20 engine (Hautala et al., 2022)

One example of the effect of discretisation length on computing times can be found in the work of Hautala et al. (2022). The authors conducted a comprehensive study of balancing model fidelity and computational expense to bring the model significantly closer to being used as a part of a hybrid power plant digital twin. They initially used a fine

discretisation length of 50 mm of a detailed model, which accurately captured flow dynamics but was impractical for real-time control due to its high computational demands.

By increasing the discretisation length to 200 mm for the intake side and 300 mm for the exhaust, they were able to significantly reduce the real-time factor of the model to being three times faster than real time. The well-thought-out strategy used in combining flow elements allowed the model to maintain accuracy of within 5 % on selected criteria of the detailed model.

In the automotive industry, Wu and Li (Wu & Li, 2016) were able to conduct a somewhat similar investigation into the optimisation of 1D models for real-time applications by performing a hardware-in-the-loop bench test of a fast-running model to validate its effectiveness for rapid control prototyping. Their approach also involved simplifying a detailed 1D engine model by increasing the discretisation length to balance accuracy with computational efficiency. As this happened already in 2016, it is evident that different industries are adopting real-time-based workflows at significantly different paces.

2.1.1.3 3D-CFD models

Three-dimensional computational fluid dynamic (3D-CFD) simulations have a significant role in all engine modelling. In the current era of internal combustion engine development, the evolvements are very complex in nature and thus require proper combinations of different development tools (Keskin et al., 2020). In these combinations, 3D-CFD simulations serve the role of intricately describing the flow patterns, pressure gradients and temperature distributions of an engine in all three spatial dimensions. These quantities are solved numerically using the Navier-Stokes equations, the cornerstone of fluid dynamic simulations (Lucchini & Wright, 2020).

The Navier-Stokes equations are a set of partial differential equations (PDE) that describe the conservation of mass, momentum and energy in a fluid. In 3D-CFD simulations the

fluid is divided into a mesh of a discrete subsections. The equations are then used for solving the fluid's properties within these subsections and their interactions.

As 3D-CFD simulations are used for mimicking reality to the furthest possible extent, the technique serves a purpose apart from being a simulation device; it can also provide information of flow without the need for measurement. In the engine environment, this translates into ability to investigate processes such as turbulence, mixture formation, combustion processes and exhaust particle formation, that are nearly impossible to capture through the use of measuring devices (Keskin et al., 2020). Thus, 3D CFD simulations can serve as a virtual eye that allows comprehensive studies of existing designs as well as relatively cost-effective development of new ones.

One of the fundamental aspects that sets 3D-CFD simulations apart from other techniques is the detailed representation of engine geometry. The simulations are done using intricate 3D models of the engine airpath and all information regarding the shapes' effects on the flow is retained. At best, this includes everything from the intake manifold, combustion chamber and valves to the exhaust system. Thus, when certain criteria regarding evaluation of a model is met, the models can be used for aforementioned purposes with high confidence.

One party to suggest guidelines for proper evaluation is the American Institute of Aeronautics and Astronautics (AIAA) (AIAA, 1998). In their guide, the validation process is divided into two main parts: verification and validation. Verification sets a foundation for the process by focusing on the consistency and reliability of the model. The phase is carried out through a series of systematic checks against the benchmark ordinary and partial differential equations (ODE and PDE) that govern fluid behaviour. As a result of this, the credibility of a model is ensured. In the validation phase, results provided by the model are compared to experimental data measured from a physical system that the model is meant to represent. The number of validation test cases is heavily impacted by

both the nature of the application and the desired level of accuracy, and thus it is impossible to determine a single set of criteria to cover all situations.

2.1.1.4 Physical model applications

By addressing different approaches for engine physical modelling, it becomes evident that each one serves a different purpose and that the coexistence is justified by the needs of different applications. In Table 2, Isermann presents a summary of the three main approaches, their key qualities and common applications.

Table 2. Recreation of a table showing qualities and use cases of different engine modelling approaches (Isermann, 2014b)

	Zero-dimensional	**One-dimensional**	**3D**
Modelling method	quasi-steady flow models, filling and emptying method	method of characteristics, method of differences	CFD methods
Flow	(quasi-) steady	unsteady	unsteady
Model complexity	low	high	very high
Computing time	short	medium	large
Real-time capability	yes	no	no
Applications	parameter studies, control system design	engine design	engine design

A key takeaway is the common nature of all three approaches; they are all derived from physical phenomena, but with varying degrees of fidelity. The workflow of using such approaches is to model an engine's behaviour and then to validate it against experimental data, rather than using experimental data to make conclusions of the engine's internal processes. Thus, one way of finding the right approach for a given application

would be to increase fidelity by moving from 0D to 3D-CFD until the limits of computing power for the application are met. Looking at Table 2, one should note that the applications do not directly dictate which approach is to be used. Rather, the application will most probably have limited computational resources, which then dictate suitable modelling approaches.

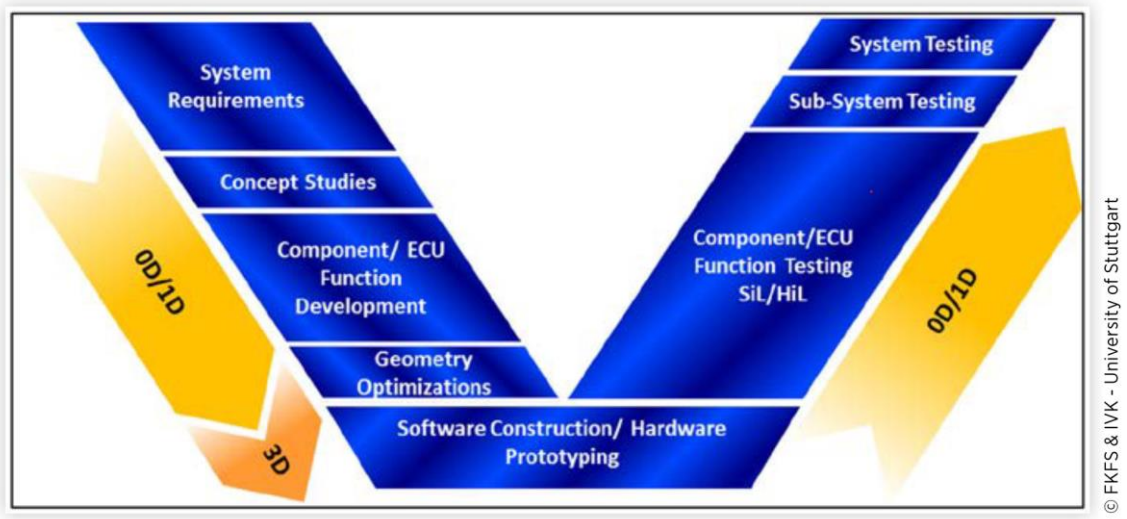

Figure 3. V-model for selecting correct levels of simulation during an engine development process (Keskin et al., 2020)

In Figure 3, Keskin et al. show a V-model of modelling approaches in different stages of engine development. The model is widely used in systems engineering and especially in automotive industry (Keskin et al., 2020). This explains the coexistence of various modelling approaches, with none being definitively superior to the others; instead, each is best suited for particular situations. 1D simulations are particularly well suited for early stages of the process, where the viability of a potential engine design is being conceptualised. As these phases are prone to demanding a multitude of case simulations for finding optimal combinations, 1D solutions coupled with 0D models for combustion chamber processes offer an almost sovereign solution (Keskin et al., 2020). 1D models hold their position in setting boundary conditions for flow components, but the scale tips in favour of 3D models when designing and prototyping them becomes critical.

When entering the testing stream or the right half of the V-model, model executing speed becomes yet again imperative. As commonly used hardware-in-the-loop (HiL) and software-in-the-loop (SiL) simulations are reliant on the simulation speed being close to or faster than real-time to function effectively, 3D CFD simulations are out of question and even 1D models are only applicable via use of special adaptations (Isermann et al., 1999; Keskin et al., 2020; Xia et al., 2017). This holds particularly true in the last stages of the V-model, where rapid simulations are necessitated by the need for extensive testing for system behaviour in different operating conditions.

Options outside the strict division into 0D, 1D and 3D models exist as well, with one example being quasi-dimensional models. In terms of both computing times and predictivity, they lie in the spectrum between 0D and higher-dimensional models. Even though the physical and chemical reactions of real combustion are dramatically reduced, quasi-dimensional models strive to give as physical a representation of the combustion phenomena as possible within the allowed computing times (Grill et al., 2010). Once more revisiting the concept of judging the correct simulation measures per application, a quasi-dimensional model must be as simple as possible but not at all simpler (Onorati & Montenegro, 2020).

The approach involves subdividing a component used in a 0D model into sub-volumes, with the behaviour of each sub-volume then being solved individually. Comparing to 0D models, quasi-dimensional models permit more detailed and realistic simulations especially in cases where the heterogeneity among sub-volumes cannot be ignored. One example of such a process is the oxidation of fuel in engine combustion chambers, which is homogeneous in terms of neither temperature nor composition (Bozza et al., 2020).

This subdivision enables a quasi-dimensional model to better capture spatial variations within the combustion chamber, which are critical for accurately modelling phenomena like flame propagation and emission formation. While still not as detailed as full 3D

simulations, quasi-dimensional models offer a pragmatic approach to capturing essential combustion characteristics with a reasonable computational load, making them well-suited for applications where a balance between model complexity and computational feasibility is necessary.

2.1.2 Data-driven models

The division between physical and data-driven engine models is simultaneously that of theoretical and experimental modelling. Physical engine modelling is the process of describing known processes and laws of nature as mathematical models, whereas data-driven models (DDM) draw from an extensive amount of documented real-world examples to predict the behaviour of a system. This is a process rather complementary than comparative to physical modelling, as the differences in paradigm render each method clearly best suited for certain types of applications. This leads to physical models and DDM:s mutually completing each other (Isermann, 2003).

A data-driven approach is also described as “bottom-up” by Onorati and Montenegro (Onorati & Montenegro, 2020), as they approach the problem of correctly depicting system behaviour by analysing the end results, or the “bottom”. Yet another description is provided by Isermann et al (1999), who call this process identification, as it relies on identifying the relations between input and output variables without necessarily revealing the underlying governing equations. For this reason, data-driven models are also called black-box models, as the fundamental dependencies between input and output factors might not be known (Coraddu, Oneto, et al., 2022).

As has been underlined previously, a key drawback of physical modelling is the slow and computationally intensive process of recreating physical phenomena, even when multiple dimensions are reduced through simplification. By shifting to the evidence-based approach of DDM:s, this element is disposed of and models can be applied with little to even no physical knowledge of the system (Coraddu et al., 2018).

Advancing in the process of creating a DDM comes the phase of data collection and pre-processing, as depicted in Figure 4. This phase is absolutely critical in setting a foundation for the models' accuracy and reliability. Cleaning, normalisation and transformation of the data ensures that the resulting dataset is consistent and subject to being effectively processed by the modelling algorithms. The correct algorithm choice depends on the task at hand, with some common techniques being neural networks, grid-based look-up tables and support vector machines.

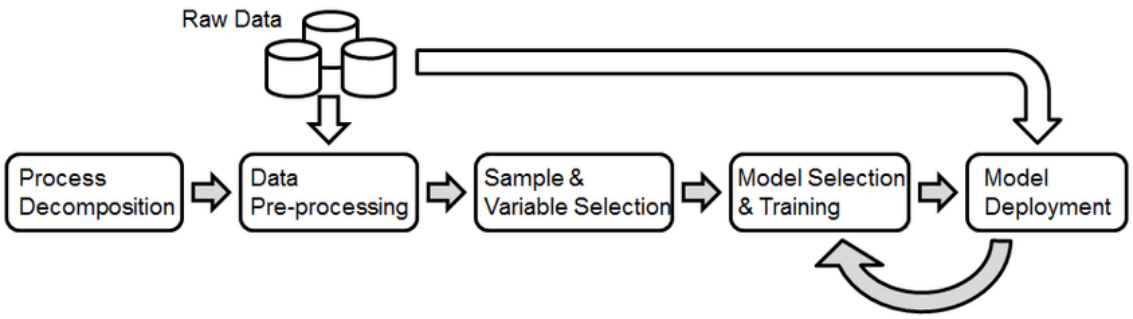

Figure 4. Workflow of data-driven modelling (Steindl et al., 2019).

Even though this route can lead to models of impressive accuracy, the data-driven approach does not come without its own caveats. For one, the modelling paradigm usually leads to parametric models without support from a corresponding physical interpretation (Coraddu, Oneto, et al., 2022). The models also require a great amount of historical data, which significantly hinders their usefulness in early stages of engine modelling, when runtime data is scarcer. These factors effectively rule the use of DDM:s out in any applications where deep understanding of the underlying phenomena is a significant target of the modelling endeavour. DDM:s are also worse suited for predictive modelling than physical models, as the models generally have no training data of situations outside the initial testing range.

2.1.3 Hybrid models

As stated before, physical and experimental modelling techniques generally complement each other. The nobility of solving an engine's physical behaviour to the point of

perfection and evolving the result into a functional model cannot be denied, but reality demands solutions that require far less effort in the development process. In addition to this, several parts of engines are still out of the reach of modern methods available for physical modelling due to the exact mathematical formulation of their processes not being known or the computational expense being too large to enable their use in control and diagnosis applications (Isermann, 2014a). On the other hand, data-driven models may fall short when the development has not yet reached a stage of prototyping and thus the data available for model training is scarce.

In a similar manner as quasi-dimensional modelling techniques lie between 3D and 0D models in the case of physical engine simulations, hybrid models fill the void between purely physical and experimental models and are a common result of modelling combustion engines (Isermann, 2014a). Due to the utilisation of both main techniques and the partial transparency, these models are also called grey-box models. In Figure 5, Isermann presents the continuum between the results of combining theoretical and experimental modelling.

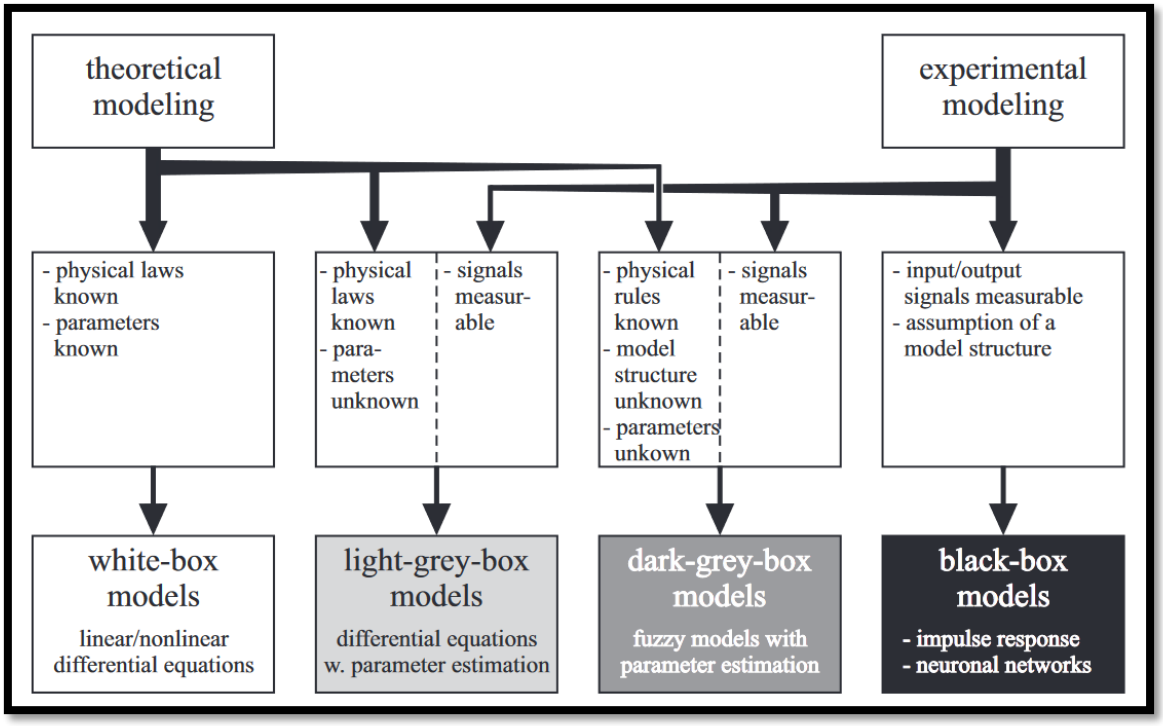

Figure 5. Continuum between white-box and black-box models (Isermann, 2014a).

The common denominator for hybrid models is the target of combining the best qualities of both physical and experimental modelling. As a relatively new approach to engine modelling and especially so in the realm of marine engines, an exhaustive framework for combining the modelling techniques is not yet readily available (Coraddu, Kalikatzarakis, et al., 2022). The technique has, however, been successfully deployed in multiple occasions. One example of this is the work of Mishra and Subbarao (2021).

In their study, Mishra and Subbarao compared a semi-empirical grey box approach based on modified knock integral method to a black-box data-driven supervised learning based artificial neural network approach. Both control-oriented approaches were able to track the selected control parameters of a reactivity controlled compression ignition (RCCI) engine in real-time. In this setting, the black-box model turned out to produce more accurate and reliable predictions across a wider range of operating conditions. The coefficients of determination for all measured control parameters surpassed those of the grey-box model in both calibration and validation datasets.

Even though the study found a black-box modelling method being better suited for this application, it is a step toward understanding the most optimal ways of combining advanced computational techniques and theoretical engine modelling. It also hints at the possibility of harnessing the virtues of both modelling techniques in an effective way, potentially enabling future endeavours to create models which reach the accuracy and stability of the introduced black-box model while shedding light to the underlying physics as well.

One way of lowering the computing cost of a physical model, while retaining its physical nature to some extent, is to use a mean value engine model (MVEM) instead of a detailed one. They are further categorised into ones that ignore and ones that consider engine manifold dynamics in their calculations (Coraddu, Kalikatzarakis, et al., 2022). An MVEM relies on using map based cylinder models instead of detailed ones (Gamma Technologies, 2023). These maps are typically derived from either experiments or simulations done on more detailed models. This approach can be expanded into replacing multiple cylinders with one mean-value element on multi-cylinder engine models.

An MVEM has the benefit of being computationally effective, but it is not well suited for applications where in-cylinder wave dynamics are of importance. Instead of actually solving the in-cylinder reactions, MVEMs only consider the inputs and outputs of the cylinder chamber at a certain resolution. One natural resolution is to calculate cylinder inputs and outputs for each full cycle instead of solving on a crank basis. Efforts have been made to combine MVEMs and 0D models into systems where one engine cylinder is modelled through a 0D approach and this model is then interfaced with the MVEM (Coraddu, Kalikatzarakis, et al., 2022).

In Figure 6, Mirfendreski et al. show the effect of reducing model complexity on the real-time factor of the system. The benchmarking has been done by utilising a detailed V6 TDI engine model and gradually reducing it. All simulations were done under equal

conditions with the commercial GT-Power flow simulation software running on an Intel Core i7 processor at a clock frequency of 2,80 GHz.

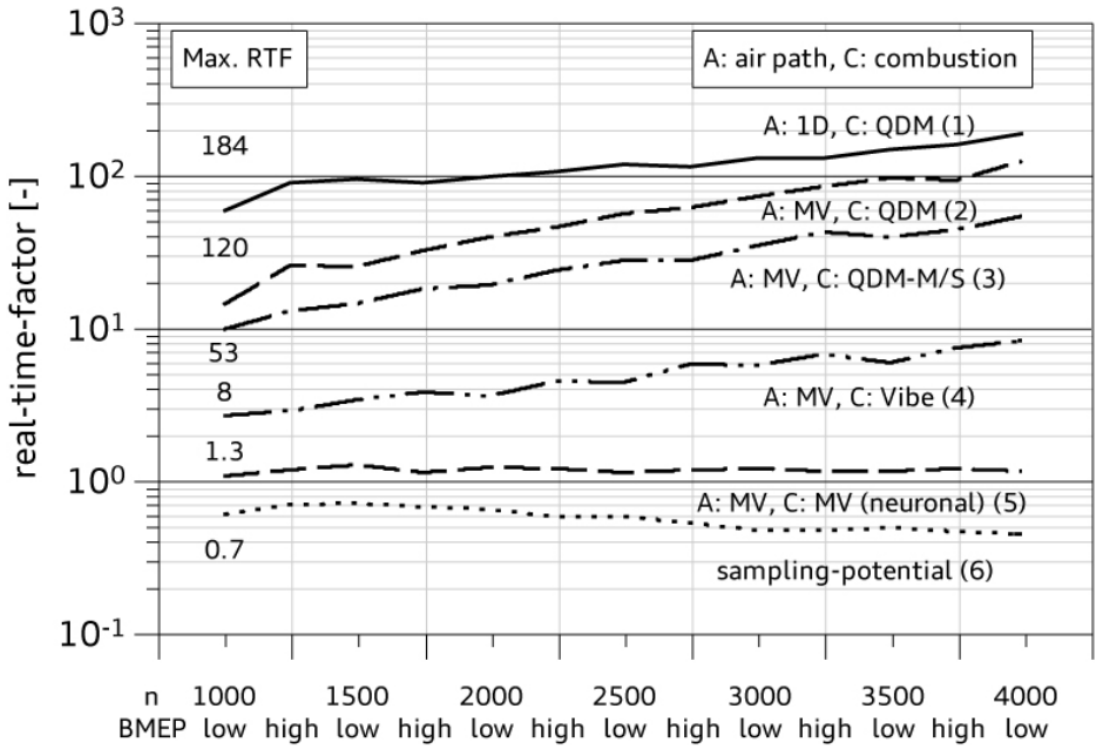

Figure 6. Real-time factors of different model concepts. QDM: quasi-dimensional, MV: mean value, M/S: master/slave cylinder (Mirfendreski et al., 2016)

This overview further underlines the significance of the used modelling concept to the computing times. Mirfendreski et al. go as far as to suggest that due to the hindered evolution of processor clock rates, one-dimensional real-time engine simulations will stay out of reach for the next decades.

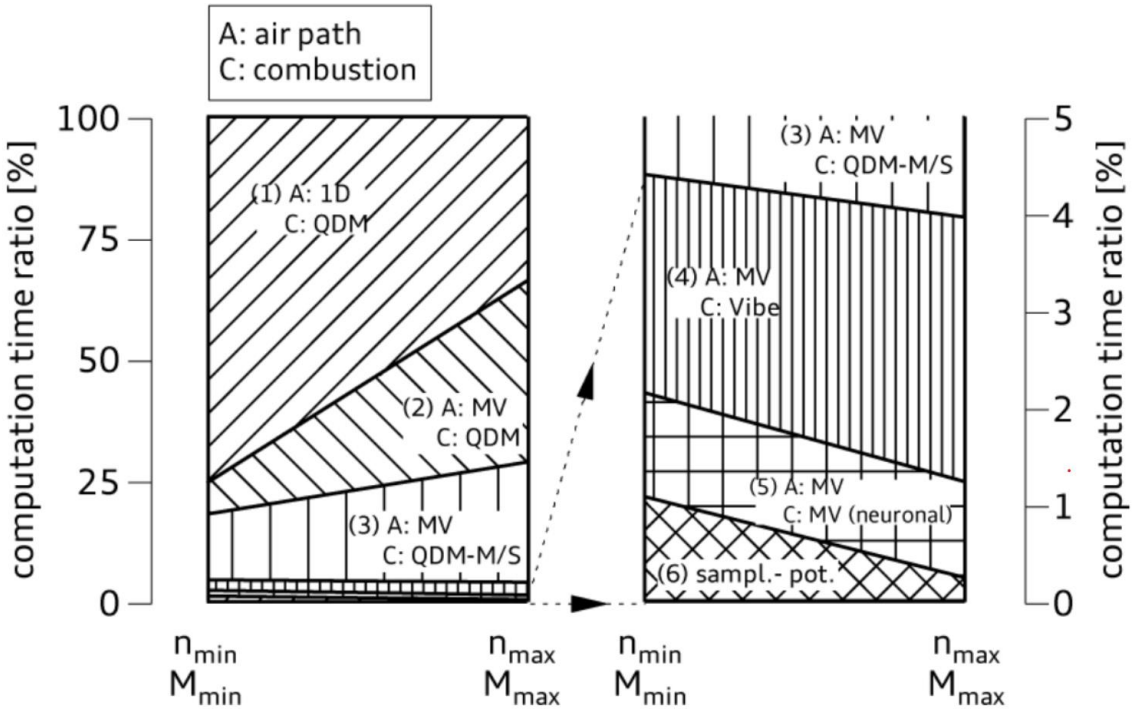

Figure 7. Relative computation times of alternative model-concepts. (QDM: quasi-dimensional, MV: mean value, M/S: master/slave cylinder) (Mirfendreski et al., 2016)

In Figure 7, Mirfendreski et al. have removed the processor as a variable and instead present the computation times in relation to those of the original detailed model. Assuming linear relation between processor clock rate and simulation speed, the detailed model's real-time factor of 184 would be covered by a processor running at 515,20 GHz.

2.1.4 Summary

In the realm of engine simulations, the endeavour to accurately represent engine dynamics creates an ever-present challenge of balancing the depth of detail with available computational resources. The challenge persists even during the advent of modern sophisticated modelling and simulation tools, which underlines the need for judicious decision-making in the process of selecting tools for a certain task.

From a research perspective, the transparency of a simulation model holds an intrinsic value. As highlighted by multiple authors of the field, the development and evaluation of a model should be guided by both its envisioned use and ultimate purpose. This

principle is further reinforced by the emphasis on clearly defined objectives and well-structured frameworks, which are key elements in the best practices of the modelling domain.

The continuous evolution in our understanding of the chemical and physical phenomena within internal combustion engines has facilitated the advent of sophisticated engine models. Despite these advancements, the increasing computing costs of high-fidelity engine models hinder the ambitious targets of encapsulating a detailed representation of all engine processes into one model. This constraint mandates a pragmatic approach to modelling endeavours, where the focus and thus the computational resources must be directed to the most critical processes per application. In Figure 8, Wurzenberger et al. present selected examples on a chart of simulation speed and model depth or predictability, underlining the trade-offs between modelling techniques.

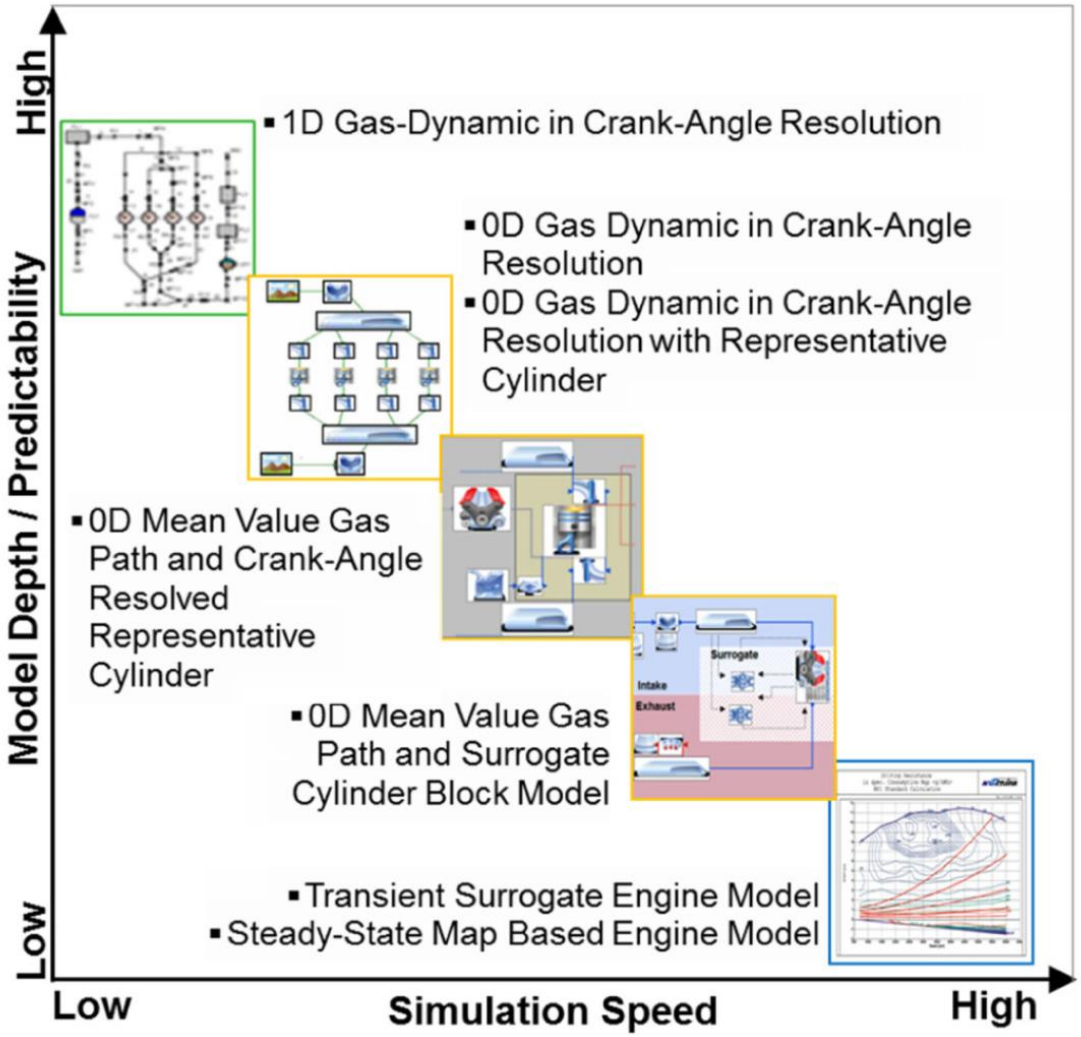

Figure 8. Engine modelling techniques from transient 1D gas-dynamic models to steady-state engine maps (Wurzenberger et al., 2013).

Within the spectrum of engine modelling approaches, three main methodologies emerge: physical, data-driven and hybrid modelling. Each approach operates on different principles and provides different advantages over each other. Physical models are rooted in the fundamental physics which govern the engine processes, whereas data-driven models utilise extensive datasets and machine learning algorithms to predict engine behaviour without a direct reliance on the underlying physical laws. Hybrid models, as the name suggests, combine elements of both physical and data-driven approaches, aiming to harness the predictive accuracy of machine learning while grounding

predictions in physical reality. This blending of methodologies seeks to optimise the strengths of each approach, offering a more versatile tool for engine simulation.

Physical models, characterised by their attention to the fundamental principles of thermodynamics, fluid dynamics, and chemical kinetics, form the cornerstone of traditional engine simulation. These models range from simple zero-dimensional approaches, which abstract the engine into a series of interconnected thermodynamic systems, to complex three-dimensional computational fluid dynamics (CFD) models that simulate the flow and heat transfer processes within the engine in great detail. The choice among these levels of complexity is governed by the specific needs of the simulation, balancing the desire for accuracy against the available computational resources.

On the other hand, data-driven models represent a paradigm shift towards empirical modelling, where the behaviour of the engine is predicted from historical data using statistical or machine learning techniques. This approach is particularly advantageous in scenarios where rapid prototyping and iterative design processes are essential, as it can significantly reduce the time and resources required to obtain practical conclusions. However, the efficacy of data-driven models relies on the quality and breadth of the data available, with their predictive power potentially limited to the operational conditions represented within the dataset.

Hybrid models are emerging as a solution to the limitations present in both physical and data-driven approaches. By integrating the robust, theory-based predictions of physical models with the adaptive, data-informed insights of machine learning, hybrid models seek to offer a balanced technique for engine simulation. This approach allows for the detailed simulation of known physical processes while also accommodating the dynamic adaptation to new data, and thus enhancing the model's predictive capability across a broader range of conditions.

In conclusion, the field of engine simulations stands at the intersection of theory and practice, where the pursuit of accuracy meets the pragmatic constraints of technology and resources. Whether through the detailed physical modelling of engine processes, the data-driven analysis of engine performance, or the integrated approach offered by hybrid models, the goal remains the same: to advance our understanding and capability in simulating the complex dynamics of internal combustion engines. As the field continues to evolve, the ongoing challenge will be to utilise these advancements in a way that maximises both the fidelity and efficiency of engine simulations, thereby contributing to the development of more efficient, powerful, and environmentally friendly engines.

2.2 Real-time simulations

The introduction of hybrid powertrains enhances complexity in energy systems, necessitating advanced tools for accurate control and monitoring. This complexity is particularly pronounced given the intermittent nature of renewable energy sources, such as solar and wind power. Efficiently harvesting these intermittent energy sources requires sophisticated real-time control and monitoring techniques (Chalal et al., 2023). In this context, real-time simulations stand out as an indispensable tool, especially for conducting Hardware-in-the-Loop (HiL) simulations, as is highlighted in the works of Wei Li et al. (2010) and Xia et al. (2017). These simulations are pivotal for the iterative development and optimisation processes and offer a practical approach to testing and refining hybrid powertrain systems under real-world conditions. Real-time simulations also lay the groundwork for the concept of digital twins, a topic explored in greater detail later in this work. As the electrical system has been shown to be real-time capable on the given hardware without adjustments by Söderäng et al. (2022), the focus of this section will be in engine real-time simulations.

2.2.1 Real-time simulation fundamentals

Engine models must be simplified in order for them to be suitable for use in real-time applications. When done correctly, this can be done while maintaining model accuracy

or predictiveness (Laurén et al., 2022). In Figure 9, model accuracy assessment is presented as a key phase in preparing a real-time model.

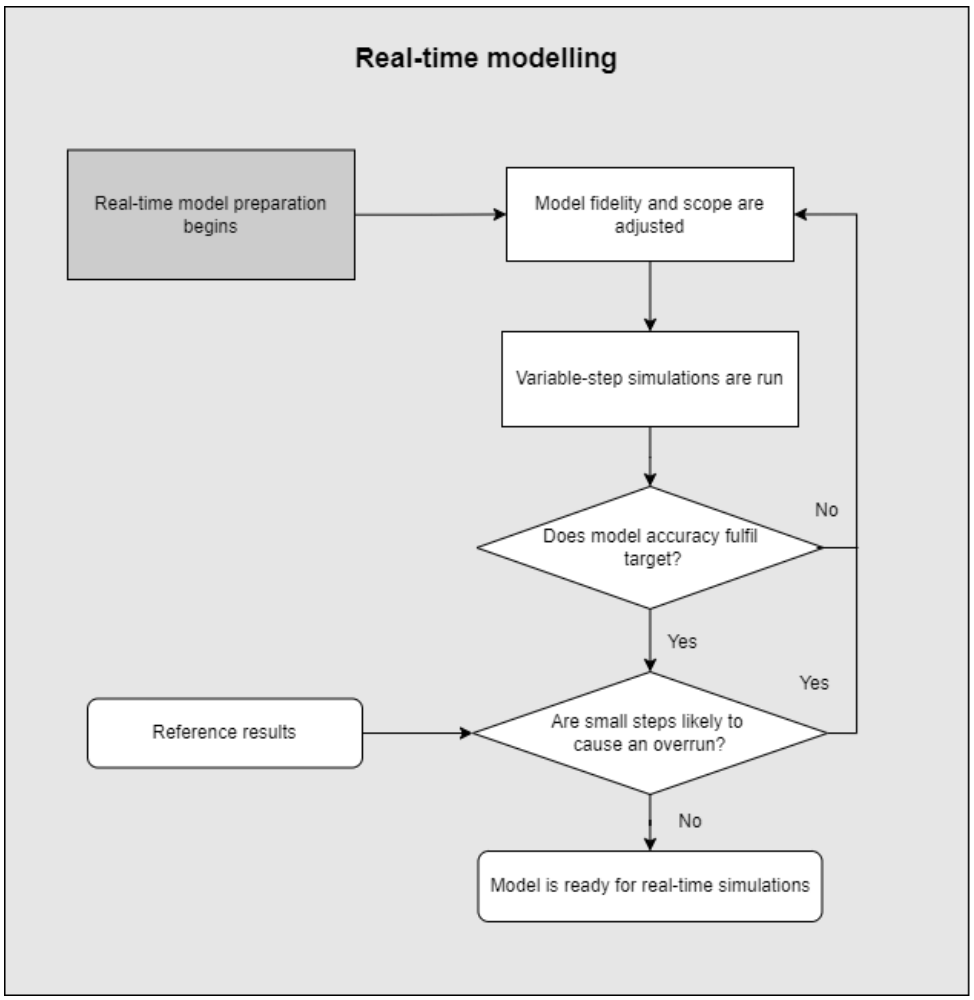

Figure 9. Real-time model preparation workflow.

A key aspect of real-time simulations (RTS) is distinguishing simulation time (also wall-clock time) from real-time. Simulation time refers to the time used by a computer to process a simulation, whereas real-time is aligned with the actual time perceived in the physical world. In an RTS, these timeframes are aligned. Figure 10 presents a situation where the task execution time T_e is shorter than the designated timestep t_n, which leads to results from the simulation timestep being available at the beginning of the next timestep.

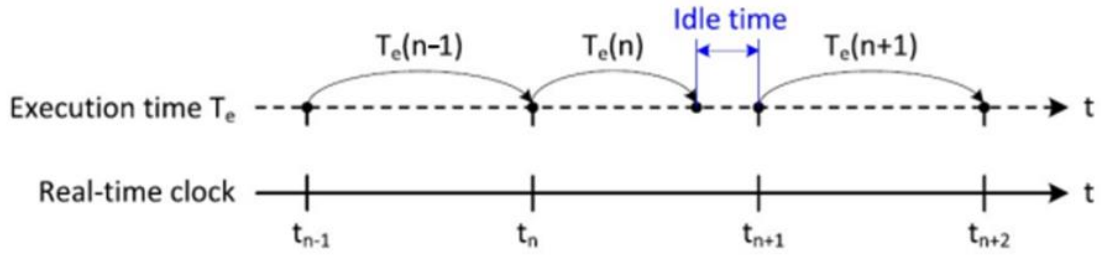

Figure 10. Real-time case (Minetti et al., 2023).

When a task execution time exceeds the available timestep, an overrun occurs, as is depicted in Figure 11.

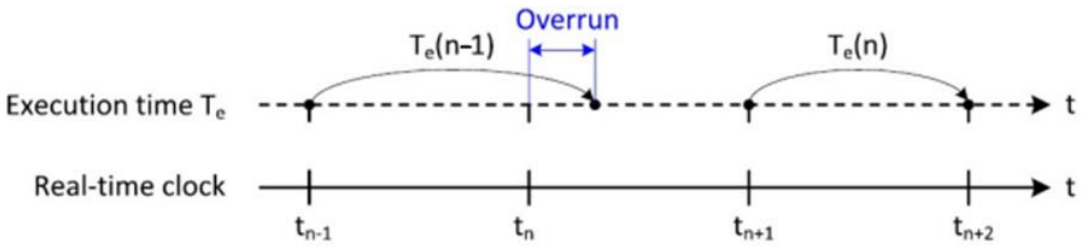

Figure 11. Overrun case (Minetti et al., 2023).

In the case of an overrun, the processor is still occupied at the beginning of the second time step, which inevitably delays the beginning of the second timestep and thus renders the system unsynchronised. Apart from this and especially as a part of a digital twin, a real-time system's performance is reliant on its ability to maintain timely interactions with all connected systems (Paquin et al., 2021).

The consequences of an overrun depend on the type of real-time system. The systems are divided into those of hard and soft real-time, with the main functional difference between them being their tolerance for overruns. In a hard real-time system, the consequences of an overrun are immediate and critical. Thus, it is imperative for each discrete time result to be published at the corresponding wall-clock time. Some typical instances of such systems are those used in aviation control and navigation, industrial automation

and control and robotics (Kavi et al., 2009). For example, in the case of power systems, there is a risk of overruns producing glitches in current and voltage waveforms, which may falsely hint at problems with the controller design. Such issues can be overcome by both optimising the simulation system for the application and implementing overrun detection within the real-time system (Paquin et al., 2021).

In the case of soft real-time systems, overruns do not cause catastrophic failure. Such systems are common in applications where the consequences of overruns usually present themselves as degraded simulation quality. Some examples can be found in consumer electronics and particularly video games, where the real-time requirement for human-machine interactions is evident. In this setting, overruns may cause glitches and thus weaken the sense of immersion and cause, but this is a minor inconvenience that the consumer must simply adapt to (Paquin et al., 2021).

2.2.2 Real-time operating systems

Due to the characteristics of a real-time system, their needs are fundamentally different than those of day-to-day desktop computing. This has led to the demand for dedicated real-time operating systems, which are purpose-built to support real-time operation. Instead of enabling a broad range of activities and easy user interaction as is the case with general-purpose operating systems (GPOS), real-time operating systems (RTOS) are engineered to be predictable, reliable and efficient (Garre et al., 2014).

As real-time systems are susceptible to catastrophically failing in the case of overruns, the design of an RTOS revolves around guaranteeing that tasks are executed within a specified timeframe (Liu, 2000). To this end, an RTOS must employ sophisticated scheduling algorithms which ensure that resources are directed correctly at all times. In contrast to a GPOS, where tasks are interrupted and swapped out based on time-sharing principles, RTOS scheduling prioritises jobs based on their urgency and impact on system functionality (Garre et al., 2014).

For the purposes of this thesis, it suffices to say that the RTOS utilised in the experimental section is a QNX Neutrino system embedded into a Speedgoat real-time target machine. QNX Neutrino is a POSIX-compliant RTOS commonly used for industrial purposes and which provides several scheduling policies (QNX, n.d.). For detailed explanations on the FIFO, round-robin and sporadic scheduling policies, the reader is directed to resources such as *Real-Time Systems (Liu, 2000)*.

2.2.3 Determining an engine model's real-time potential

In the realm of engine simulations, the type of model greatly affects the model's potential for RTS. It is noteworthy that, as a definition, a system being "real-time" only dictates a simulation step having to be computed in less time than the correspondent wall-clock time, but sets no numerical limits for how long either time can be. Thus, the RTS potential of a given model is determined by both the computational demands of the model and the desired model fidelity. In addition to this, in order to efficiently support hardware-in-the-loop (HiL) simulations, it does not suffice for the total computing time of a simulation to happen in less than the wall-clock time. Instead, each individual time step must be computed as fast or faster than the corresponding physical time span (Wurzenberger et al., 2013).

A real-time factor (RTF) is calculated by dividing the time spent for a simulation by the corresponding wall-clock time. Thus, an RTF of under 1 generally implies real-time capability. The earlier statement of a system being "real-time" not dictating the time step is especially true for engine simulations. Additional complexity is introduced by the measure of engine timing, which is typically measured in degrees of crank axle revolution. Thus, a complete engine cycle would correspond to 360 or 720 degrees for two-stroke and four-stroke engines respectively. This leads to the corresponding wall-clock time being dependent on engine speed, as engine speed dictates the amount of crank angle turns per unit of time. Another notable effect of this is that the higher the RPM, the smaller the sample time must be to accurately capture the combustion event.

For example, in Table 3, Wurzenberger et al. present the results from carrying out a simulation of an engine cycle using different engine modelling approaches. A four-cylinder turbocharged direct ignition gasoline engine with an engine displacement of 1,6 lt was modelled using eight different modelling approaches, with the focus being on comparing their computational performance. The employed models range from a detailed 1D model in crank-angle resolution to a map-based engine model. As the 1D-CRA model is crank angle-resolved, it is prone to having significant fluctuations in RTF during an engine cycle due to the effects of the combustion event.

Table 3. Real-time factors of different engine modelling approaches for a 1.6 lt four-cylinder turbocharged direct ignition gasoline engine (recreation of table by Wurzenberger et al., 2013).

Name	Model type	RTF
1D-CRA	1D gas-dynamic in crank-angle resolution	35
CRA4	0D gas dynamic in crank-angle resolution	0,92
CRA4M	0D gas dynamic in crank-angle resolution with representative cylinder	0,42
CRA1	0D mean value gas path and crank-angle resolved representative cylinder	0,3
RVM	0D mean value gas path and surrogate cylinder block model	0,28
INN	0D mean value gas path and surrogate cylinder block model	0,29
dINN	Transient surrogate engine model	0,04
Map	Steady-state map-based engine model	0,03

The 1D-CRA (1D gas-dynamic in crank-angle resolution) model is a high-fidelity approach that involves solving the transient Euler equations for mass, momentum, energy, and species balances along the engine's gas paths. This model captures the essential pressure wave dynamics within the engine, which are critical for accurately predicting the volumetric efficiency, turbocharger performance, and other gas exchange processes. The model employs finite volume discretisation combined with shock-capturing techniques to ensure numerical stability and accuracy, particularly important for high-speed engines with low swept volume. While highly detailed, this model typically requires substantial computational resources, making it less suited for real-time applications.

The CRA4 (0D gas dynamic in crank-angle resolution) model simplifies the engine into zero-dimensional gas dynamic elements resolved at crank-angle increments. This approach retains some detail of the gas exchange process while significantly reducing computational complexity compared to full 1D models. It involves averaging pressure, temperature, and species concentrations over the engine's volumes, which helps maintain computational efficiency. This model is useful for applications requiring a balance between detail and computational speed but may still struggle with real-time constraints.

CRA4M (0D gas dynamic in crank-angle resolution with representative cylinder) takes the CRA4 approach further by introducing a representative cylinder model. This method averages the behaviour of multiple cylinders into a single representative cylinder, reducing the overall computational load. This simplification maintains the advantages of crank-angle resolution for key processes while enhancing the model's applicability to real-time simulations by decreasing the number of computations needed per engine cycle. It is particularly useful in scenarios where engine cylinders operate under similar conditions.

The CRA1 (0D mean value gas path and crank-angle resolved representative cylinder) model combines a zero-dimensional mean value approach for the gas path with a crank-angle resolved representative cylinder model. This hybrid technique aims to achieve a balance between the computational efficiency of mean value models and the detailed dynamic response captured by crank-angle resolved simulations. It is designed to provide real-time capabilities by focusing computational efforts on the most critical engine processes while simplifying others.

The RVM (0D mean value gas path and surrogate cylinder block model) approach integrates a zero-dimensional mean value model for the gas path with a surrogate model for the cylinder block. This method replaces the detailed physical modelling of cylinder processes with data-driven surrogate functions that can rapidly compute the necessary outputs. These surrogates are typically trained on extensive datasets to predict torque, heat

flow, and other key variables, making the RVM model highly computationally efficient and suitable for real-time applications.

Similar to the RVM approach, the INN (0D mean value gas path and surrogate cylinder block model) model uses a zero-dimensional mean value gas path model combined with an intelligent neural network-based surrogate cylinder block model. This method leverages machine learning techniques to create accurate and fast-running surrogates for cylinder processes. The INN model is particularly effective in scenarios where traditional surrogate models might struggle, offering improved prediction capabilities and stability across a broader range of operating conditions.

The dINN (Transient surrogate engine model) model represents a further advancement in surrogate modelling, focusing on transient engine behaviour. It uses data-driven techniques to create surrogates that can accurately predict the dynamic response of the engine under varying operating conditions. This model is designed to provide high accuracy and computational speed, making it ideal for real-time simulations where detailed transient responses are required.

The Map (Steady-state map based engine model) model uses pre-calculated steady-state performance maps to represent engine behaviour. These maps are generated from detailed simulations or experimental data and provide quick look-up tables for engine variables such as torque, fuel consumption, and emissions. While extremely fast computationally, this approach is limited to the conditions represented in the maps and may not accurately capture transient behaviour or off-map conditions. It is best suited for applications requiring rapid, steady-state predictions with minimal computational overhead.

As the values in Table 3 are calculated as a mean value from a drive cycle simulation, the real-time capabilities are not yet clear. This is due to the mean value of an RTF over a simulation period being under 1 not excluding the possibility for the RTF of a single time step being over 1. Wurzenberger et al. also bring into attention that approximately 50 %

of CPU time in each single calculation time step must be reserved for data exchange and other activities. This further implies that for a model to be real-time capable in this particular hardware setting, the RTF must stay below 0,5. In Figure 12, the real-time factors of individual time steps for select modelling approaches are shown.

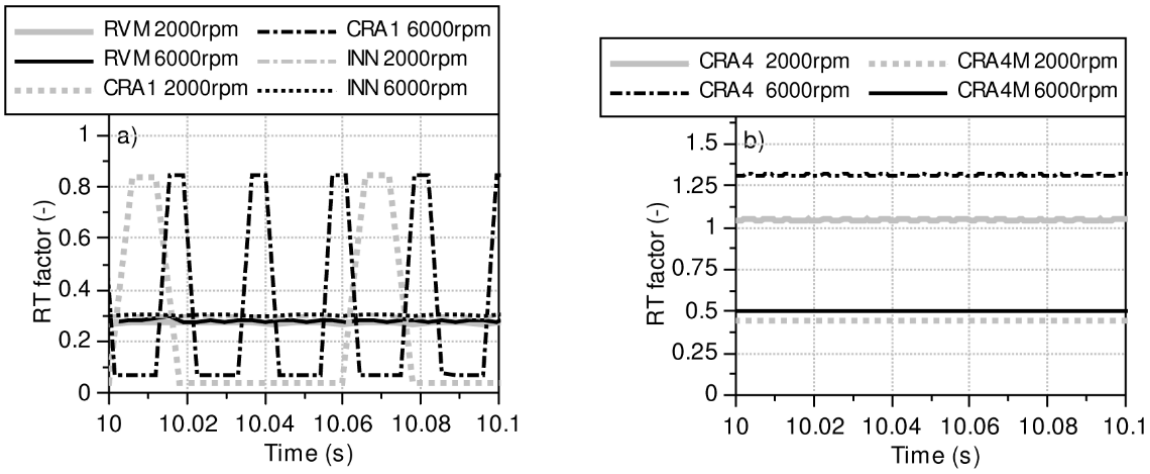

Figure 12. RTF profiles of different modelling approaches (Wurzenberger et al., 2013).

Upon inspecting the figure, it becomes clear that the RVM and INN approaches consistently fulfil the requisite of an RTF of under 0,5, with their RT profiles being relatively flat. However, it is equally clear that even though the drive cycle averaged RTF of the CRA1 approach is under 0,5, the expected spikes on the RT profile corresponding to the beginning of each engine cycle cause the RTF to regularly surpass the limit of 0,5. Thus, the CRA1 cannot be considered real-time capable in this application, as its real-time operation is not secured in each individual time step.

Equally clear is the effect of engine speed. When calculating the RTFs in Table 3, the integration step has been 1 ms for all mean-value air path models (CRA1, RVM, INN). However, the integration step for the crank-angle resolved models model is set to be 1 crank-angle degree (CAD). As this makes the integration step in milliseconds a function of engine speed, an additional caveat of stability issues was experienced at engine speeds below 1600 RPM as the step grew too large.

When comparing the CRA1 plot at different speeds in Figure 12, the significance of this discussion becomes evident. If the simulation time over which the average RTF is calculated was strictly presented as a unit of time, the CRA1 model would show a significantly larger average RTF at the higher engine speed. By using a whole engine cycle instead, the RTF better represents the model's capability for real-time and thus HiL applications. This also sets automotive and marine applications clearly apart. As marine engines generally operate in a significantly lower RPM range than automotive engines, the engine simulation time steps can be consistently set higher while keeping the same resolution in relation to a complete engine cycle.

As another key finding, the comparison of simulation results provided by the different approaches highlights the advantages of crank-angle resolved cylinder block models for MiL and SiL applications in early engine development. These models respond more accurately to changes in hardware or controls, which is crucial given the lack of experimental evidence in early phases. Detailed modelling approaches, while highly accurate, generally do not meet real-time constraints.

Lastly, Wurzenberger et al. conclude that no single global model suits all applications and that different development stages call for different model depths. Physics-based engine models with crank angle-resolved cylinder block descriptions are found to be the most promising, and continuous optimisation and improving computer hardware are expected to bring the approach closer to meeting HiL constraints.

2.3 Model-based design

Having matured over the course of several decades, model-based design (MBD) represents a transformative methodology in engineering. Closely related to model-based systems engineering (MBSE), which focuses on integrating models of different disciplines, MBD is rooted in the systematic use of mathematical models to guide the development, validation and optimisation of complex control systems. The framework has been

adopted across multiple industries, including power generation, automotive, and aerospace. The value of MBD lies in its ability to minimise physical prototyping by enabling engineers to conduct thorough testing and validation of system behaviour through digital simulations early in the design process. Thus, adopting the approach shifts the focus from physical prototypes and textual specifications to a model, which acts as a single source of truth throughout a development project (Aarenstrup, 2015).

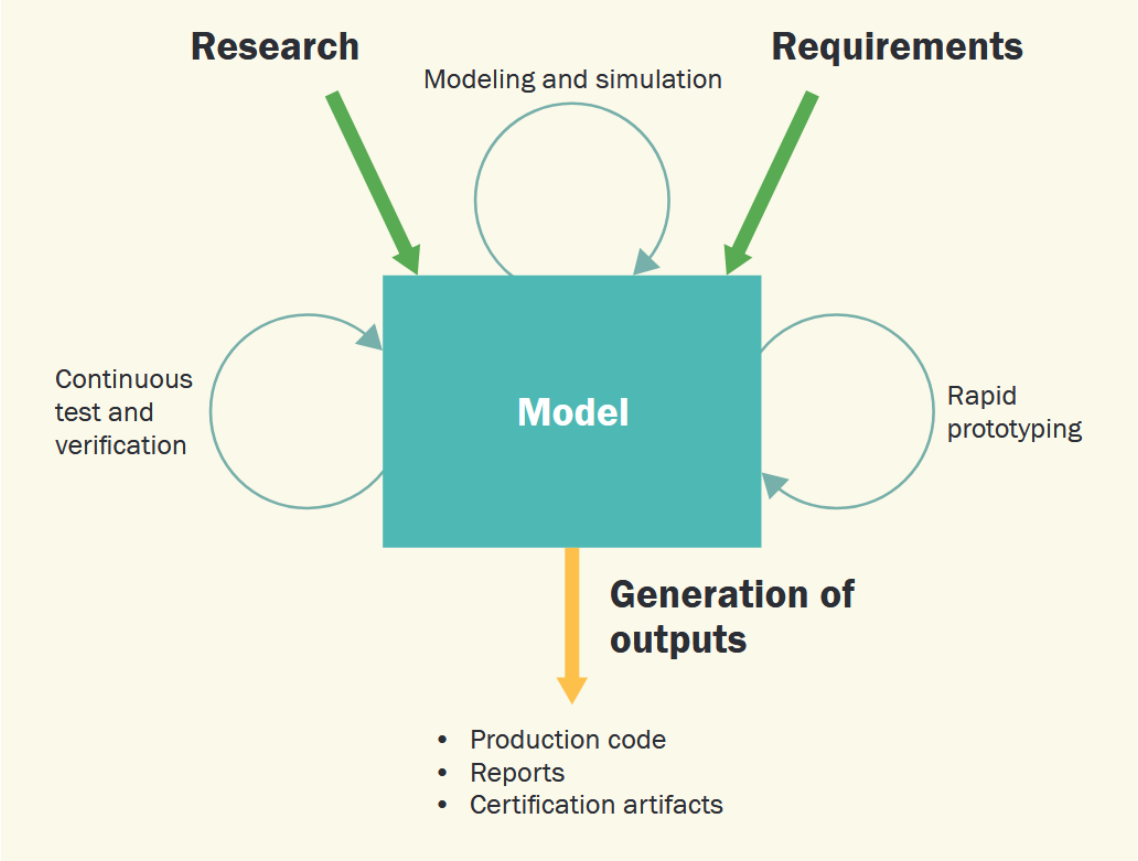

Figure 13. Model-based design workflow (Aarenstrup, 2015).

The workflow described in Figure 13 shows how – when it includes all components relevant to system behaviour – the model can serve as a source of many outputs, including HDL and C code and reports. The process of generating these outputs from the research and requirements feeding the model is constantly refined through modelling and simulation activities, continuous testing and verification along with rapid prototyping. This workflow also describes the most vital feature of MBD, which is the capacity to simulate

real-world dynamics with high fidelity, providing a virtual flexible environment for streamlined experimentation across various scenarios. Cutting from the effort needed in prototyping activities, MBD can also significantly decrease production time along with potentially improving overall quality of delivery (Smith et al., 2007).

In this context, technologies meant for enabling MBD workflows play a key role. Simulation-in-the-loop (SiL), model-in-the-loop (MiL) and hardware-in-the-loop (HiL) simulations and their subclasses are employed as a bridge between the real and virtual world, with each holding a relevant position in the evolution of a development project. In their respective order, the technologies begin from the purely virtual SiL activities and gradually evolve towards HiL simulations, where dedicated target machines operationally indistinguishable from their representees offer a significantly lighter solution for last-stage testing.

2.3.1 Key components of model-based design

Model-based design uses several technologies to streamline the design, simulation, and validation processes in complex engineering projects. Among these, the roles of Hardware-in-the-loop (HiL), Model-in-the-loop (MiL), and Software-in-the-loop (SiL) are key. Each component serves a specific function in validating system behaviour under simulated conditions in different development phases, thus reducing development time and enhancing the reliability and performance before deployment. In Figure 14, the differences between these methodologies are displayed.

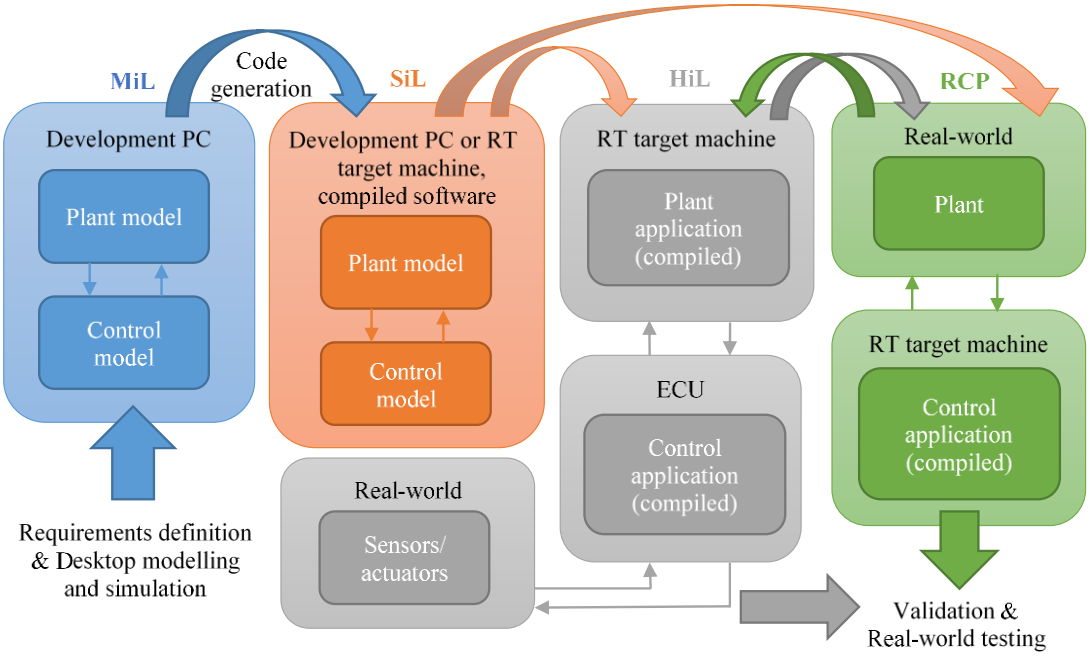

Figure 14. The main differences between MiL, SiL, HiL and RCP methodologies along with the key steps in the model-based design approach, ranging from design to realization.

MiL testing allows engineers to verify the accuracy and effectiveness of a model in simulating the behaviour of a system under various conditions. This loop involves simulating all components within a model to ensure that the interactions and outputs are as expected. As described in Figure 14, in a MiL setting the model is run on a development computer where both the plant model and the control model are executed together. MiL serves best as an early stage of validation, particularly useful when the system is too complex or costly to prototype physically. As a tool for early production phases, where prototypes are not generally available, it allows for early detection and rectification of potential issues (Plummer, 2006).

SiL simulations integrate software components into the model-based design framework to test software interactions within a simulated system environment. This component is important for validating the software's performance and its interaction with hardware and other system components. As shown in Figure 14, in a SiL setting, the plant model runs on a development PC or a real-time target machine, and the control model is

compiled software that interacts with the plant model. SiL ensures that the software performs reliably under various scenarios, thus reducing the need for revisions during later stages of development. By embedding the software within a simulated model, it can be verified that the software behaves as intended when interacting with the broader system, which is crucial for ensuring overall system reliability and performance (Dillaber et al., 2010).

As a next step in a development process, HiL simulation provides a way for testing the real-time response of control systems against a simulated system even without an existing prototype, and thus ensuring that the control systems can handle actual operating conditions. HiL has also been established as the standard method for testing electronic control units (ECU) (Kelemenová et al., 2013). As indicated in Figure 14, in a HiL setup, an application representing the plant is compiled and run on a real-time target machine, with which the control application undergoing testing interacts through real-world sensors and actuators.

In the physical world, a HiL system encompasses multiple interconnected systems. A typical HiL setup includes a real-time simulator (RTS), which replicates the behaviour of the physical system under test. This simulator uses high-performance processors, such as multi-core CPUs or FPGAs, to execute complex mathematical models at high speeds, ensuring the simulation runs in real-time. Interface hardware is used to connect the real-time simulator with the control hardware, converting signals from the simulator into formats that the hardware can process and vice versa. Common interfaces include analog-to-digital converters (ADC), digital-to-analog converters (DAC), and communication protocols like CAN or Ethernet. The actual control hardware, such as electronic control units (ECUs) in automotive applications, is then integrated into the HiL setup, interacting with the simulator as if it were connected to the actual physical system. These physical connection to real-world elements necessitate real-time capability of the simulation models (Knaus & Wurzenberger, 2020).

Finally, moving towards the finished physical system on the spectrum of development phases, lies rapid control prototyping (RCP). As illustrated in Figure 14, RCP takes the process one step further, deploying control applications onto real-time target machines and interfacing with actual physical plant systems for immediate testing and refinement. Once the system is set up and the toolchain has been established, this approach allows for streamlined final-stage testing and adjustment of the control system.

Another key concept in X-in-the-loop (XiL) simulations is the division between open-loop and closed-loop simulations. In control development, the target usually is to have a system capable of self-monitoring and control. This is done by building a feedback loop, where the system compares its current state to the desired one and self-adjusts its function to reach it. This is called closed-loop, as the control model and the plant model feed each other and the result of a simulation directly affects the next input. In contrast to this, open-loop simulations are vastly simplified. Instead of enabling two-way communication between models, they are fed a predetermined input. Manual adjustments can naturally take place after a simulation to match the intended output, but the system itself cannot, by definition, do this adjustment on behalf of the user. Note, that even though control development is widely used as an example of XiL activities, the “X” in the loop does not necessarily need to be the control unit (Xia et al., 2017).

As this thesis is limited to demonstrating the real-time capability of the models acquired from previous works, the SiL system will remain incomplete in the sense of the control system not being deployed. Rather, by recreating the open-loop simulations carried out by Söderäng et al. (2022) on upgraded hardware, the system’s potential for acting as a part of a XiL system will be determined.

2.4 Digital twins

The ultimate target of the ongoing work, which far surpasses the span of this thesis, is to build a digital twin of the hybrid power plant in question. Towards that end, it is necessary to establish an understanding of the concept. Being a relatively new concept in

the energy engineering landscape, digital twins are still in a phase where the term by itself is not enough to unambiguously explain the concept. The following part will aim to clarify the meaning of a digital twin at least in regard to the relevant project, and to give a brief overview of the concept's state in present time. As it will become clear, the state of this project does not yet justify calling it a digital twin, and thus the focus will be directed on the prerequisites rather than the implementation of one.

2.4.1 The concept of a digital twin

The concept of a digital twin is relatively new in the world of engineering, dating back to 2002 in the form of a presentation of the formation of a product lifecycle management (PLM) centre, as is described in Figure 15 (Grieves, 2023; Kritzinger et al., 2018; Singh et al., 2021).

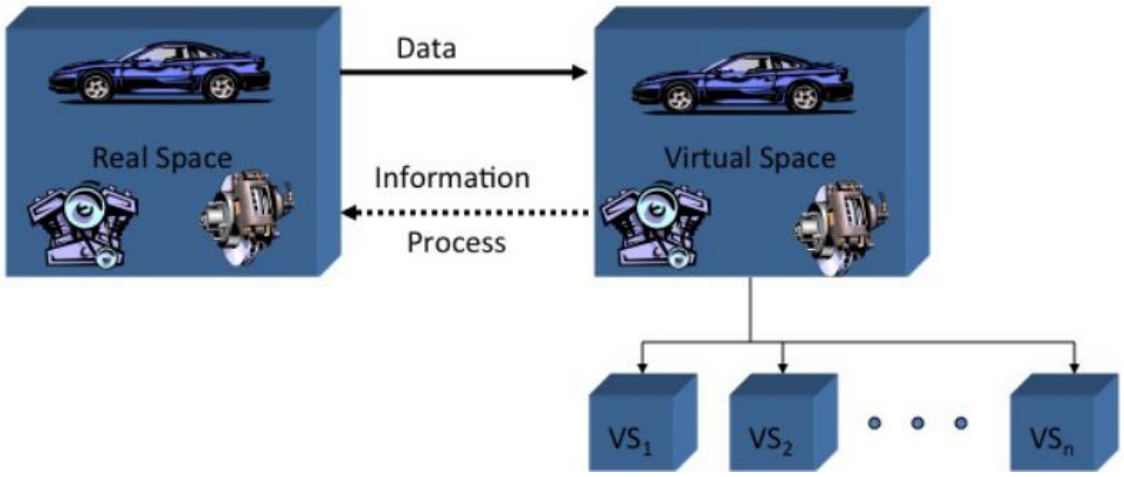

Figure 15. Conceptual ideal for product life management (Grieves, 2002).

The main takeaways of this initial description are the simultaneous and identical existence of both the physical and digital instance and, more importantly, the connections between them. The physical model would be constantly sending data on its current state, which would trigger its digital counterpart to predict alternative progressions and provide information on them. Even though Grieves is commonly credited for the original

idea of a digital twin, relatively similar concepts have been introduced in e.g. *Mirror Worlds* by David Gelernter (Gelernter, 1991).

The concept was later formalised as a digital twin by NASA, which defined the term "Digital twin" in a 2010 technical report (Shafto et al., 2010). The definition describes a comprehensive and highly realistic simulation that accurately replicates the life cycle of its physical counterpart by incorporating multiple physical sciences and scales along with probabilistic elements. To supplement this, sensor data from the vehicle's on-board systems is fed to the digital twin, which in turn recommends changes in mission profile to increase both the life span and the probability of mission success. This definition was later clarified and reformulated by Glaessgen and Stargel (Glaessgen & Stargel, 2012) in an aviation setting: "A Digital Twin is an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin. "

After these initial steps, multiple definitions for a digital twin have emerged in the literature, with their variety and occasional vagueness posing a potential hindrance for advancements of the technology (Heluany & Gkioulos, 2024; Jones et al., 2020; Singh et al., 2021; Tao et al., 2018; Wright & Davidson, 2020). Regardless, the fundamental idea of a digital twin being a pair of matching physical and digital entities exchanging data in real time is well supported by literature (Singh et al., 2021; Tao et al., 2018; Wright & Davidson, 2020). For instances which fail to meet these criteria of a digital twin, related concepts of digital shadows and digital models are often included in discussion. Outlined by Kritzinger et al. (2018) and described in Figure 16, a digital shadow differs from a digital model by having automatic data flow only from the physical object to its digital counterpart, which allows the digital shadow to adjust its function according to real-life parameters.

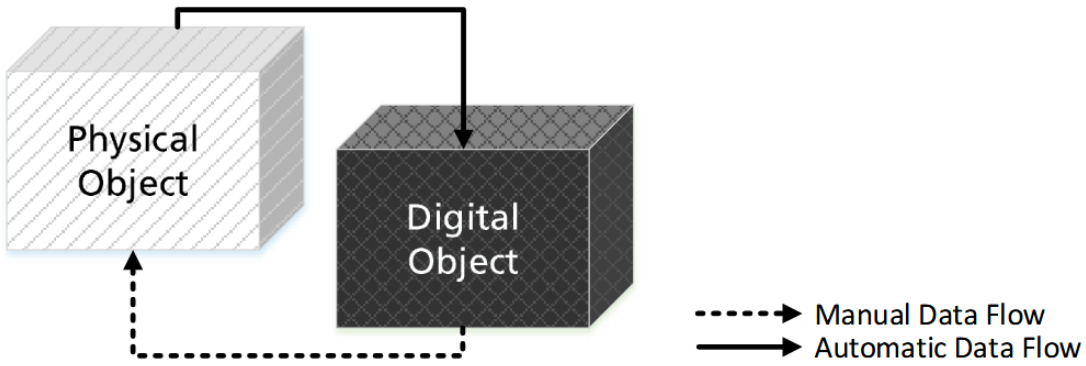

Figure 16. Data flow in a digital shadow (Kritzinger et al., 2018).

Respectively, a digital model would require manual data flow in both directions for the physical and digital counterparts to stay synchronised, which underlines the fundamental difference between a digital twin and a digital model. Rather than being a more or less accurate representation of a system's function at a certain time, a digital twin is a living, breathing, constantly evolving counterpart to its physical twin and mirrors its function accurately throughout its lifecycle.

2.4.2 Current and future applications of digital twins in the energy sector

Digital twins are currently applied in multiple fields, of which Crespi et al. (2023) focus on seven: production, energy, buildings, IT applications, education, health sciences and society. According to Crespi et al., each of these seven fields currently makes use of the concept in different ways, which renders the current use cases fragmented and specialised. In the field of energy production, publications on digital twin applications are scarce, despite the trend of them being a necessary future step in optimising large generators (Ghenai et al., 2022). Ghenai et al. further suggest that digital twins in the energy sector will provide a significant advantage in addressing energy and climate change challenges.

A recent review by do Amaral et al. (2023) takes a refreshed look into the state-of-the-art applications of digital twins in the energy industry, with an emphasis on analysing the entire energy supply chain. They suggest digital twins in the energy sector to bring

several advantages, along them holistic and detailed systems analysis, reduction of energy consumption and decreased CO2 emissions and real-time control of complex systems. Noise handling and communication latency, along other factors, are listed as possible issues.

Both Ghenai et al. and do Amaral et al. agree on the significance and potential of digital twin technology in the energy sector. The consensus is that current research trends point towards recognition of digital twins as indispensable tools for advancing the energy sector's efficiency, sustainability, and adaptability to changing environmental conditions. Also agreed upon is the need to address the potential data management issues and processing capabilities for real-time data processing. Altogether, both sets of authors expect digital twins to play a vital role in future energy systems.

2.5 Related work

As mentioned before, many techniques described in this thesis are already in common use in the automotive industry. Progression is also made in the energy and marine industries, but a fully physics-based digital twin of a hybrid power plant is not known to exist (Söderäng et al., 2022). For example, as Söderäng et al. (2022) point out, Perabo et al. (2020) presented a digital twin of a ship’s AC power and propulsion systems, but the representation of the engine was done by a simple transfer function. Following their overview of the state-of-the-art at the time, Söderäng et al. suggest that the lack of hybrid power plant digital twins incorporating a physics-based engine model is hindering development of grid- or vessel-level control strategies.

After the work of Söderäng et al., some endeavours in the field have taken place, with none still fulfilling the need for a physics-based digital twin of a hybrid powertrain. Madusanka et al. (2023) provide a comprehensive review of Digital Twin (DT) technology in the maritime sector, tracing its origins, current applications, and future trends. DT applications in the maritime domain are diverse, encompassing monitoring, simulation, and predictive maintenance. The slower adoption of DT technology in the maritime industry

compared to sectors like aerospace and manufacturing is proposed to originate from key challenges such as inadequate infrastructure, connectivity issues at sea, and concerns about data security. Despite these challenges, the review outlines significant future trends. The integration of DT technology with artificial intelligence (AI) and advancements in communication technologies are expected to enhance maritime operations' efficiency and safety. These developments are projected to enable more sophisticated predictive maintenance, reduce operational costs, and improve overall system reliability.

Hu et al. (2024) present an in-depth study on developing a digital twin model for in-cylinder combustion in diesel engines using a data-driven approach. This quasi-dimensional model combines physical principles with advanced data-driven techniques to achieve high predictive accuracy and real-time operational capabilities, combining the Wiebe function with a neural network.

The authors detail the methodology for constructing the digital twin, emphasizing the integration of data-driven methods like the snake optimization algorithm (SO) and convolutional neural network-bidirectional long short-term memory (CNN–Bi-LSTM) networks. These techniques enhance the model's ability to predict combustion parameters under various operating conditions. Through experimental validation, good predictive performance could be confirmed.

However, the implementation of the model is not done without challenges. The requirements for extensive data collection and processing capabilities pose significant hurdles, as is typical for data-driven models. Furthermore, in the context of hybrid power plants, the focus lying on a combustion digital twin is still far narrower than what is needed for optimising hybrid powertrain behaviour.

In their review, Mauro and Kana (Mauro & Kana, 2023) point out that most digital twin activities in the maritime world have so far been directed towards predictive maintenance. Regarding digital twins focusing on the operational phase, most works fall far

behind of acting as a holistic representation of the ship power system. Rather, literature on the subject revolve around models which could be incorporated into such a digital twin. The authors were also able to recognise some instances of the term “digital twin” being erroneously used for data-based or system engineering models.

Through this overview, it becomes clear that digital twins of hybrid power plants are practically non-existent in the literature. This is especially true when considering the self-imposed real-time targets of the one being developed in University of Vaasa. The target of time steps equivalent to single-digit changes in crank angle are in a completely different scope than the definition of do Amaral et al. (2023), which considers DTs updated at an interval of an hour “near real-time”.

3 Methodology

The main tasks involved in answering the research questions will revolve around introducing the models used by Söderäng et al. to the upgraded simulation framework. This process is then followed by a series of simulations to first validate the upgraded models, then to systematically find the limits of the setting.

The need for upgrades is dictated by the requirements of the Speedgoat system in relation to Matlab and Simulink, which are used for coupling the engine and electrical models. The study carried out by Söderäng et al. was done on Matlab R2017a and an engine model created in GT-Power v2019. As Speedgoat has since transitioned to the use of a QNX-based real-time operating system, the Speedgoat Mobile real-time target machine is limited to using Matlab versions of R2021a and above (*Mobile Real-Time Target Machine | Speedgoat*, n.d.). In regard of the engine model, documentation of and support for running models created in GT-Power in this environment are made available from software version v2023 onwards.

This work aims to take a step towards creating a digital twin of a grid-connected hybrid powertrain situated in the Vaasa Energy Lab. The power plant and its sub-systems as well as their modelling processes have been documented and described in detail in earlier works (Hautala et al., 2022; Söderäng et al., 2022).

3.1 Engine model

The test engine is a research version of a four-cylinder in-line engine with a 200 mm bore, the Wärtsilä 4L20. The basic specifications for the engine are presented in Table 4. For a more detailed description, the reader is directed to the Wärtsilä 20 Product Guide (Wärtsilä Finland Oy, 2020).

Table 4. Research engine specifications.

Wärtsilä 4L20 research engine specifications	
Cylinder configuration	Four cylinder inline
Bore	200 mm
Stroke	280 mm
Swept volume per cylinder	8 800 cm^3
Compression ratio	16:1
Rated speed	1000 RPM
Brake power	848 kW
Fuel system	Common rail
Turbocharger	ABB TPS48E01
Valve system	four valves/cylinder

For the purposes of this work, it will suffice to outline the original engine model from the perspective of computational load. Likewise, the process of reducing the engine model has been carried out and extensively documented by Hautala et al., so the main focus will be on the real-time potential achieved by the reduction.

3.1.1 Original 1D model

The engine has been originally modelled by Wärtsilä Finland, that created a detailed 1D model for research and development purposes. The model was later refined and tuned to better match the particularities of the research engine. It has been discretised through creating individual flow elements for all bends, contractions and flow splits. The 3D-CAD drawings for each part have been utilised in the process of further solidifying the close match between the model and the real world (Hautala et al., 2022).

The test engine has been modelled in the GT-suite environment, a sophisticated engineering simulation software that specialises in engine, powertrain and vehicle systems. GT-suite encompasses physics-based libraries for analysing phenomena such as engine dynamics, combustion processes and acoustics. In this application, GT-suite has been used by Wärtsilä Finland particularly to create a 1D flow model of the research engine. The combustion phenomena are handled in a simplified zero-dimensional manner and the turbocharger submodels are map-based. To further reduce computational load, any emission sub-models have been excluded from the in-cylinder model. This approach

allows for efficient computing while still accurately depicting parameters such as fluid flow, pressure changes and temperature variations along a single spatial dimension. Figure 17 shows an overview of the flow components with a total of 181 flow components with which the model completes a steady-state case in 77 seconds.

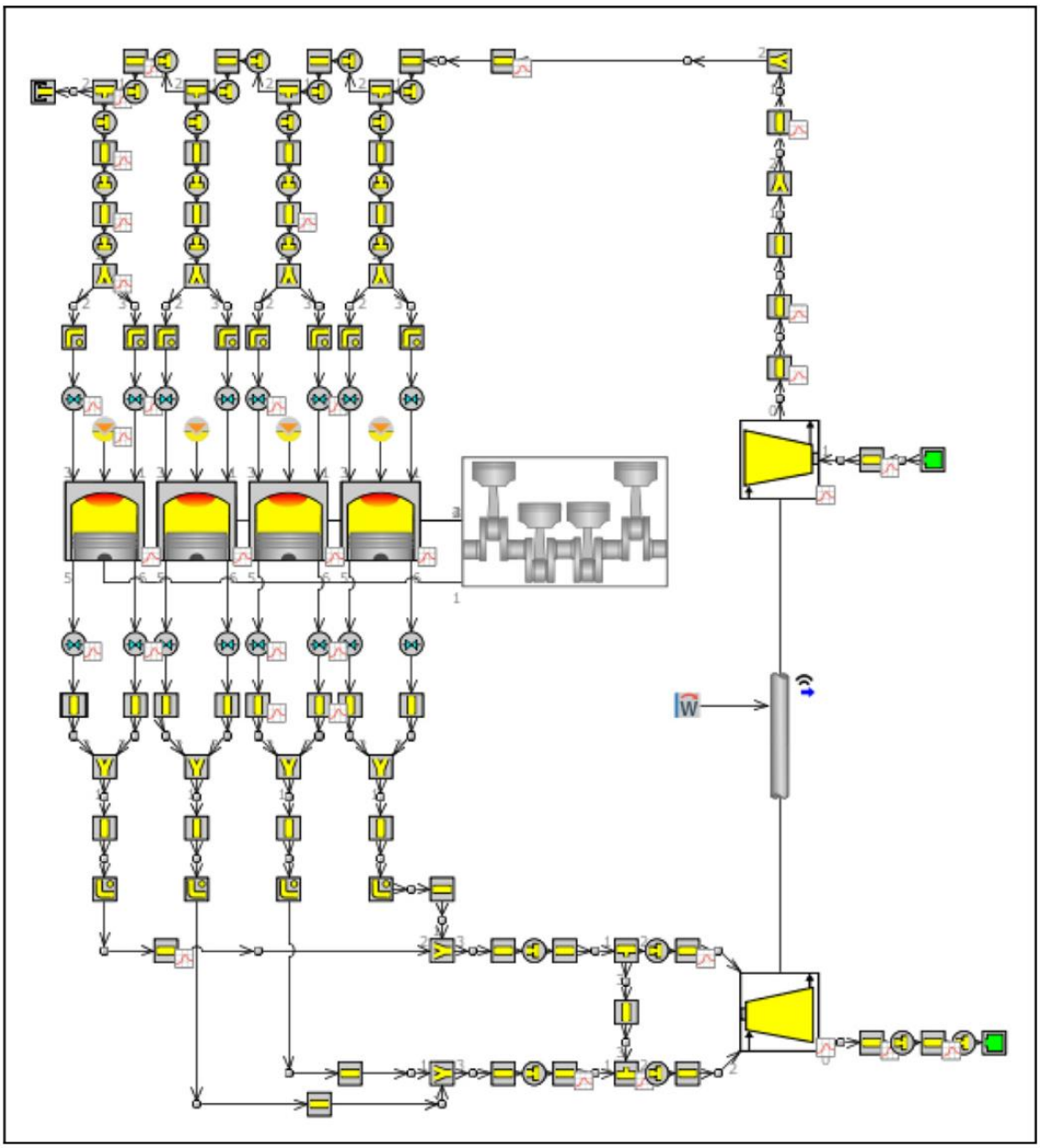

Figure 17. Overview of 1D model representing the Wärtsilä 4L20 research engine (Hautala et al., 2022).

3.1.2 Reduction to FRM

To adapt the model to the inescapable requirements of real-time applications, Hautala et al. set out to reduce the original 1D engine model into a fast-running model (FRM). The need for this was dictated by the original 1D model having calculation times approximately six times longer than real-time. An option alternative to this would have been to reduce the model into a map-based mean-value model (MVM), which would have sacrificed the predictiveness of the model to an extent. As retaining the physics and predictiveness of the model was deemed essential for enabling future endeavours, the choice was clear. Figure 18 describes the planned reduction and simulation process.

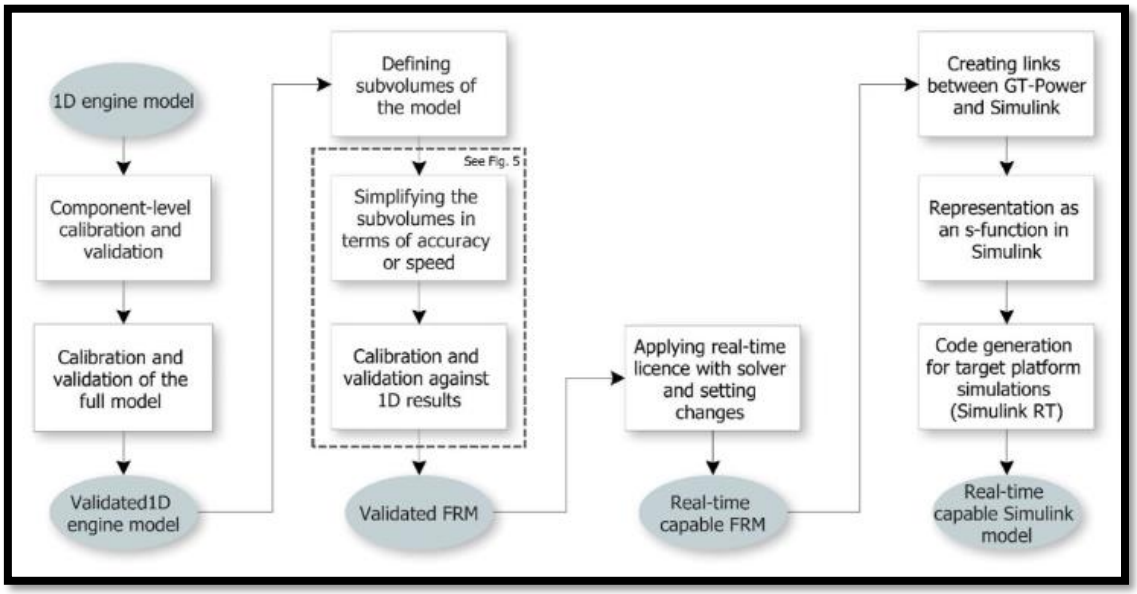

Figure 18. Real-time capable engine model development chart (Hautala et al., 2022).

The process of reducing the model into an FRM began with the strategical aggregation of individual flow elements into larger subsystems. This approach was carefully balanced to retain the ability to accurately reproduce pressure pulsations. Alongside this, the discretisation lengths of some parts were extended. The decision on which parts to model independently was made on the basis of their relevance in pressure pulsation

reproduction. Elements deemed less critical in this aspect would then be aggregated to streamline the model while preventing loss of essential functionalities.

This was directly followed by the simplification of certain subsystems in terms of their resolution and accuracy. As a rule of thumb, all passive flow components (e.g. pipes, bends and orifices) between active components were merged into one. Some exceptions were made, such as with the intake manifold, where the individual runners were retained in order to correctly depict cylinder aspiration efficiency. For the exhaust side, all components were merged, but the twin-turbine entry for correctly recreating turbocharger efficiency was maintained.

All remaining subsystems were then simplified in terms of accuracy by allowing simpler approaches for calculating heat losses and friction instead of very detailed and predictive ones. However, this strategy was only carried out in cases where sensitivity analysis suggested it to only have a minor effect on model predictiveness.

As a result of this, the model originally depicted in Figure 17 was reduced into one shown in Figure 19.

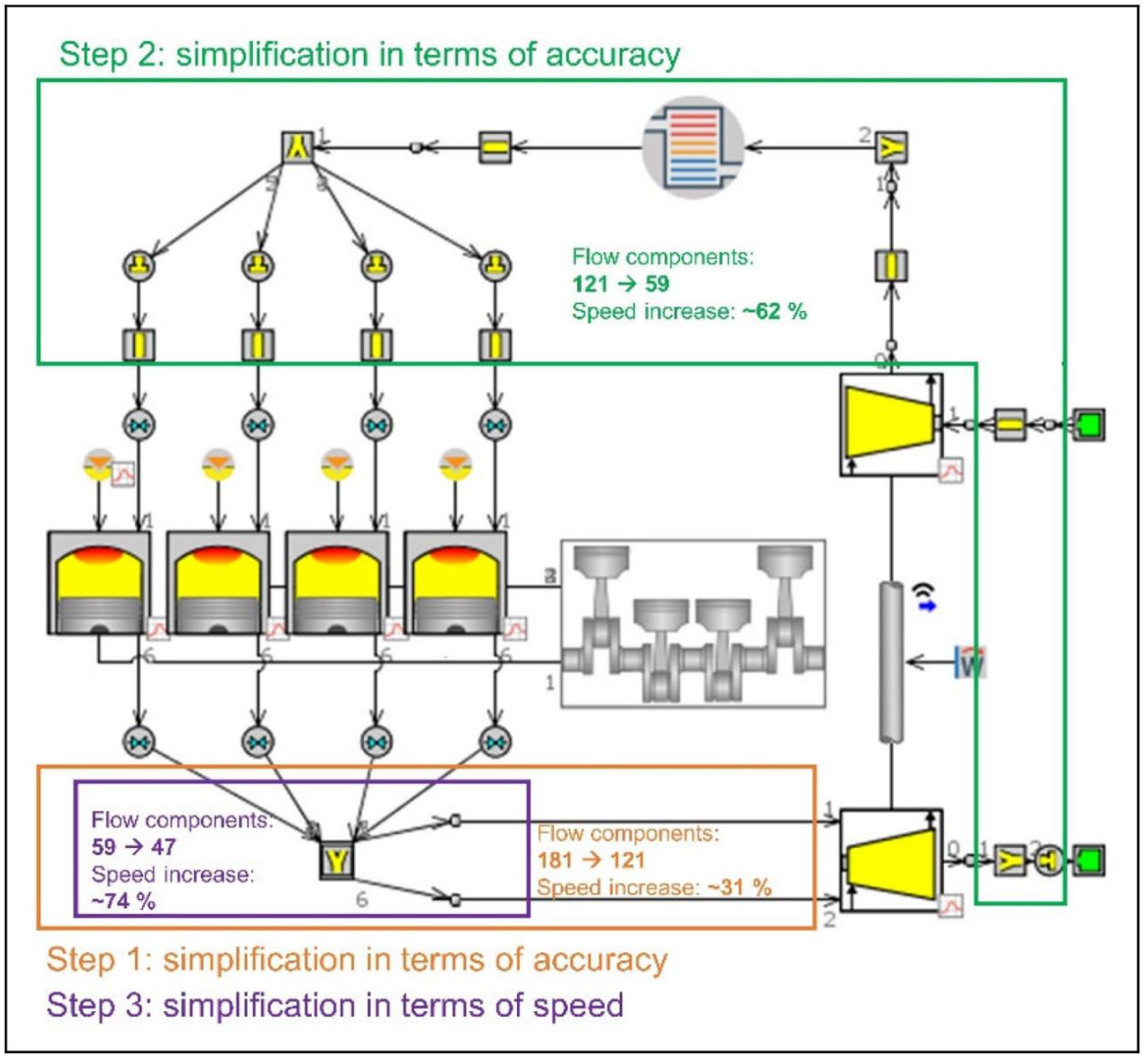

Figure 19. 1D FRM of the Wärtsilä 4L20 with outlines of main model reduction steps (Hautala et al., 2022).

With a 74 % reduction in the number of flow components, the model was capable of achieving significantly reduced computing times while still using the Runge-Kutta solver. For the original detailed model, the user-imposed maximum time step was set to 1 CA degree to ensure maximal accuracy. However, for the fast-running model (FRM), this constraint was relaxed to match the time needed for one full engine cycle in order to support fast run times. These changes were enough to reduce the real-time factor from roughly six to one to two, depending on the load point.

As the real-time factor must consistently stay below one for the model to fulfil real-time requirements, further steps to streamline the model were taken. By migrating to a GT-Suite-RT license and thus to an explicit Euler-based real-time solver, the model could be consistently run with a real-time factor of 0,35 on the development computer. Through a validation process discussed in detail by Hautala et al., the resulting real-time executable compilation was shown to also be well within targeted accuracy limits.

Upon deploying the acquired RT-executable model onto the target platform, real-time capability was lost due to the inferior computing power of the target computer. The target computer used was a repurposed personal computer with an Intel Core Q9400 CPU with significantly less computing power than the development computer running an Intel Core i7-8750H. The lack of computing power caused the need to increase the sample step size to 1 s, which expectedly sacrificed detailed dynamics of in-cylinder processes.

3.2 Combined hybrid power plant model

In a direct continuation of the work done by Hautala et al., Söderäng et al. set out to broaden the acquired real-time engine model to encompass the entirety of the hybrid powertrain situated in the Vaasa energy business innovation centre (VEBIC). This would include coupling the engine model with models of the generator, frequency controller, battery energy storage system, AC grid and transformer. The model would then be run on a real-time target machine to show the system's HiL capability.

The modelling process is discussed in detail by Söderäng et al. (2022), where the reader too is directed for a thorough view of the subject. For the purposes of this work, this section focuses on giving an overview of the subsystems within the hybrid powertrain. Figure 20 shows the Simulink layout and subsystem connections of the hybrid power plant model.

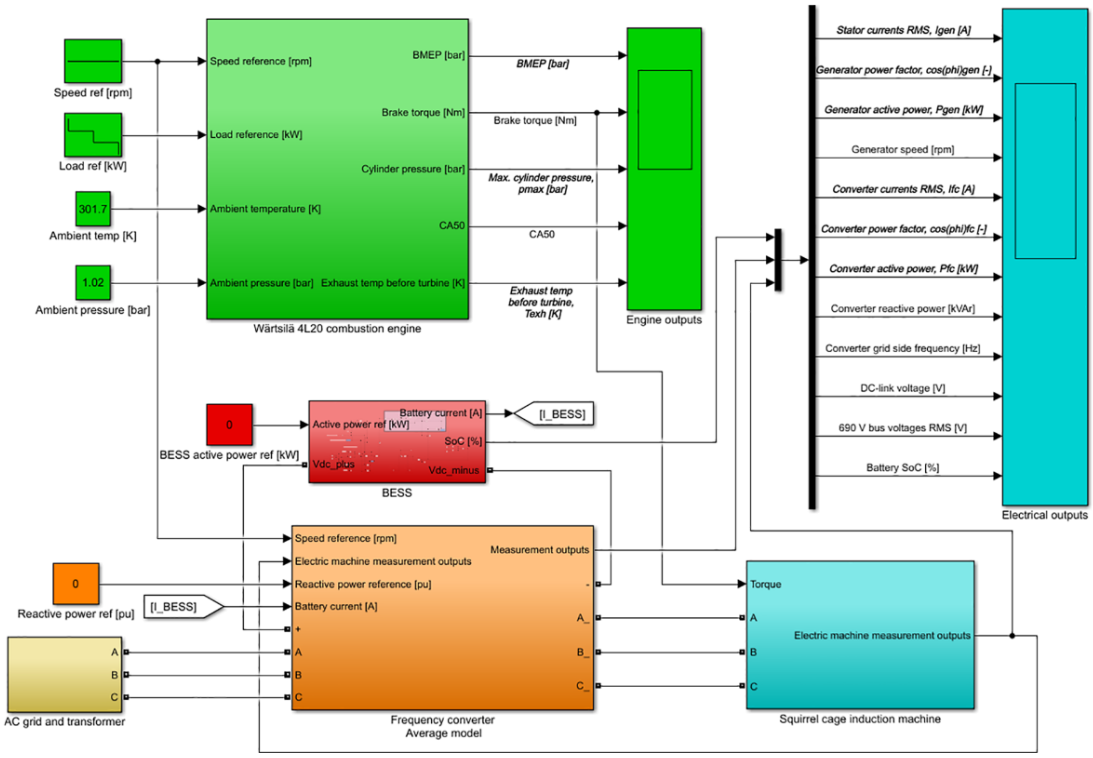

Figure 20. Simulink layout of the hybrid power plant model (Söderäng et al., 2022).

3.2.1 Squirrel cage induction generator

As shown in the layout, the engine is connected to a squirrel cage induction generator (SCIG), particularly an ABB M3LG 450LC 6G. The selection of an SCIG over a synchronous generator is dictated by the requirements of the engine test cell, where the generator must handle transient test cycles even when used as a motor. The SCIG's key parameters such as rated power and speed, along with electrical characteristics like voltage, frequency, and current, are outlined in Table 5.

Table 5. SCIG key parameters.

Parameter	Value
rated power/speed	1050 kW/1007 rpm
nominal apparent power	1200 kVA
rated voltage	690 V ± 5%
rated frequency	50 Hz
rated current	1042 A
rated power factor cos(ϕ)	0.87
rated torque	- 10228 Nm
rotor/stator resistance ratio, running	0.98
rotor/stator inductance ratio, running	0.71
rotor inertia	39.5 kgm2
number of pole pairs	3
efficiency @100% load	97.3%

Following the approach of Domínguez-García et al. (2012), the SCIG and the frequency converter (FC) were modelled with a few adjustments to better align with the requirements of the physical system. By parametrising the generic asynchronous machine block from the Simscape SPS library with generator inductances and resistances acquired from the manufacturer, a close match between the model and the actual system could be established.

3.2.2 Frequency converter

In parallel with the integration of the SCIG, a frequency converter (FC), particularly an ABB ACS800-17LC, is employed to effectively manage the power conversion processes between the generator and the power grid. This FC, tuned to the SCIG's operational specifications, also ensures that the generator can seamlessly transition between motor and

generator modes, crucial for handling the highly dynamic testing scenarios presented in transient cycle tests. The key parameters of the FC can be found in Table 6.

Table 6. Frequency converter key parameters.

Parameter	Value
Nominal power	2000 kW
Nominal voltage	690 V
Nominal maximum current for continuous loading	2035 A
Total losses @100% load	79.5 kW

The FC adjusts the AC output based on real-time data from the SCIG. Its ability to modulate voltage and frequency in real-time is necessary for maintaining system stability and efficiency, particularly under the fluctuating loads characteristic of engine test cells. The adaptive response is facilitated by an arrangement of insulated-gate bipolar transistor (IGBT) inverters and rectifiers, connected in a back-to-back configuration with a common DC link.

Due to the strict real-time requirements, the IGBT bridge model provided in the Simscape library had to be waived in favour of an average converter model, which in turn has the trade-off of harmonics not being represented. This was a necessary change, as achieving a 2 µs sample time required by the IGBT bridge was not realistic in this scope.

3.2.3 Battery energy storage system

In conjunction with the SCIG and FC, a battery energy storage system (BESS) is included to stabilise and optimise power delivery. As a dynamic energy storage, it plays a significant role in managing load variations and ensuring continuous power delivery. Unlike other components of the electrical system, the BESS is purely virtual, although one's acquisition to complete the physical system is currently in process. The BESS model's key parameters can be found in Table 7.

Table 7. Virtual BESS key parameters.

Parameter	Value
Nominal capacity	300 kWh
Nominal voltage	480 V
Maximum discharge rate	500 kW
Maximum charge rate	500 kW
Energy efficiency	95%
Cycle life	10,000 cycles

A lithium-ion battery solution was chosen for its cost effectivity and availability for large-scale battery storages, along with the necessary high energy density and fast response times. The battery model is connected to a battery management system (BMS) model, which regulates charging and discharging to prolong battery life and maintain safety.

3.3 Target machine and real-time performance

Söderäng et al. were running the hybrid power plant model on a legacy personal computer repurposed as a HiL target machine, specifically a Dell OptiPlex 760 from 2008 with a four-core Intel Core Q9400 CPU at 2,66 GHz and 8GB of RAM. By using this target machine to provide the power plant model with a real-time feed of the physical counterpart's measurements and vice versa, the process described in Figure 21 would be completed.

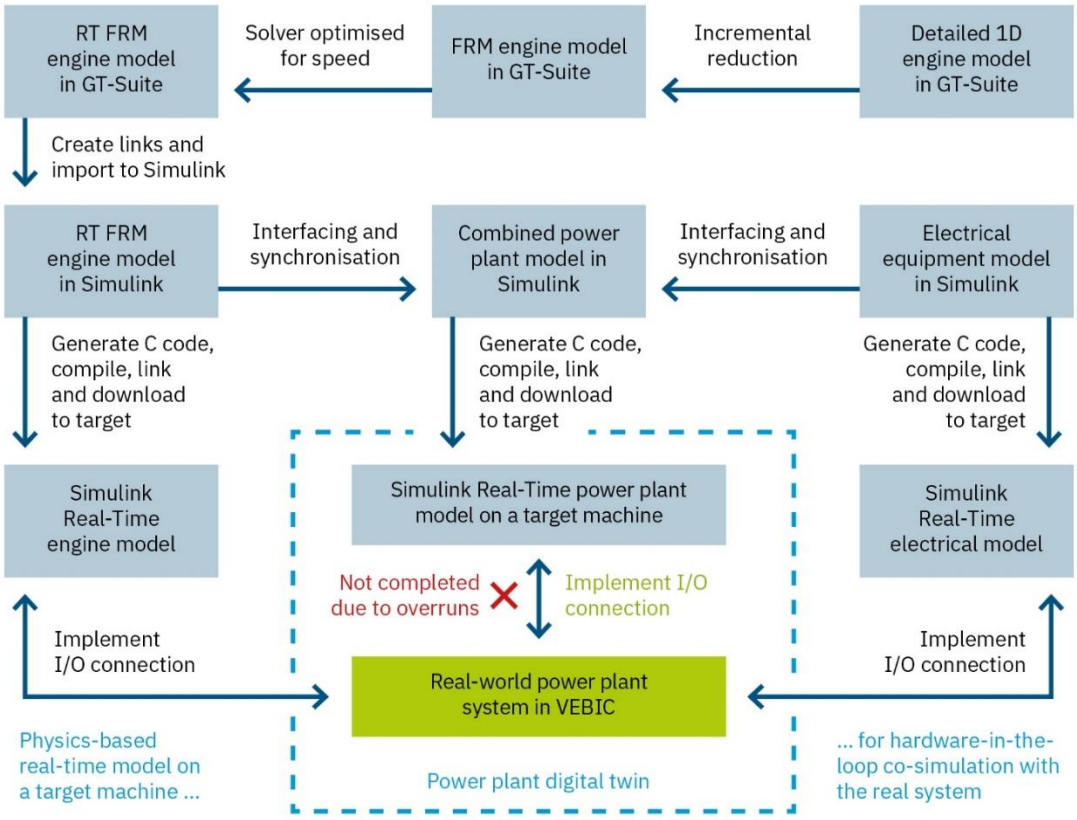

Figure 21. The process of creating a physics-based digital twin of a hybrid power plant (Söderäng et al., 2022).

The limited computational power available mandated the real-time simulation of the engine to be run at a coarse time-step of 1 s. This would be in range for typical airpath response times. However, at an engine speed of 1000 RPM, this would mean over eight full engine cycles per time step, which is inevitably too long to capture detailed in-cylinder dynamics.

3.4 Speedgoat target machine

With one ultimate goal of endeavours surrounding the VEBIC hybrid powertrain being a complete HiL system, a key part of this work is to deploy the hybrid powertrain model onto the real-time platform. For this purpose, a Speedgoat Mobile real-time target machine (Figure 22) has been acquired. The platform was selected due to the deep

integration with Matlab and Simulink, as opposed to utilising a repurposed personal computer as was done by Söderäng et al. (2022). In addition, active support from Gamma Technologies is available for embedding models created in GT-power into Simulink and further deploying them into a real-time target machine. The machine can operate in unison with any Matlab distribution from R2021a onwards.

The target machine in question is a model specifically meant for portable solutions. Thus, rather than emphasising computational power, it is used as a convenient test platform for models to be later deployed to and run on more powerful target machines. The exact processor model is not disclosed but it has been declared to be a two-core Intel Core i7 running at a clock rate of 2,5 GHz. Assuming that the declared clock speed is a base clock speed of a commercially available central processing unit, a search with these terms results in four possible options displayed in Table 8.

Table 8. Possible CPU models incorporated in target machine (*CPU-World: CPU Chart of Modern Intel and AMD Microprocessors*, n.d.).

Manufacturer	Intel	Intel	Intel	Intel
Family	Core i7 Mobile	Core i7 Mobile	Core i7 Mobile	Core i7 Mobile
Model number	i7-3555LE	i7-6500U	i7-6498DU	i7-7660U
Socket	BGA1023	BGA1356	BGA1356	BGA1356
Frequency (MHz)	2500	2500	2500	2500
L2 cache (KB)	512	512	512	512
L3 cache (KB)	4096	4096	4096	4096
Cores	2	2	2	2
Threads	4	4	4	4

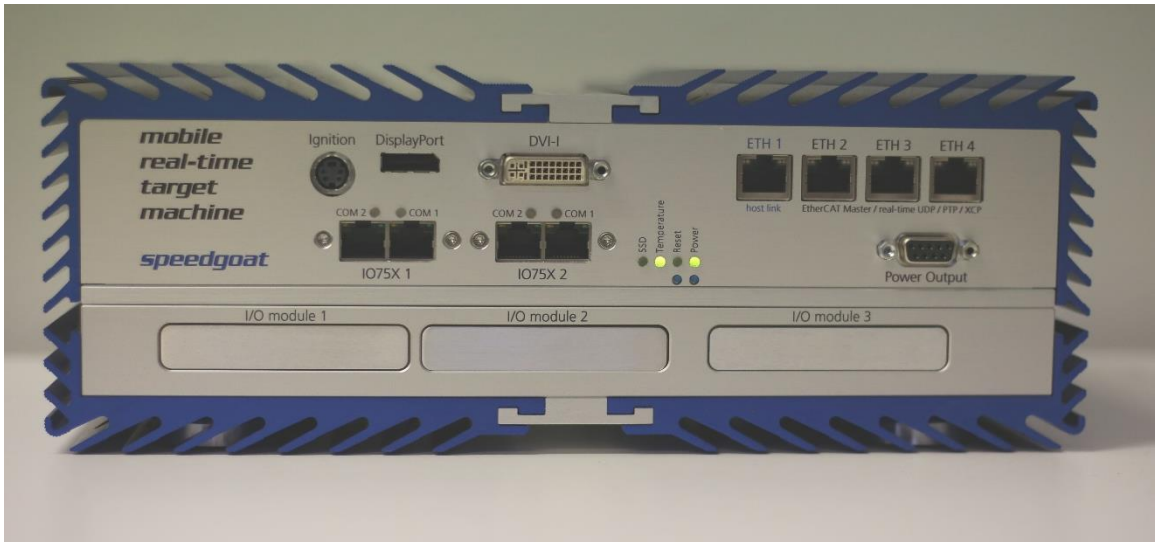

Figure 22. Speedgoat target machine.

As outlined in section 2.2.2, the target machine uses a QNX-based RTOS. The deterministic nature means that the system's response to inputs is predictable and happens within a guaranteed time frame. This is imperative in applications such as XiL simulations, where the target machine needs to reliably run the preinstalled program while potentially communicating with real-world sensors.

Apart from a robust design and purpose-built software, the particular Speedgoat Mobile is equipped with Modbus capabilities. This allows it to communicate directly with the laboratory equipment of Vaasa Energy Lab. An option to embed an FPGA module for high-frequency applications exists, but this particular unit has none. This is due to the CPU comfortably handling sample rates up to the range of 1-20kHz (*Speedgoat and Simulink Real-Time Workflow | Speedgoat*, n.d.), which covers the intended engine control applications.

3.5 Implementing task execution time monitoring

Since version 2020b, Simulink Real-Time has included an *SLRT Overload Options* block specifically for real-time task execution time and overrun monitoring. In addition, the block allows the user to set an upper limit for overloads allowed before terminating the

simulation. The user can also set an amount of time steps from the beginning of simulation during which any number of overloads will be allowed.

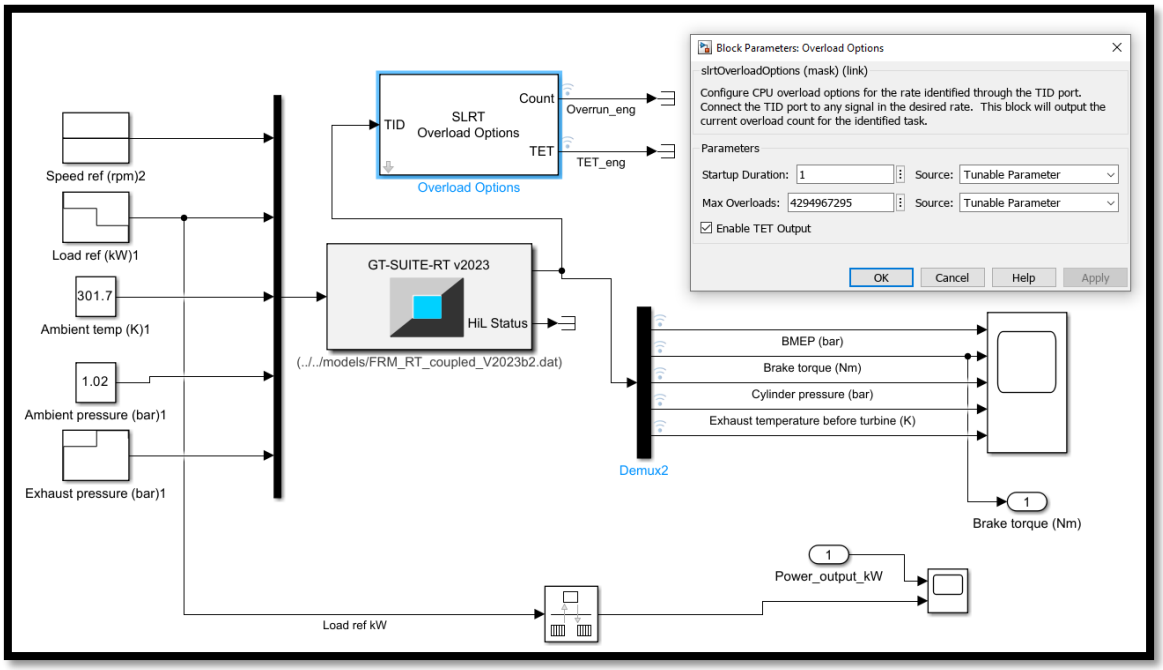

Figure 23. Engine model with highlighted overload options block added.

In Figure 23, the overload managing block is connected to the engine model, which in turn has been exported from GT-Power into a .dat file and then brought to the Simulink environment as an S-function. The sample rate being monitored by the overload managing block is inherited from the signal, which in this case is determined by the sample time set in the S-function's parameters, as described in Figure 24.

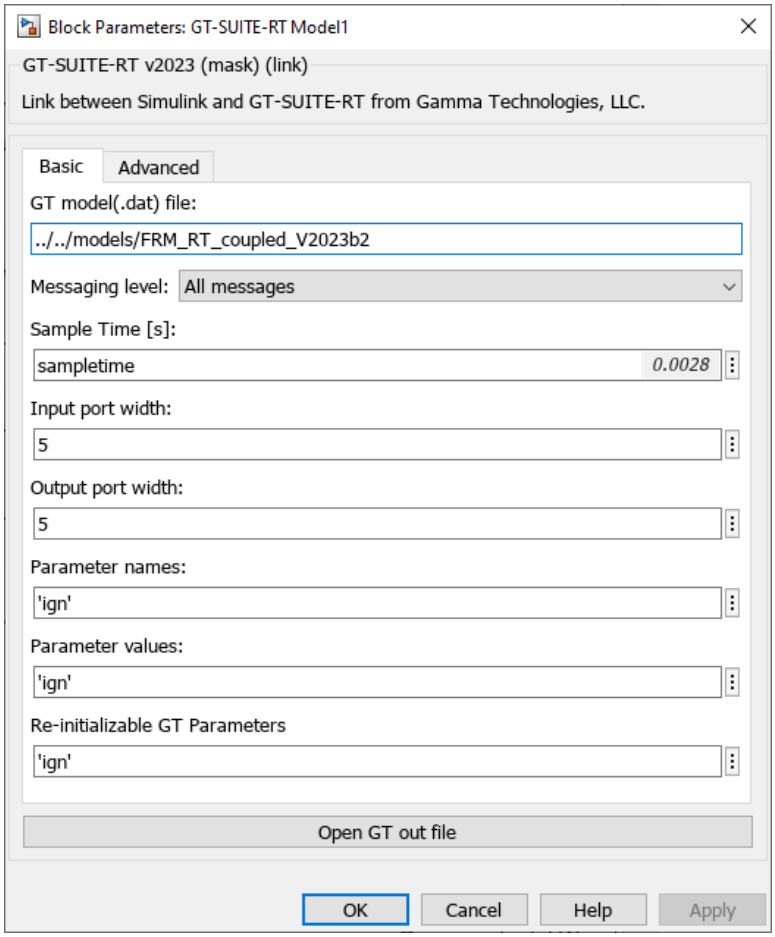

Figure 24. Engine S-function parameters.

Similar overload blocks have been added to two other locations: one connected to the electric drive and BESS subsystem running at a sample rate of 50 µs and another connected between the speed controller and field-oriented controller submodels, where the torque signal is transferred at a sample rate of 200 µs. In the simulation phase, the overload limit of each is set to the maximum value.

3.6 Validation methodology

As a result of the multi-stage model reduction process described in section 3.1, the modified model has undergone several rounds of validations. This process was first done by Hautala et al. (2022), who reduced the original detailed 1D model to an FRM. The detailed 1D model was first validated against experimental data and selected parameters

were raised into discussion. These parameters were intake manifold pressure (p_3), exhaust temperature at turbine inlet (T_5), maximum cylinder pressure (cycle and cylinder averaged) and brake thermal efficiency (BTE), as presented in Figure 25. The first two were selected as they accumulate errors of the airpath model and significantly shape model performance through feedback/forward mechanisms. The p_3 value affects the pressure of all cylinders and is in turn affected by T_5, which shapes turbine power. The load points begin at 100 % and are reduced in 25 % increments.

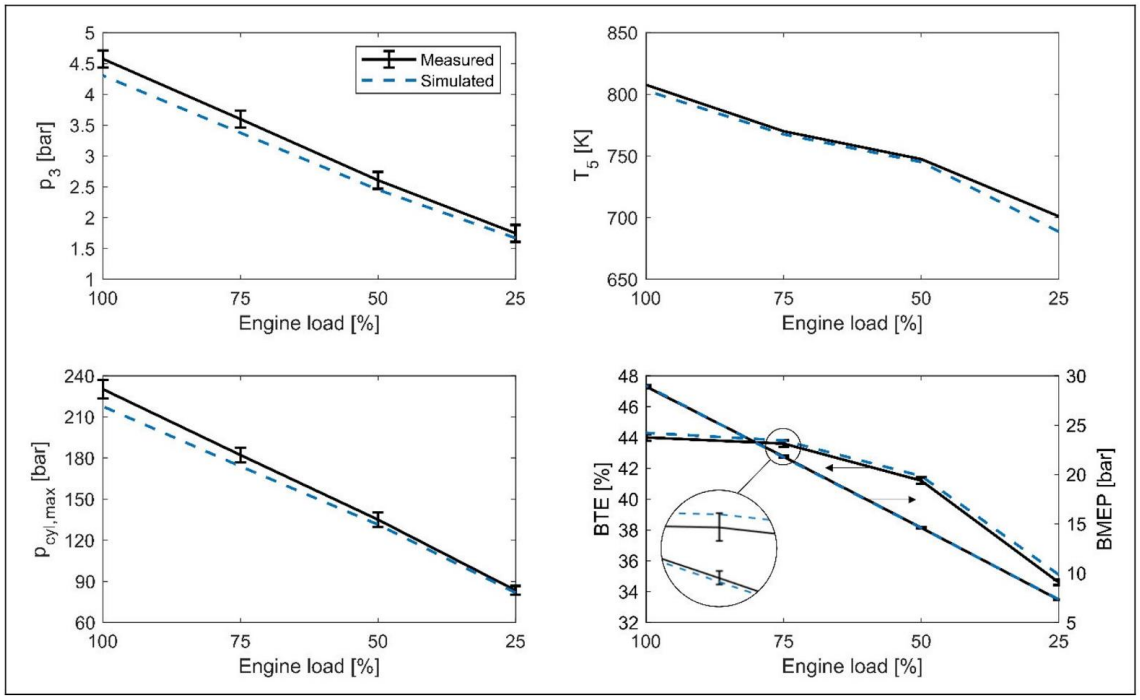

Figure 25. Measured and simulated 1D engine model validation parameters of all four load points together with estimated error bars (Hautala et al., 2022).

The allowed tolerances and their definition are extensively discussed by Hautala et al. The only value consistently exceeding these tolerances was p_3, which was due to lack of re-calibrating the complete engine model with p_3 and rather balancing the model using the p_2 value from component-level validation. For thorough discussion on the matter and its effect on simulation accuracy, the reader is directed to *Toward a digital twin of a mid-speed marine engine: From detailed 1D engine model to real-time implementation on a target platform* by Hautala et al. (2022).

Upon reducing the model into an FRM, another round of validation took place, as the different stages of FRM were compared to the detailed model. Apart from the values presented in Figure 25, volumetric efficiency was added as an important calibration factor. These values along with the RTF were plotted for each FRM reduction phase (Figure 19) separately and the resulting graph is presented in Figure 26. The allowed tolerances are listed in Table 9.

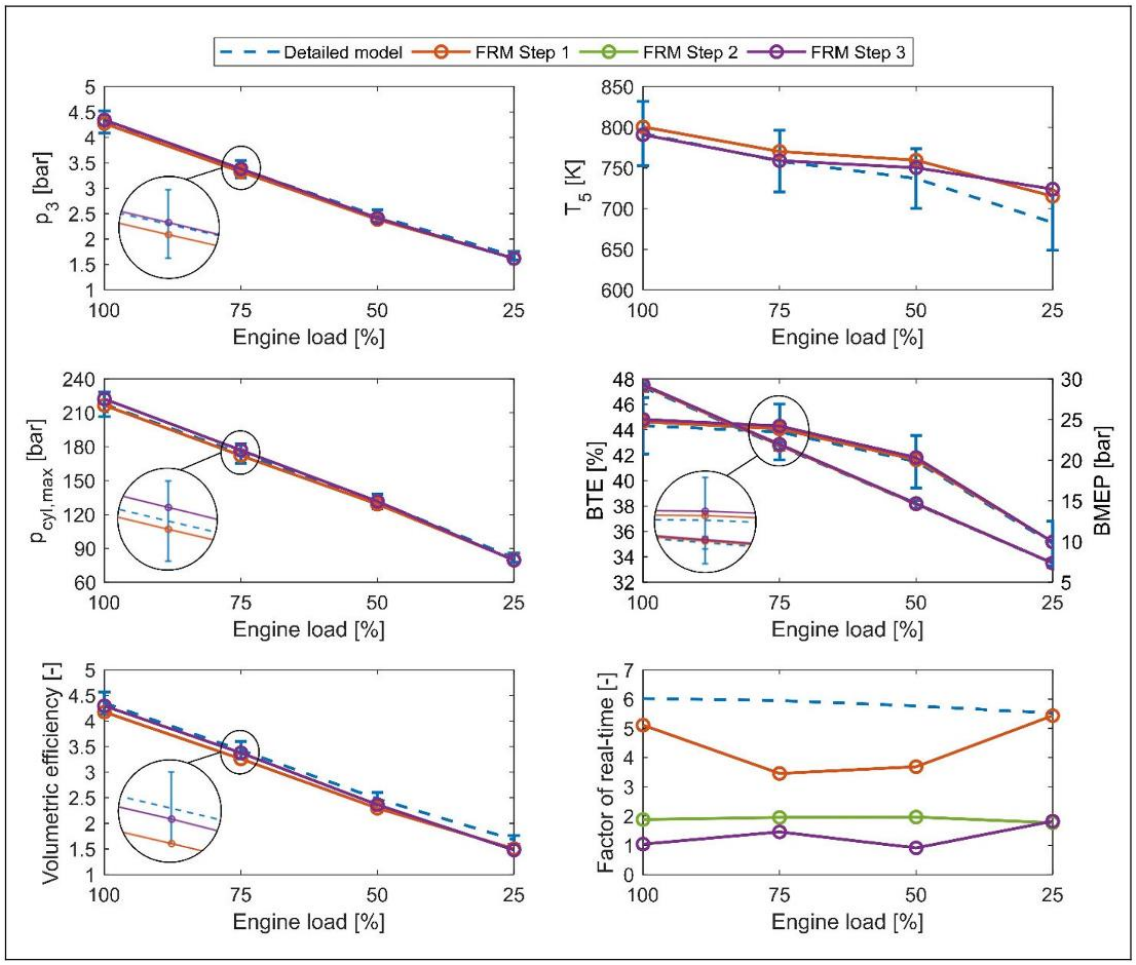

Figure 26. Simulated engine performance parameters and real-time factors of the key steps of the FRM conversion (Hautala et al., 2022).

Table 9. FRM key variables and tolerances (Hautala et al., 2022).

Variable	Units	Tolerance type	Tolerance
BMEP	bar	Absolute	0,5
BTE	%	Percentage	5,0
Volumetric efficiency	fraction	Percentage	5,0
Average max. cylinder pressure	bar	Percentage	5,0
Intake manifold pressure	bar	Percentage	5,0
Exhaust temperature at turbine inlet	K	Percentage	5,0

The simulations consistently fulfilled the tolerance limits, with an exception of the T_5 value. However, Hautala et al. deemed this uncertainty acceptable in the light of known challenges and inaccuracies in exhaust temperature measurement.

After the FRM was satisfactorily validated, Söderäng et al. (2022) moved the model from standard GT-Power license to GT-Suite-RT, which effectively changed the solver from explicit, forward Runge-Kutta method to explicit Euler. This was found to have almost no impact on model accuracy, as is shown in Figure 27.

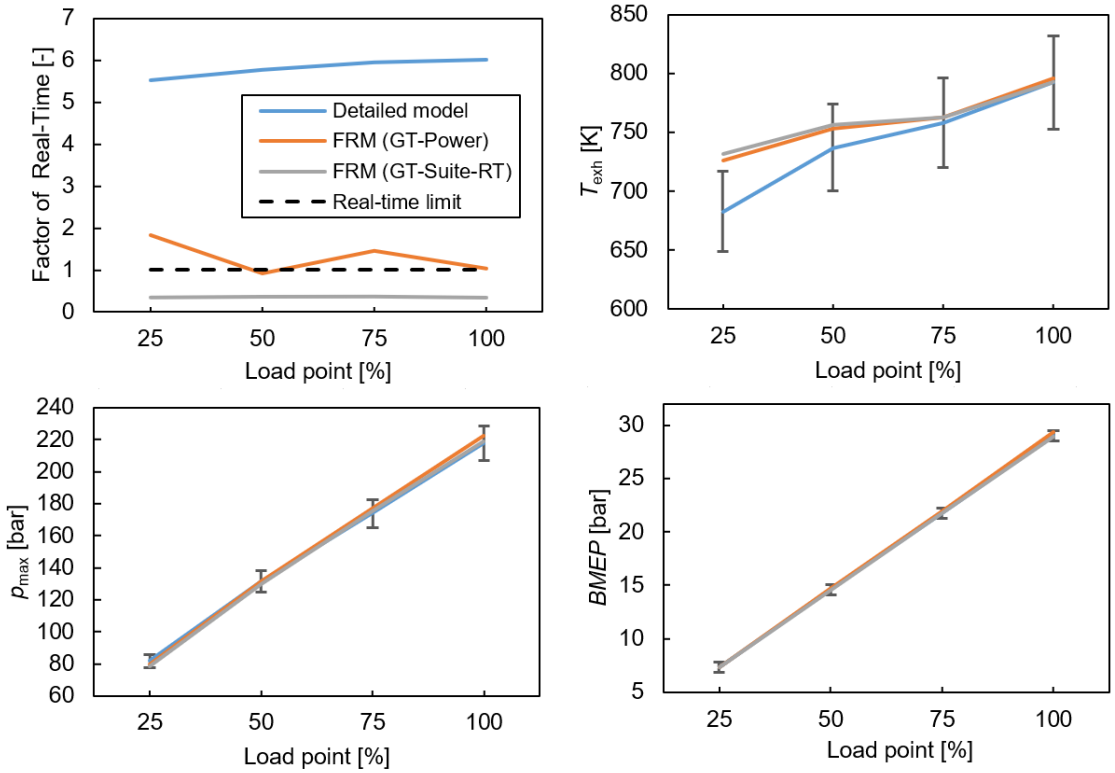

Figure 27. RT license effect on FRM accuracy (Söderäng et al., 2022).

As mentioned before, the Speedgoat target machine requires a Simulink version of R2021a or above, and GT-Power has begun providing official support for the Speedgoat target system since v2023. As Söderäng et al. were using software versions 2017a and 2019, respectively, the models must be migrated to up-to-date software.

GT-Power offers a tool for evolving models created in previous versions of the software upon opening one, as shown in Figure 28. As the fast-running model created by Hautala et al. has been thoroughly validated against experimental results with a good match, it is imperative not to cause significant deviations through this version upgrade. Note that assessing potential improvements accessible by using updated modelling options is beyond the scope of this thesis, and the focus will be solely on recreating results from an earlier software version.

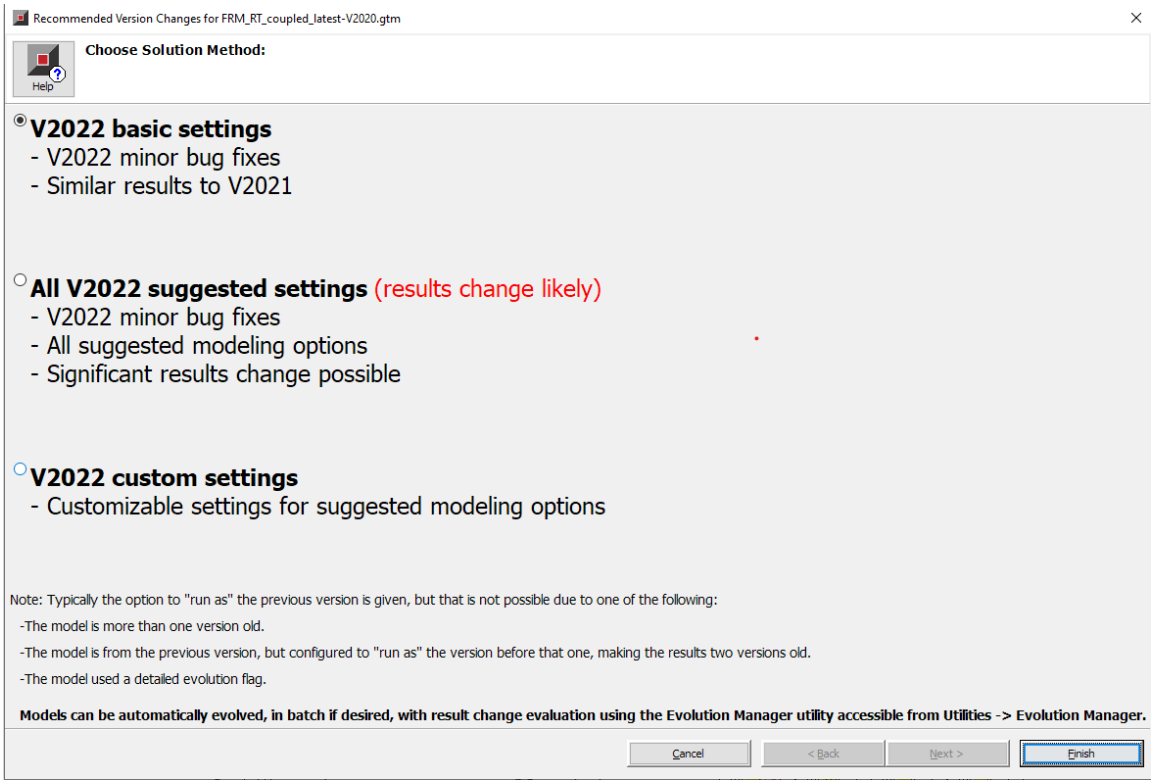

Figure 28. Evolution option used in upgrading the engine model.

Due to limitations of the GT-Power model evolution tool, the model must be evolved twice; first from v2019 to v2022 and then from v2022 to v2023. The simulations will be carried out using the same stair input that was used by Söderäng et al. and the results compared using the comparison tool of Simulink Simulation Data Inspector.

Additionally, steady-state results at the load points used by Hautala et al. and Söderäng et al. will be collected during simulations of the combined model and compared to the original validation data. The comparisons will be done according to Figure 27 for different sample times, with the RTF being presented and discussed separately. To ensure comparability of results, the values from the last second of each load point shall be averaged. Maximum average cylinder pressure forms an exception, as the values to be averaged shall be the highest pressures of each engine cycle that occurs within the last second of each load point.

3.7 Use-case definition

As an additional experiment, the combined power plant model has been included in a system-level simulation demonstration within the CASEMATE project. The experiment was carried out by presenting the simulation as functional mock-up units, which were exported from GT-Power for the engine model and from Simulink for the electrical model. The Functional Mock-up Interface (FMI) is an open standard for the exchange and co-simulation of dynamic models. It was developed to facilitate the integration of models from various tools into a single environment, thus enabling seamless exchange and interoperability of complex systems. FMI allows models to be exported from their original development environments as Functional Mock-up Units (FMUs), which can then be imported and utilised in different simulation platforms. The standard supports both model exchange and co-simulation, meaning that FMUs can either be used to provide model equations (for simulation by a host environment) or to execute a simulation in conjunction with other FMUs or simulation components (*Functional Mock-up Interface Specification*, n.d.).

A Functional Mock-up Unit (FMU) in turn is the executable model or component that adheres to the FMI standard. FMUs encapsulate the model equations, algorithms, and associated data, packaged in a standardised format that includes an XML file describing the model interface and binary files containing the compiled code. This encapsulation ensures that models can be shared and executed across different simulation tools without needing to be reconfigured or redeveloped, significantly enhancing the efficiency and flexibility of the model-based design process. FMUs can be utilised for both model exchange, where they provide the necessary information for a host tool to simulate the model, and co-simulation, where the FMU itself performs simulations and exchanges data with other FMUs or simulation components during runtime (*Functional Mock-up Interface Specification*, n.d.).

Following the principles of model-based system engineering, the system boundaries were drawn according to Figure 29.

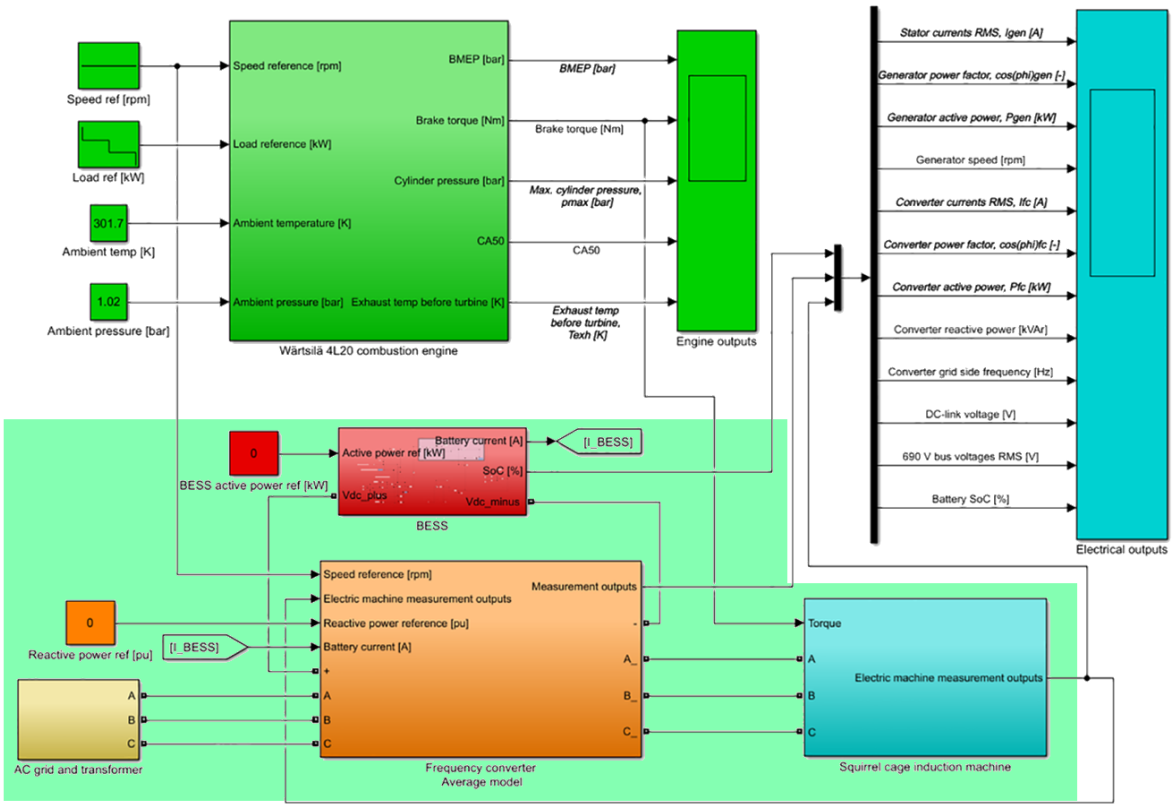

Figure 29. System boundary (highlighted in green).

As system-level simulations usually employ 0D or data-driven simulations, the 1D flow solution presented here is a novel solution by itself. Due to its predictive and physical nature, other values can be exported from the same engine model on-demand by merely changing the co-simulation setup and recompiling the model into an FMU. As the original model has been well validated and documented, the FMU derivative is expected to far surpass the functionalities of alternative modelling options.

3.8 Scope of simulations

3.8.1 Engine model validation

As the standalone engine model has been upgraded into newer software versions, it is necessary to confirm that model accuracy has not degraded as a result. Towards this end, the model will be run and the results compared to those of the original model. The measured parameters and their respective tolerance goals are listed in Table 10.

Table 10. Standalone engine model validation tolerances.

Variable	Relative tolerance (%)
BMEP (bar)	0,1
Brake torque (Nm)	0,1
Cylinder pressure (bar)	0,1
Exhaust temperature before turbine (K)	0,1
CA50	0,1

During combined model simulations performed on the target machine, the engine model outputs will be recorded at different sample times for validation purposes. The sample times are listed in Table 11.

Table 11. Sample times for engine model validation.

T_s (CA)	T_s (ms)
6000	1000,00
360	60,00
180	30,00
90	15,00
45	7,5
22.5	3,75
18	3,00
17	2,83

3.8.2 Determining real-time capability of the standalone engine model and the combined hybrid power plant model

As the target is to include the control-oriented physical engine model in a digital twin, the simulations will be aiming to find out the limits of the newly acquired hardware in this purpose. By altering the *sampletime* value of the engine model S-function (Figure 24), the sample time will be gradually decreased, starting with that of a full engine cycle, until overruns occur. The increments will be based on the time that it takes for the crankshaft to turn one degree when the engine is running at 1000 RPM, close to its rated speed. These increments can be found in Table 12. Separate simulation series will be carried on the engine model with and without being connected to the electrical model.

Table 12. Sample times used in simulations.

Engine model		**Combined model**	
T_s (CA)	**T_s (ms)**	**T_s (CA)**	**T_s (ms)**
6000	1000,00	6000	1000,00
360	60,00	360	60,00
180	30,00	180	30,00
90	15,00	90	15,00
45	7,50	22.5	3,75
22,5	3,75	18	3,00
18	3,00	17	2,83
17	2,83		
15	2,50		
14	2,33		
13	2,17		
12	2,00		
11	1,83		
10	1,67		
9	1,50		
8	1,33		
7	1,17		
6	1,00		
5	0,83		
4	0,67		

The simulation scheme for individual simulations will be similar to the work by Söderäng et al. (2022). Thus, the load signal for the engine will begin at 100 % and step down by 25 % in 5 s intervals. This leads to a simulation length of 20 s for each individual simulation. As the models have been validated in previous works both separately and combined, the focus will be solely on the computational performance.

3.8.3 System-level simulations

As a part of the CASEMATE project, the hybrid power plant model is to be coupled into a system-level simulation tool. Towards this end, hybrid power plant model will be exported into a Functional Mockup Unit (FMU). An FMU will be created of both the standalone engine model and the electrical system.

After FMU compilation, the models will be tested on Simulink to ensure that no significant accuracy loss would occur from running the models as FMUs. As only one output for both the engine model and the electrical model will be used for this initial demonstration, only engine brake torque and total electrical power output will be measured and compared to the respective measurements from the original models. Figure 30 shows the resulting, simplified model structure. Note, that the electrical power signal going from the electrical model into the engine model was not used and rather is a remnant of the original measurement setup, where all measurement blocks were situated within the engine subsystem.

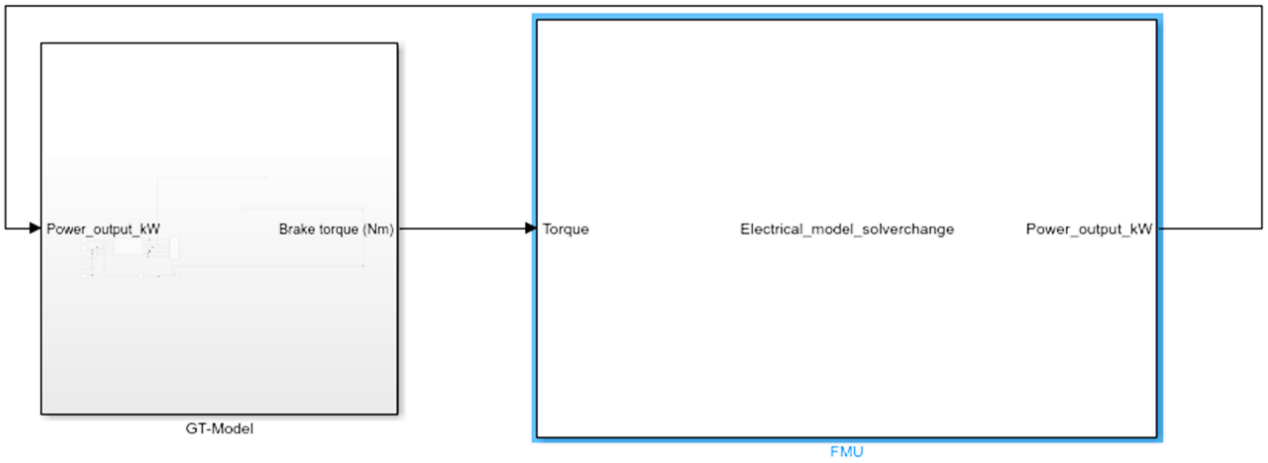

Figure 30. FMU test layout in Simulink.

To ensure that the accuracy has indeed been retained, the model will be tested against a 2 % allowed relative and 2 units absolute tolerance in both engine torque and electrical power output. These validations will be done by running the FMUs in Simulink and comparing their results to the original ones in the Simulation Data Inspector signal comparison tool.

Additionally, the RTF of the hybrid power plant model will be determined by recreating the 20 s simulation period from Table 13 and measuring the elapsed wall-clock time.

Table 13. Decreasing power request in system-level simulation.

Simulation time (s)	New power request (% of nominal)
0	100
5	75
10	50
15	25
20	0

4 Results

The results presented in this section offer an analysis of the engine and combined model simulations, emphasising their real-time capabilities. Key performance metrics such as task execution times and system stability are systematically recorded and analysed to assess model accuracy and speed. The simulations were conducted across various sample times, starting from 1 second and progressively decreasing to the lower double-digit milliseconds range. This examination both highlights the computational performance achieved with the Speedgoat real-time target machine and identifies potential areas for further optimisation. The findings aim to validate the hybrid power plant model's readiness for HiL applications and its eventual evolution into a digital twin.

4.1 Model accuracy

4.1.1 Validation of upgraded engine model

By parametrising a signal comparison session with the parameters presented in Table 10, all five signals could be simultaneously checked for deviances. The resulting view displayed in Figure 31 shows that with an allowed relative tolerance of 0,1 %, there are mismatches within the BMEP and brake torque values.

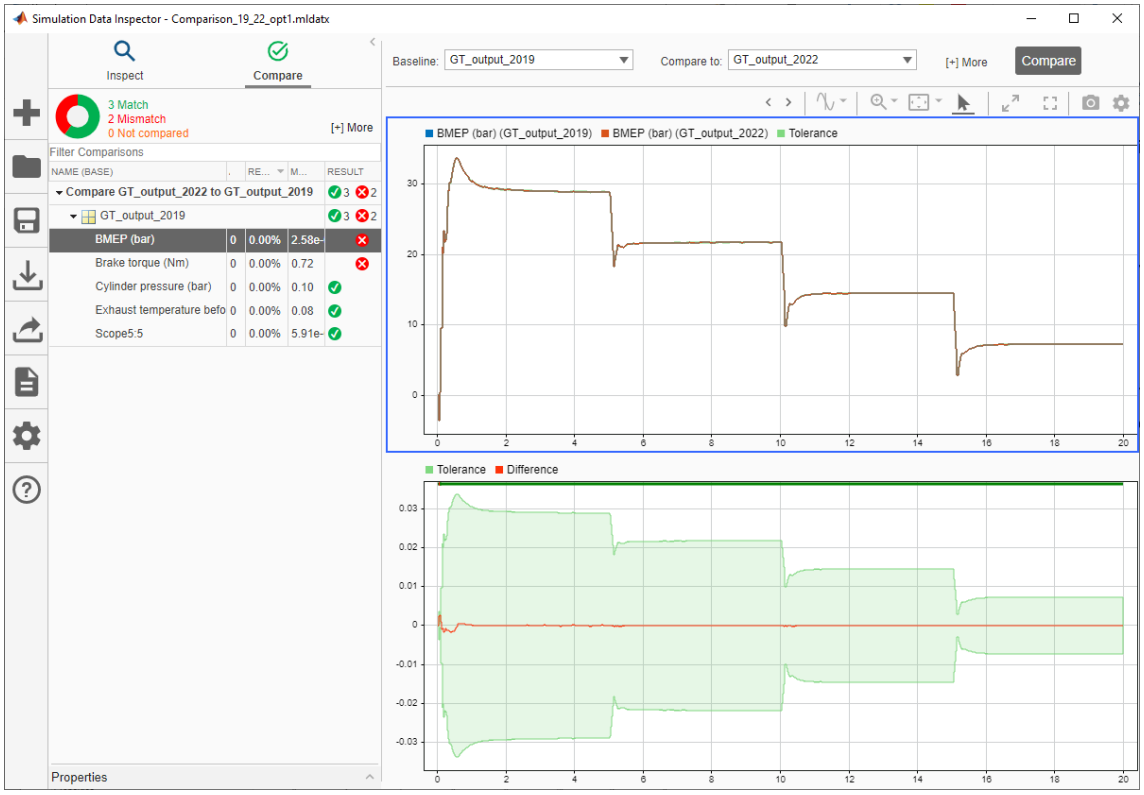

Figure 31. Results of comparison between simulations done on GT-Power versions 2019 and 2022.

A closer look into the plot is taken in Figure 32, where the focus is on differences over time instead of absolute results over time.

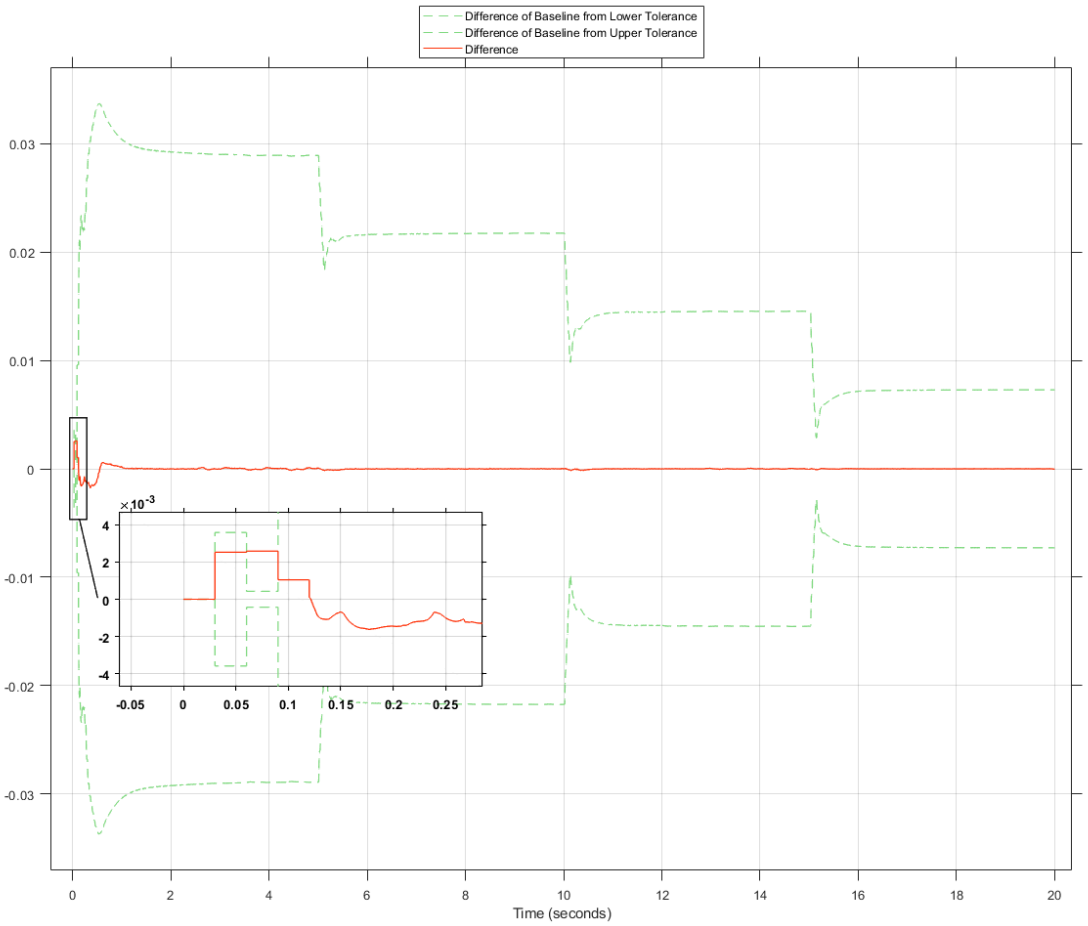

Figure 32. Differences on BMEP simulation results between GT-Power v2019 and v2022.

Upon closer inspection it becomes clear that the dissonance is limited to happening within one time step in the initial stages of the simulation. As the BMEP value at this time step is low, the absolute error of 2,58 mbar is enough to exceed the relative tolerance of 0,1 %.

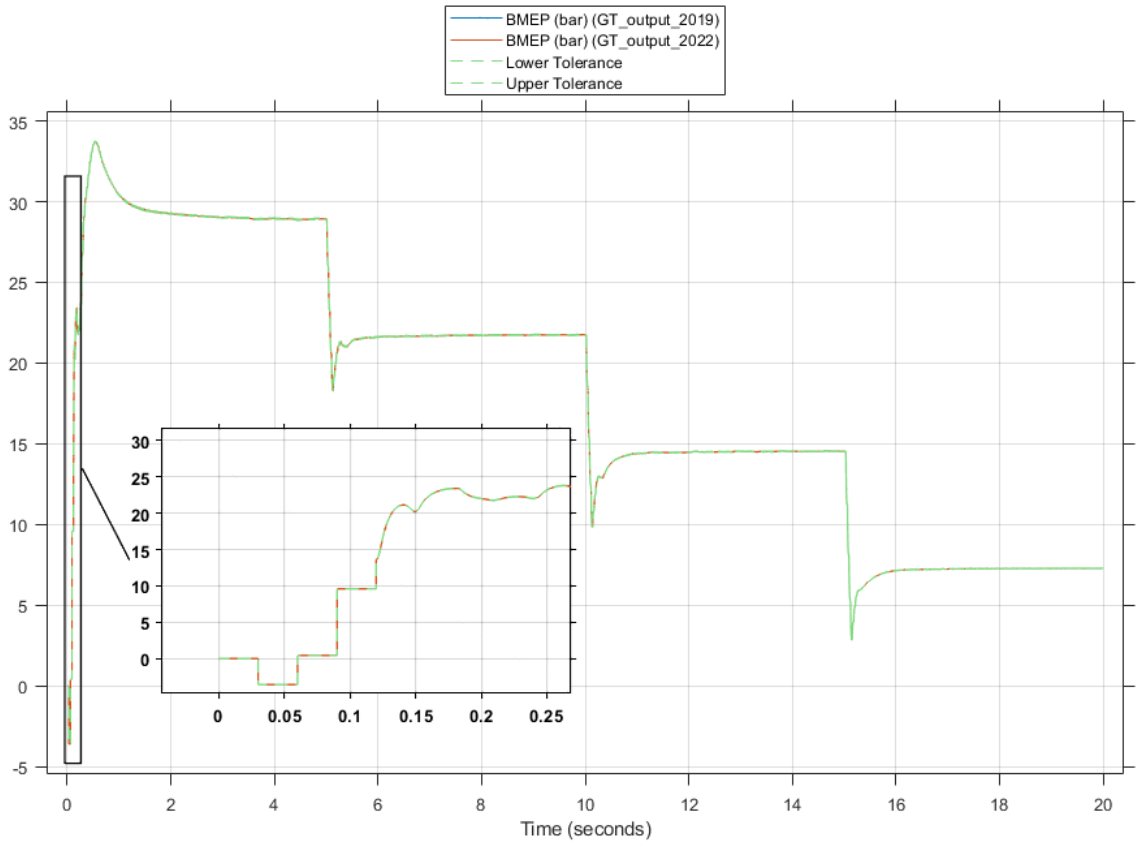

Figure 33. Absolute results of engine simulations on GT-Power v2019 and v2022.

A magnification of the absolute results of the simulations in Figure 33 shows that differences in the results are visually imperceptible on the BMEP curve. This is due to the magnitude of the error being 10^{-3} whereas the BMEP values in the affected time steps are in the range of single digits.

The transition from GT-Power v2022 to v2023 was done on a similar manner and with the same options as described in Figure 28.

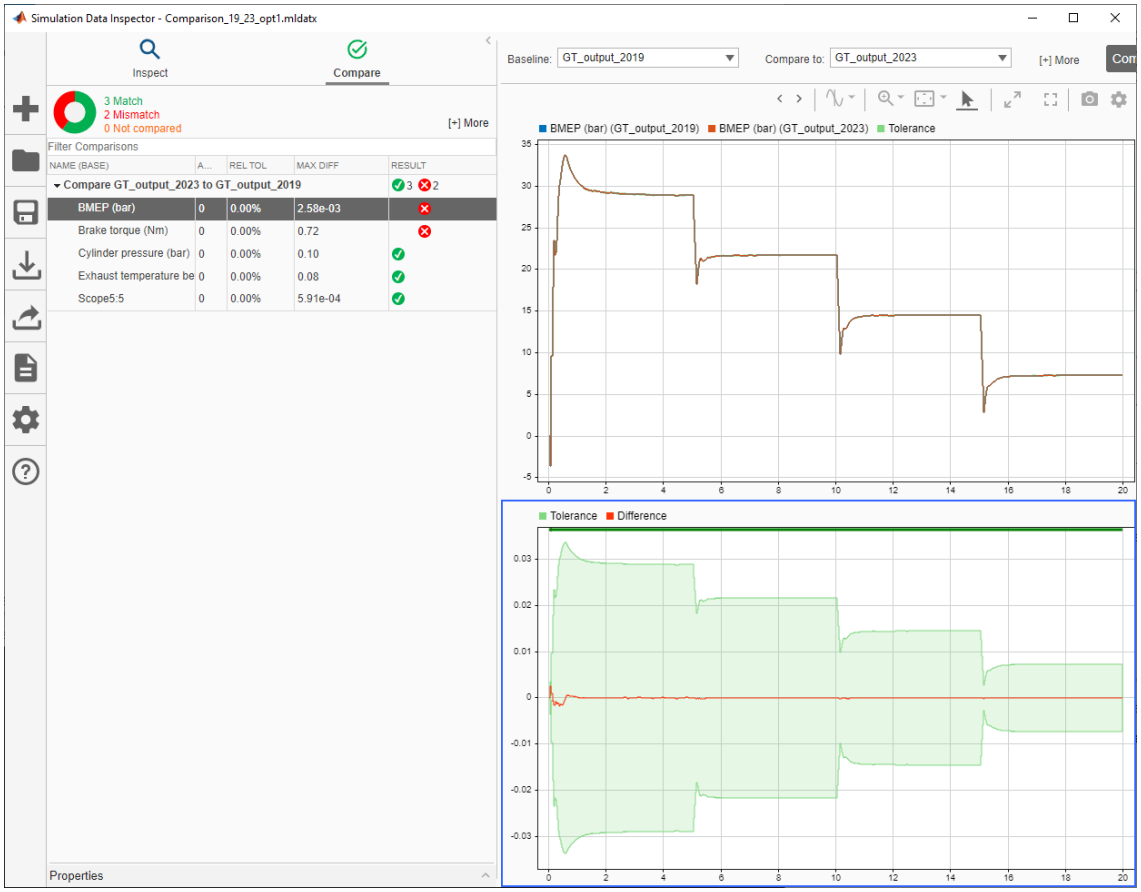

Figure 34. Results of comparison between simulations done on GT-Power versions 2019 and 2023.

Figure 35 shows that comparing the simulations done on v2019 to those done on v2023 yield the exact same results as the comparison to v2022.

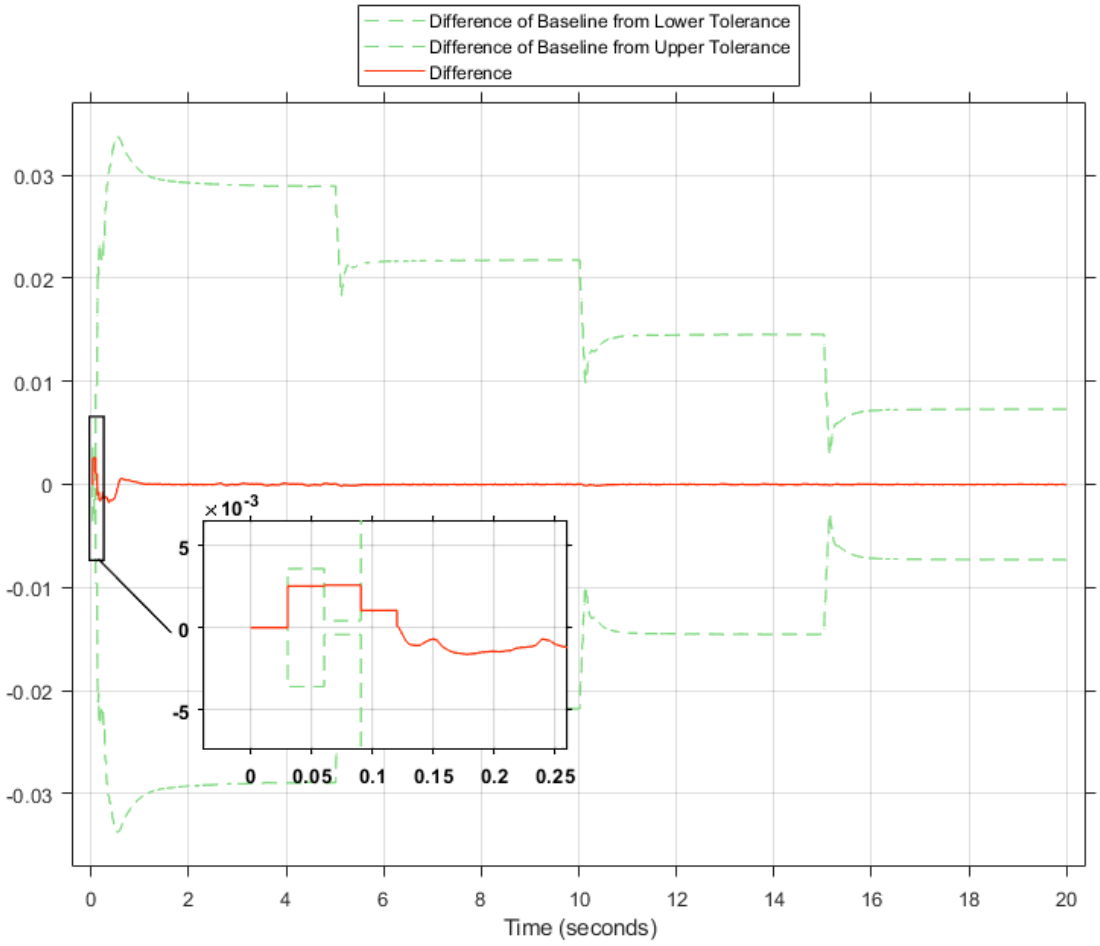

Figure 35. Differences on BMEP simulation results between GT-Power v2019 and v2023.

As is seen in Figure 35, the cause and time of the dissonance are identical as well. Thus, it can be stated that allowing the tolerances presented in Table 14, the results match.

Table 14. Tolerances for evolving engine model from GT-Power v2019 to v2023.

Variable	Relative tolerance (%)	Absolute tolerance
BMEP (bar)	0,1	$2{,}58*10^{-3}$
Brake torque (Nm)	0,1	0,72
Cylinder pressure (bar)	0,1	0
Exhaust temperature before turbine (K)	0,1	0
CA50	0,1	0

Comparing to Table 15, where Hautala et al. present the tolerances allowed in the reduction of the detailed model into an FRM, these newly presented tolerances comfortably fall within the original margins and thus do not jeopardise model accuracy.

Table 15. FRM key variables and tolerances (Hautala et al., 2022).

Variable	Units	Tolerance type	Tolerance
BMEP	bar	Absolute	0,5
BTE	%	Percentage	5,0
Volumetric efficiency	fraction	Percentage	5,0
Average max. cylinder pressure	bar	Percentage	5,0
Intake manifold pressure	bar	Percentage	5,0
Exhaust temperature at turbine inlet	K	Percentage	5,0

4.1.2 Validation of real-time simulations

The validation process continued by comparing the averaged results of simulations run at different sample times for maximum cylinder pressure, exhaust gas temperature and BMEP. In Figure 36, the averaged cylinder pressures of the detailed model and FRMs are presented along with results from simulations on different sample times. The error bars for the following validation graphs are based on the detailed model and Table 15, where applicable.

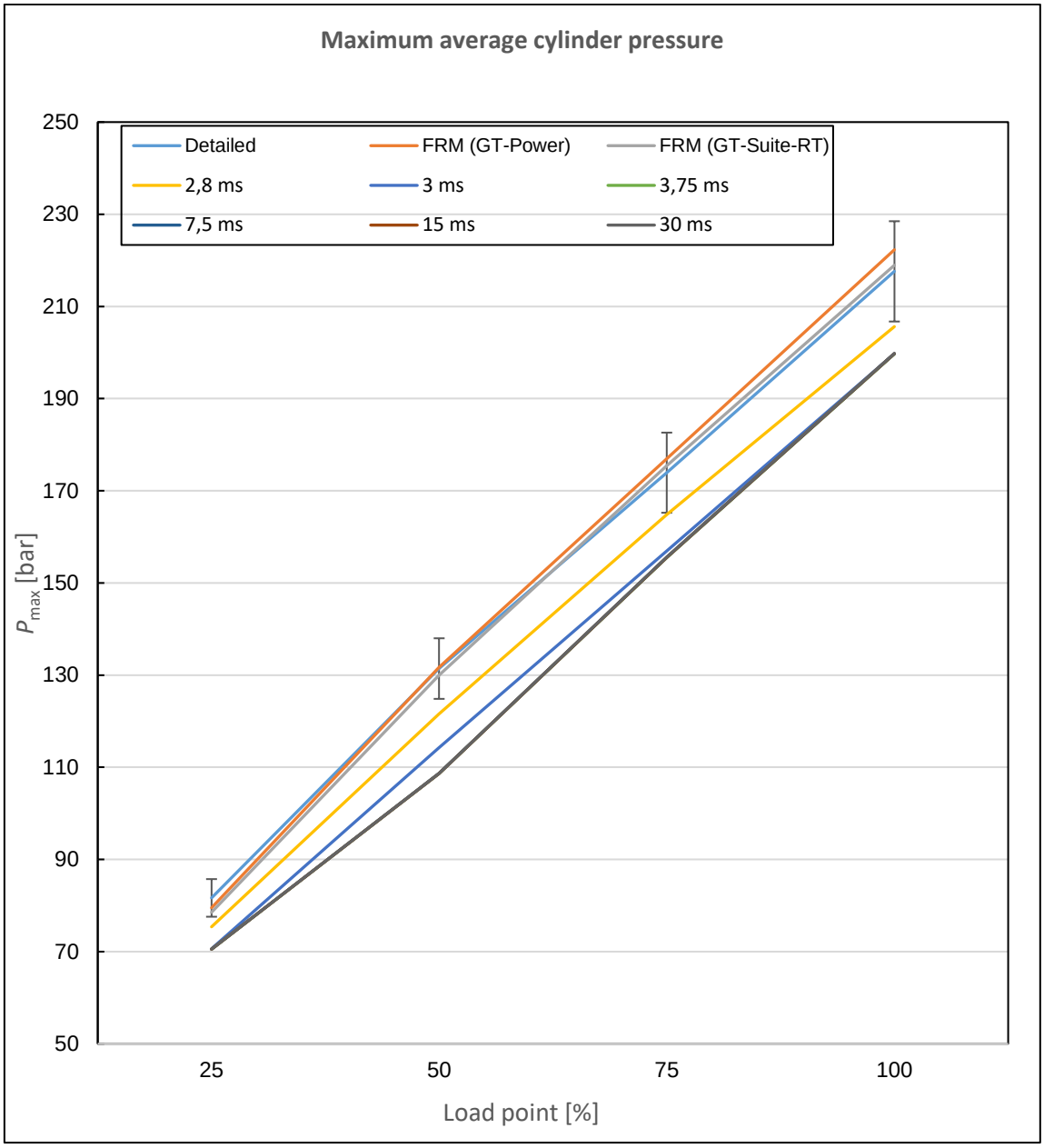

Figure 36. Comparison on maximum average cylinder pressure.

The two coarsest sample times had to be waived from discussion on average maximum cylinder pressure (P_{max}), as they were unable to reproduce in-cylinder pressure to a level of providing sensical results. The comparison between the remaining simulations shows that the 2,8 ms sample time barely falls short of the 5 % tolerance. Apart from it, the 3 ms sample time follows behind and the rest are visually imperceptible from each other.

For a closer view, Figure 37 presents the absolute deviation of the P_{max} value for each simulation type.

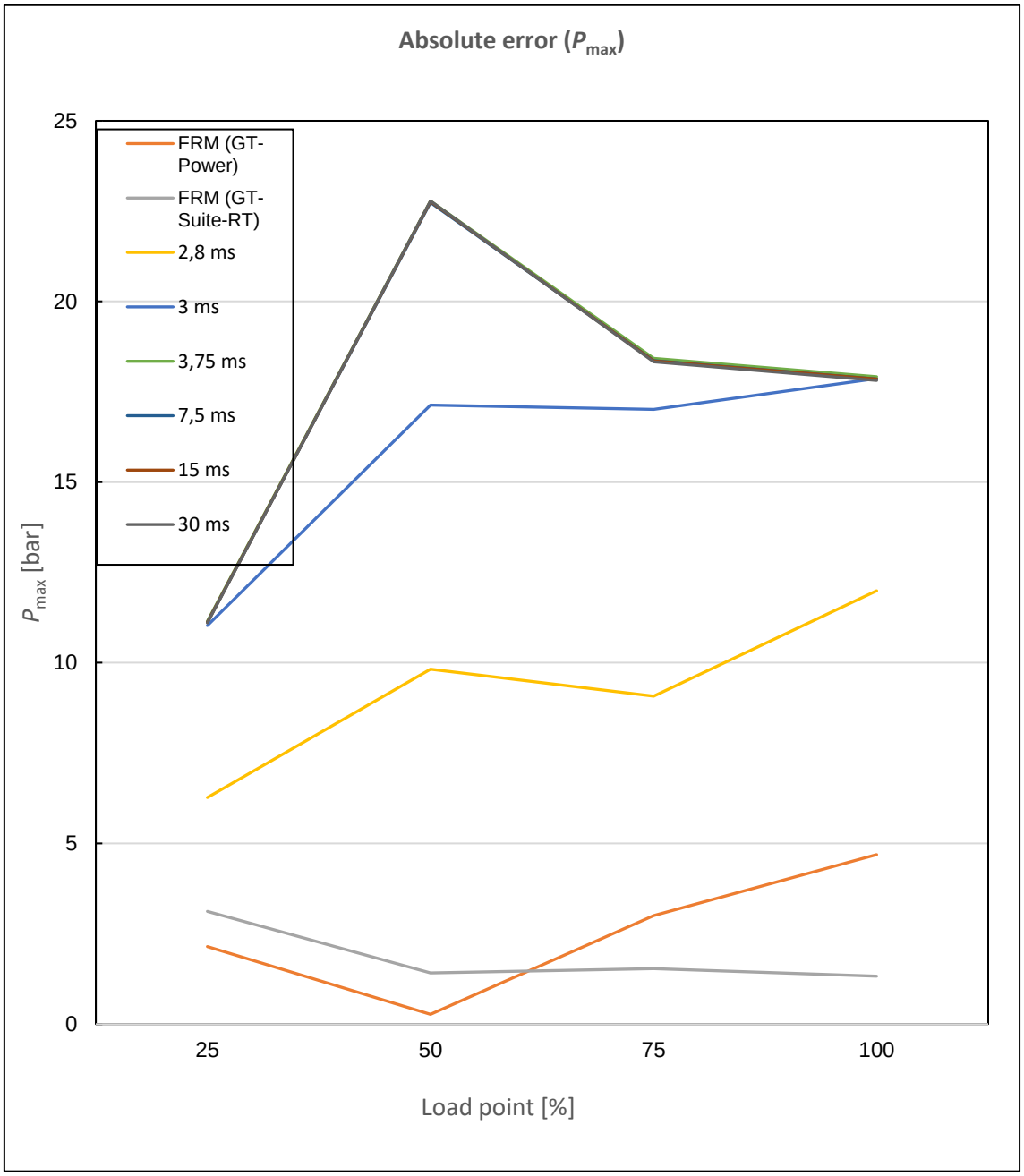

Figure 37. Absolute deviations of the P_{max} value.

The difference between the coarser sample times stays imperceptible, but the improvement between the finer ones becomes clearer. There is no absolutely clear trend in

relation to the load points, which is due to a notable shortcoming of this validation approach.

As the sample times are predetermined and are never synchronised to the engine cycle after starting the simulation, the sampling practically happens at an arbitrary point on the cylinder pressure curve. Averaging values from a longer simulation time period can mitigate this issue to a certain extent, but at certain cases any length of simulation period will not fix the issue. In cases where the sample time is predetermined, the engine is running at a stable speed and the engine cycle time happens to be an integer multiple of the sample time, the sampling will always happen at the same time in relation to the combustion event. Thus, the measurement will always hit the same relative spot on the pressure curve. In this work, this was observed at the sample time of 30 ms, which is exactly a quarter of engine cycle time at 1000 RPM. This led to the cylinder maximum pressures during each load point being identical to the first decimal.

In the other hand, this caveat underlines the reason for endeavours such as this one. In order to consistently reproduce cylinder pressure, the sample times cannot be extended at will in pursuit of real-time capability.

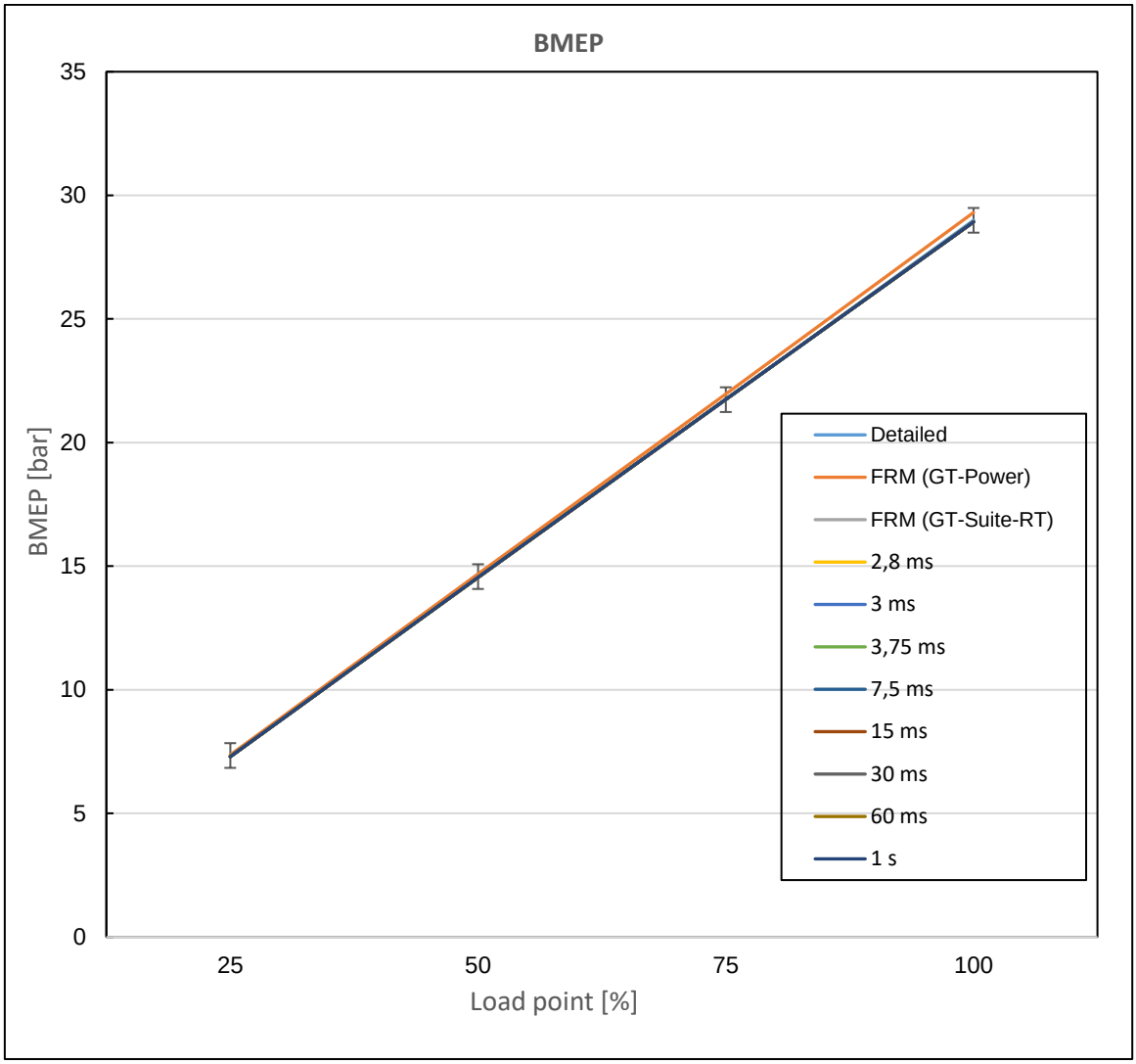

Figure 38. Comparison on BMEP.

In Figure 38, it is clearly seen that the BMEP values fall comfortably within the tolerances at all simulation fidelities. Interestingly, the deviation profile significantly differs between the FRM on regular GT-Power license and all other simulations, as is shown in Figure 39.

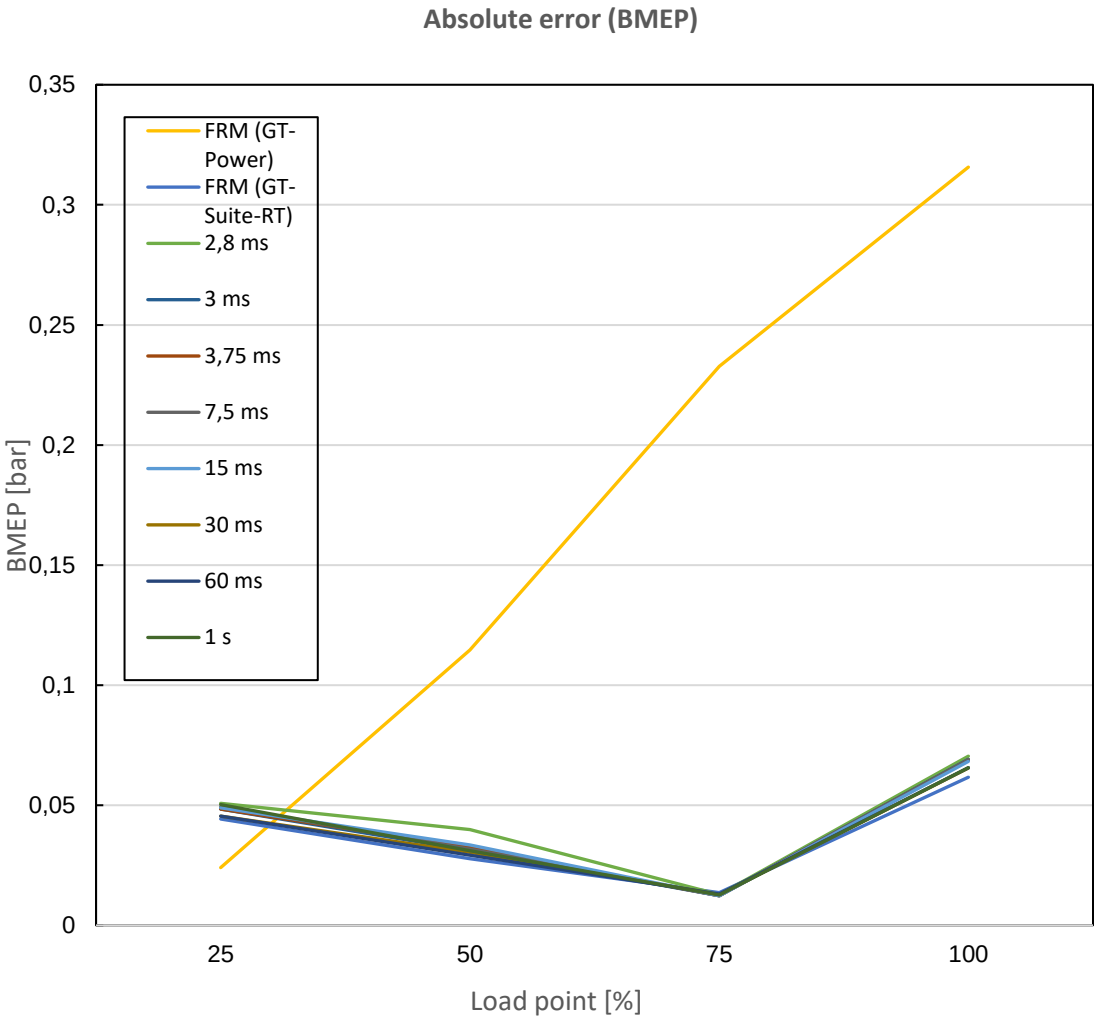

Figure 39. Absolute deviations of the BMEP value.

Finally, the exhaust temperatures are plotted on Figure 40.

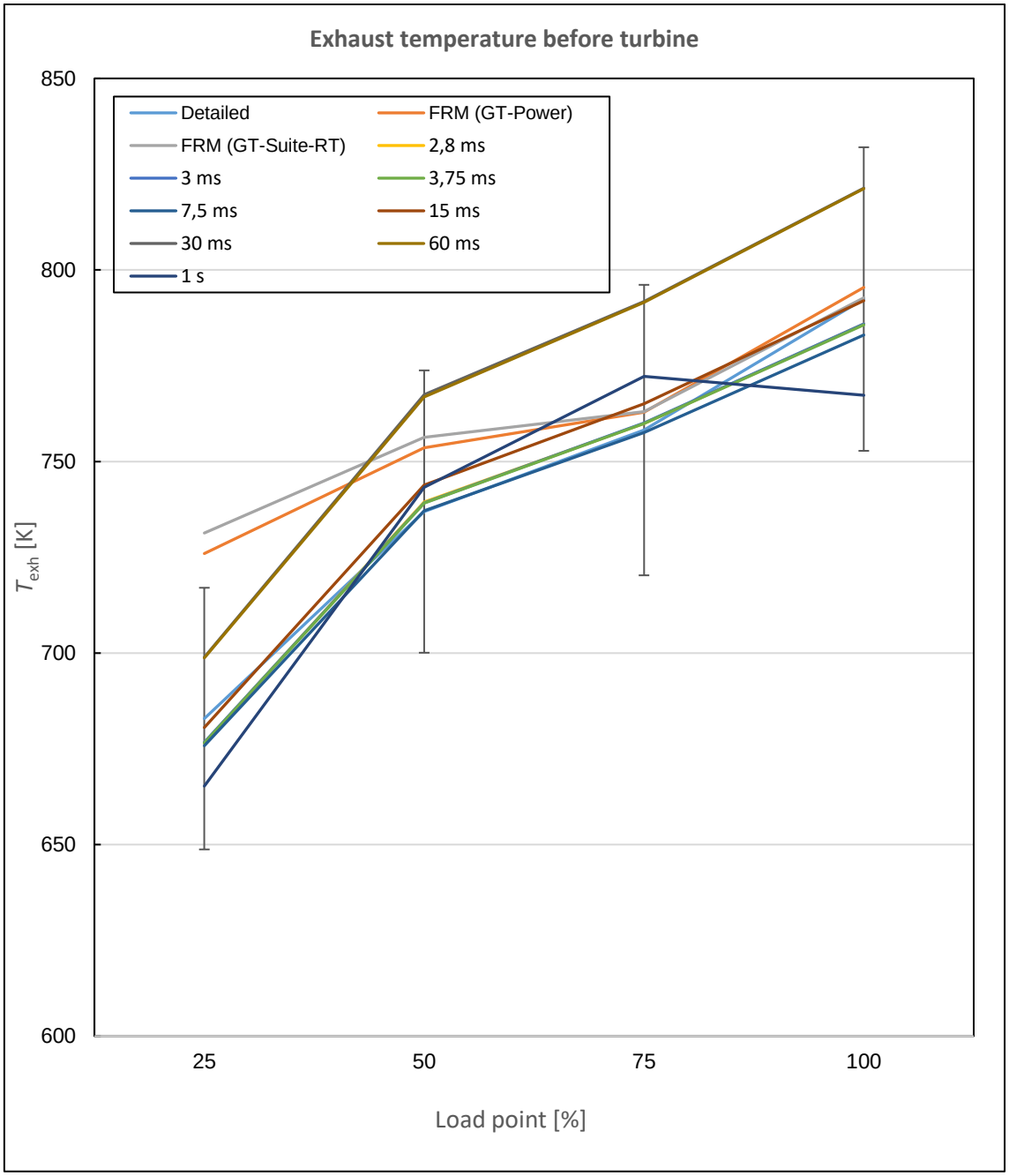

Figure 40. Comparison on exhaust temperatures.

Figure 40 shows that all simulations run on the real-time target machine fulfil the tolerance limit. Interestingly, they even surpass the original FRMs in accuracy. Another noticeable takeaway is that the best results are achieved by the simulation running at a sample time of 7,5 ms. This is further clarified in Figure 41.

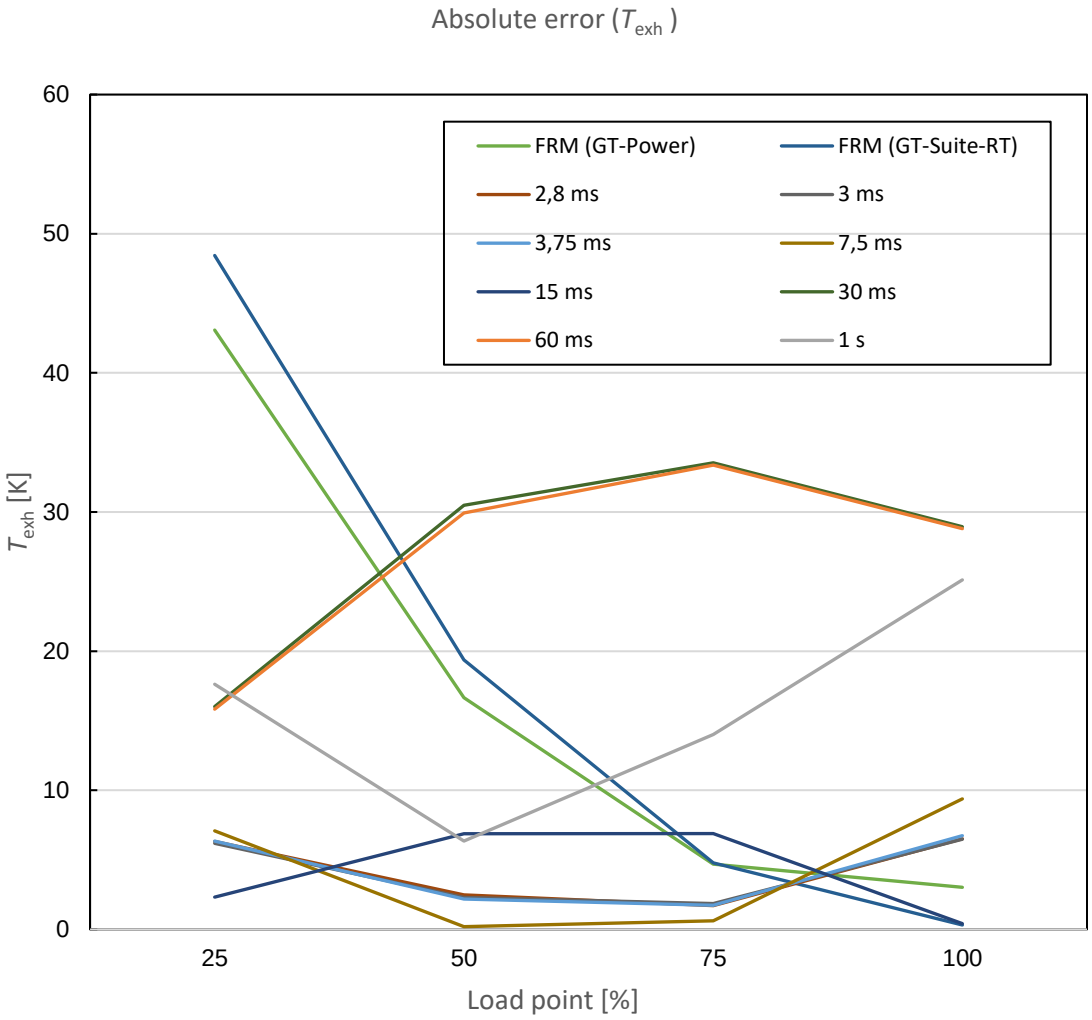

Figure 41. Absolute deviations of the T_{exh}value.

Apart from the 7,5 ms simulation being the most accurate, the results roughly follow the logic of lower sample times providing more accurate results.

4.1.3 Overall engine model accuracy

As a result of this validation process, it is evident that any alterations done to the models within this work pose no substantial threat to model accuracy. Upgrading the engine model caused only negligible changes in simulation results and the largest deviations consistently occurred during transients.

The validation simulations done on the real-time target machine showed the effect of adjusting the sample time. As expected, the largest differences were found in the simulations' capability to accurately reproduce in-cylinder pressure traces.

As a key finding, reducing the sample time does not automatically lead to more accurate results. As was discussed earlier, sample time selection is not necessarily just a matter of choosing the lowest computationally viable one, but rather it should be fitted to model particularities such as intended engine speed. This also underlines the purpose of this work. Even though better model accuracy might be reached by stretching the averaging period towards infinity, XiL simulations seldom allow such novelty. Thus, efforts must be made towards creating systems that can produce accurate results in real time.

4.2 Simulation speed

4.2.1 Standalone engine model

Figure 42 shows the results of the first simulation, where the task execution times measured from the engine output signal are plotted against a time axis. This sample time is equivalent to the one used to test the previous target machine.

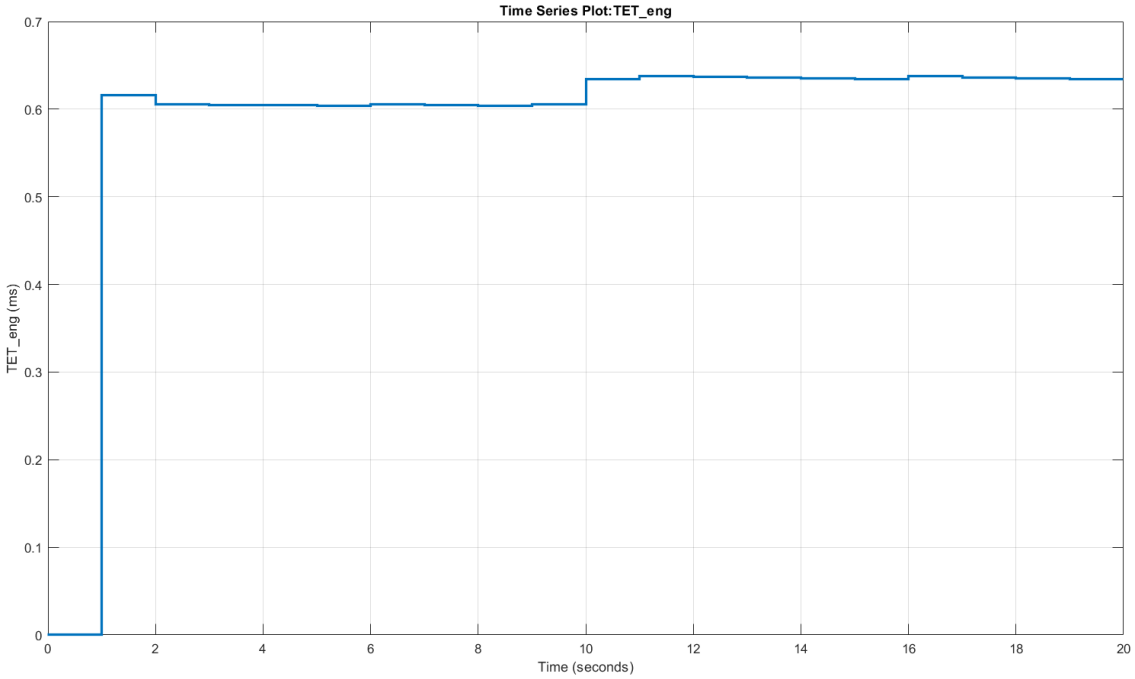

Figure 42. Results of engine model simulation at a time step of 1 s.

All task execution times stayed well below the 1 s mark. The next sample time was chosen to be 60 ms, which corresponds to a full turn of the crank axle when the engine runs at 1000 RPM. The results are plotted in Figure 43.

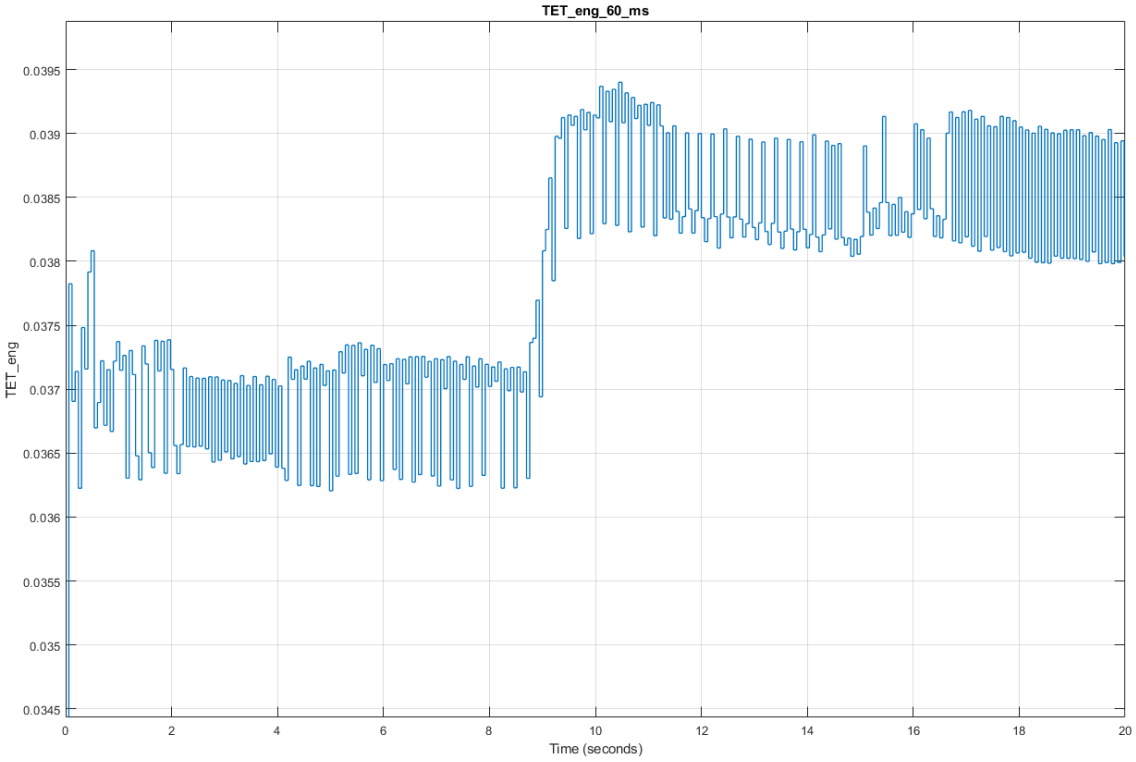

Figure 43. Results of engine model simulation at a time step of 60 ms.

Following the procedure of halving the sample time for each run until reaching low double-digits, Figure 44 shows the results of the following runs.

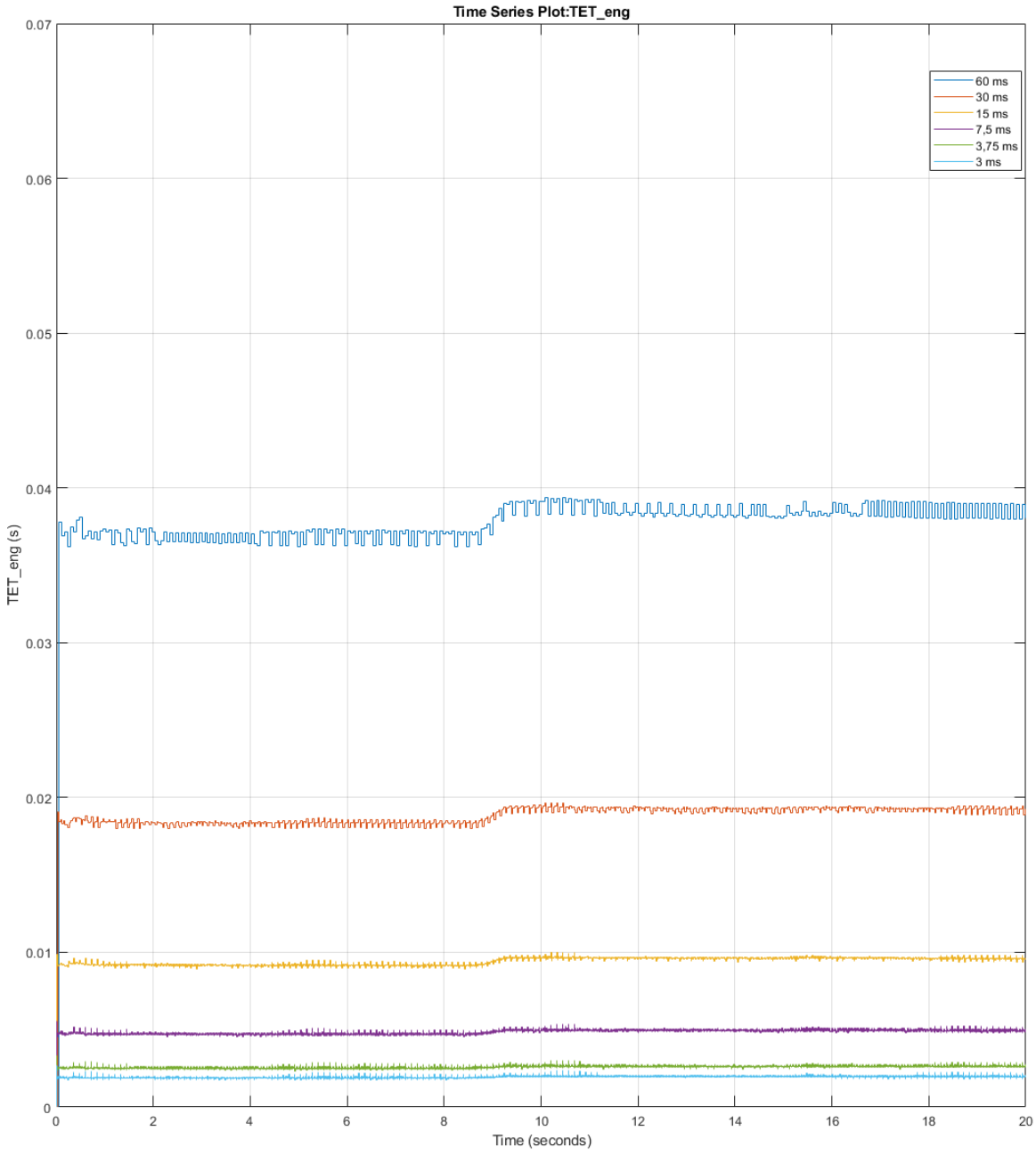

Figure 44. Combined results from sample times between a full crank cycle and double-digit degrees CA.

During the simulations plotted in Figure 44, no overruns occurred. The profile of the task execution times plot changed noticeably, and significant differences between individual time steps could be observed, such as in Figure 45.

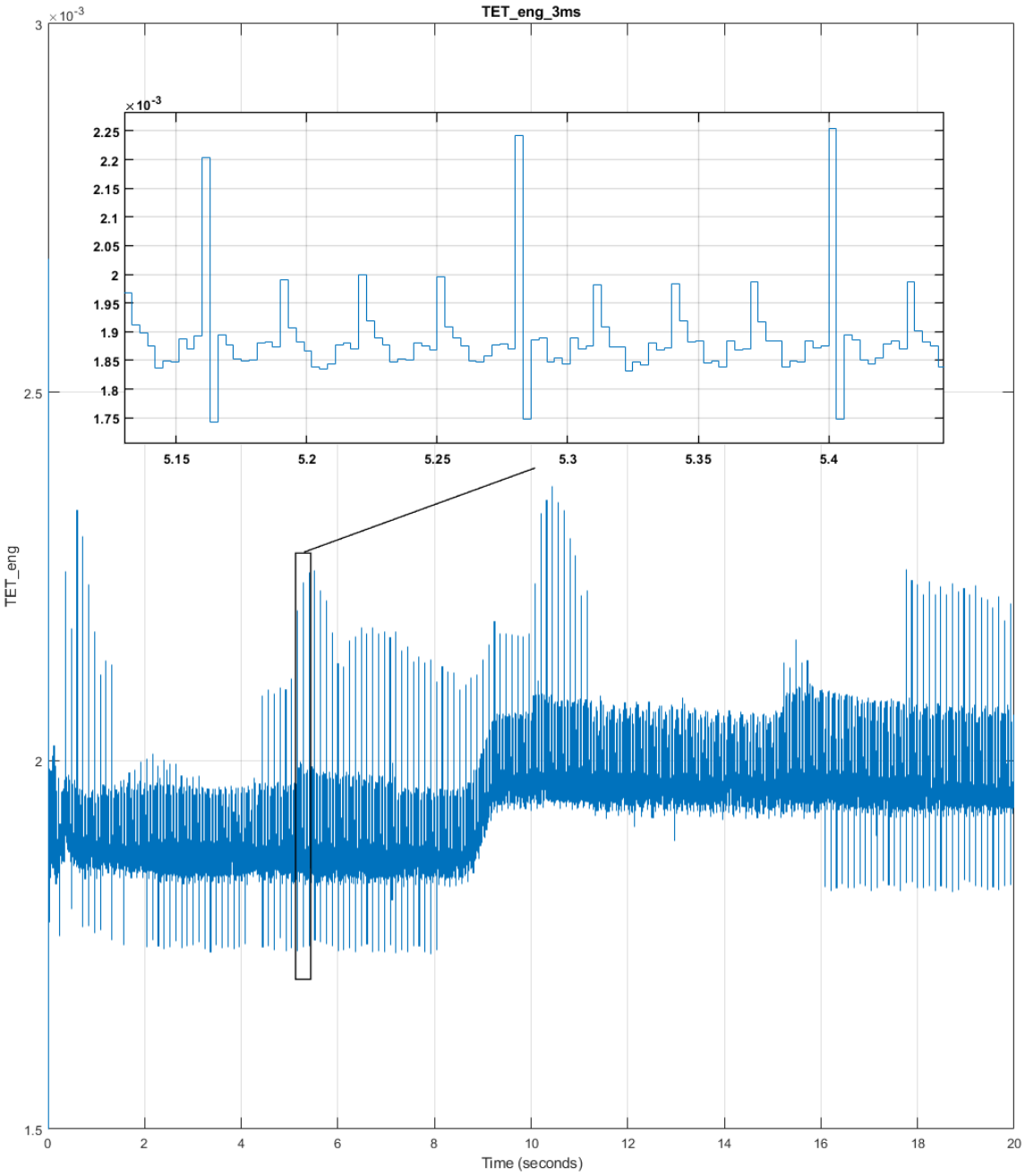

Figure 45. Results of engine model simulation at a time step of 3 ms.

Though all task execution times stay below the 3 ms threshold, there are clear spikes of which the larger ones correspond to the beginning of each engine cycle. The smaller ones occur at a quarter of this rate. The following simulations follow a similar trend. The first sample time to cause an overrun at the first time step was 2,17 ms, which corresponds to 13 degrees CA (Figure 46).

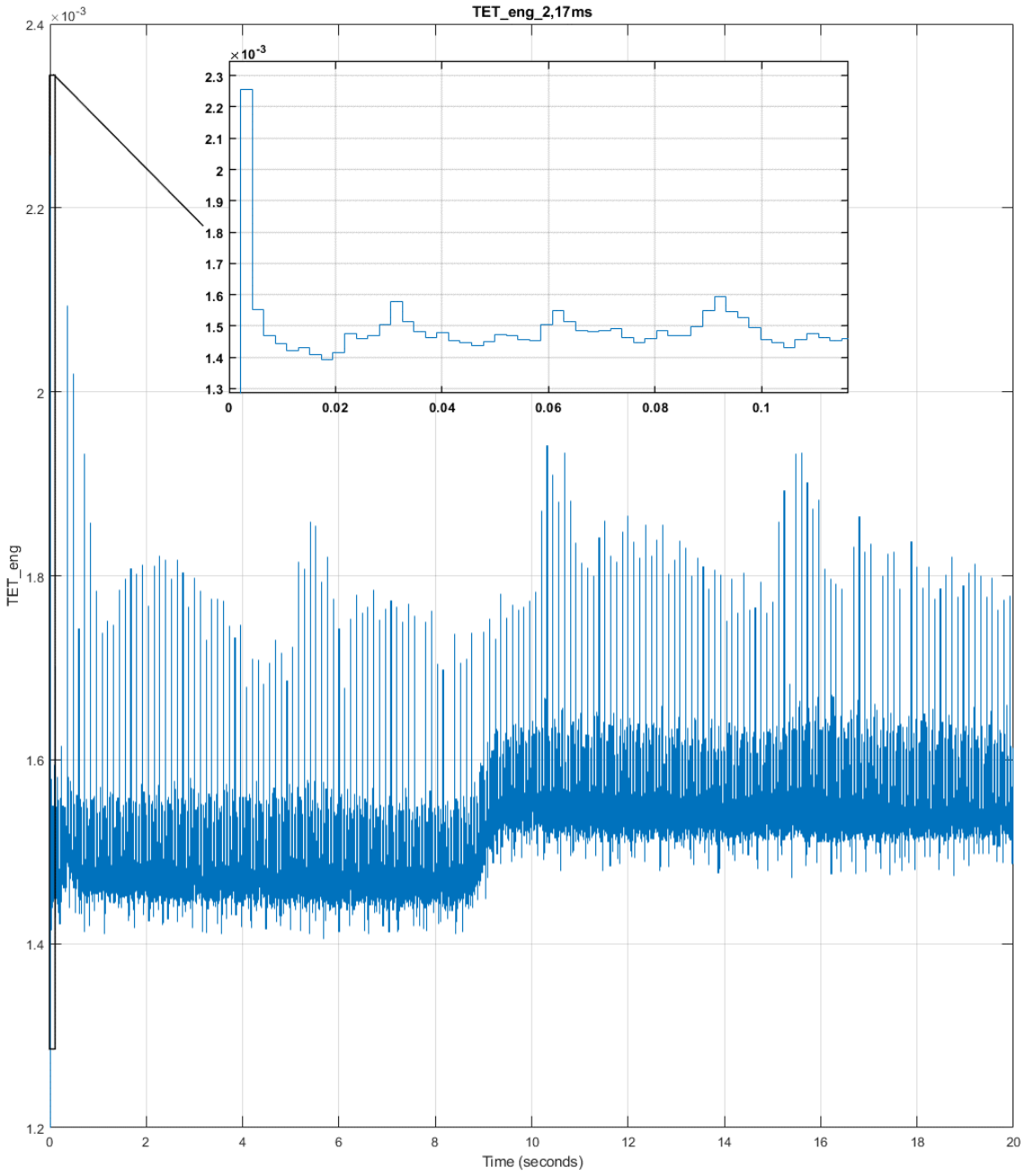

Figure 46. Results of engine model simulation at a time step of 2,17 ms.

After this, overruns occurred in each simulation run at a smaller sample time. Table 16 shows the number of overruns at each tested time step along with the maximum and average sample times.

Table 16. Engine model task execution times.

T_s (CA)	T_s (ms)	Over-runs	Max TET (ms)	Avg TET (ms)	Max TET of st	Avg TET of st
6000	1000,00	0	638,08	592,69	63,8 %	59,3 %
360	60,00	0	39,4	37,71	65,7 %	62,9 %
180	30,00	0	19,66	18,81	65,5 %	62,7 %
90	15,00	0	10,01	9,41	66,8 %	62,8 %
45	7,50	0	5,57	4,85	74,3 %	64,7 %
22,5	3,75	0	3,32	2,58	88,6 %	68,7 %
18	3,00	0	2,68	1,93	89,4 %	64,5 %
17	2,83	0	2,69	1,93	96,0 %	69,1 %
15	2,50	0	2,46	1,73	98,4 %	69,4 %
14	2,33	0	2,26	1,52	97,0 %	65,2 %
13	2,17	1	2,26	1,52	104,0 %	70,0 %
12	2,00	1	2	1,31	99,9 %	65,4 %
11	1,83	3	2,06	1,32	112,4 %	71,9 %
10	1,67	2	1,82	1,1	109,2 %	65,8 %
9	1,50	3	1,82	1,09	121,3 %	72,9 %
8	1,33	4	1,59	0,89	119,3 %	67,0 %
7	1,17	66	1,59	0,88	136,2 %	75,4 %
5	0,83	163	1,31	0,67	157,5 %	80,4 %
4	0,67	23 143	1,29	0,67	192,1 %	100,6 %
1	0,17	118 309	1,01	0,26	592,5 %	152,4 %

The results expectedly show few overruns at sample times straight after the first overrun, as the overruns occur on spikes within the first seconds of the simulation. Overruns begin happening consistently at a sample time of 0,83 ms, where most spikes cause one.

4.2.2 Combined model

As the engine model is expected to be the main computational load of the hybrid power plant simulations, the combined model was tested in a less rigorous manner. The computational load caused by the electrical system is not expected to significantly change and its sample time is a fixed 50 µs.

The measured task execution times of the engine model followed the same pattern as in the individual simulations, with a spike occurring at the beginning of each engine cycle. Measurements of the electrical system task execution times show that they pose no

issue in this setting, as they consistently stay well below their respective sample time threshold. Table 17 shows the results of these simulations.

Table 17. Combined model task execution times.

T_s (CA)	T_s (ms)	Overruns	Max TET (ms)	Max TET of st
6000	1000,00	0	692,90	69,29 %
360	60,00	0	42,68	71,13 %
180	30,00	0	21,61	72,03 %
90	15,00	0	11,29	75,25 %
22.5	3,75	0	3,59	95,74 %
18	3,00	0	2,82	93,99 %
17	2,83	112	3,15	112,62 %

4.2.3 Optimising sample time

The simulations were carried out to experimentally determine the limits of the particular combination of software and hardware in hybrid powertrain simulations. Thus, the system was pushed by gradually reducing the engine model sample time until overruns occurred.

A key finding was an expected one; the TETs are not uniform throughout the simulation period when operated on sample times below an equivalent of a full engine cycle at 1000 RPM. This has a straight correlation with the work of Wurzenberger et al. (2013), which showed a similar spike in TET at the beginning of each engine cycle. This underlines the importance of distinguishing between the TET of a time step and the average TET of the simulation period. As the control development of hybrid powertrains can be expected to require models capable of hard real-time performance, the time step-specific TETs are the main interest. However, the average TET of a given simulation does answer the following question: “If the computational load of each time step was the same, at what sample time would real-time capability be lost?”

In Figure 47, the maximum and average TETs of the engine model are presented as a fraction of the corresponding sample time. The sample times are presented on a

logarithmic scale and in units of crank angle degrees of an engine rotating at 1000 RPM. The hypothesis induced by the work of Wurzenberger et al. was proven right; the average TET of a 20 s simulation run is, without exception, significantly lower than the maximum TET. It should, however, be noted that the difference grows at lower sample times.

Assuming that 100 % of the computational expense of running the engine model on the Speedgoat target machine is caused by model calculations would lead to expecting a linear growth of TET fraction of sample time. Figure 47 clearly shows that this is not the case. Rather, the advancement resembles a second-order curve. This suggests that running the model has additional computational expenses that are not reliant on the sample time of the engine model, and which then use a larger and larger fraction of the available computing power as the sample time is lowered.

Interpreting the graph tells that in the case of the isolated engine model, hard real-time capability is lost when the sample time corresponds to 14 crank-angle degrees (CAD). If the TET of each time step were to be the same, the sample time could be comfortably reduced to 5 CAD. Even though the ultimate goal of this thesis is to determine the capabilities in simulating the entire power plant, these individual tests on the engine model set a foundation for evaluating the engine model’s effect on computational load.

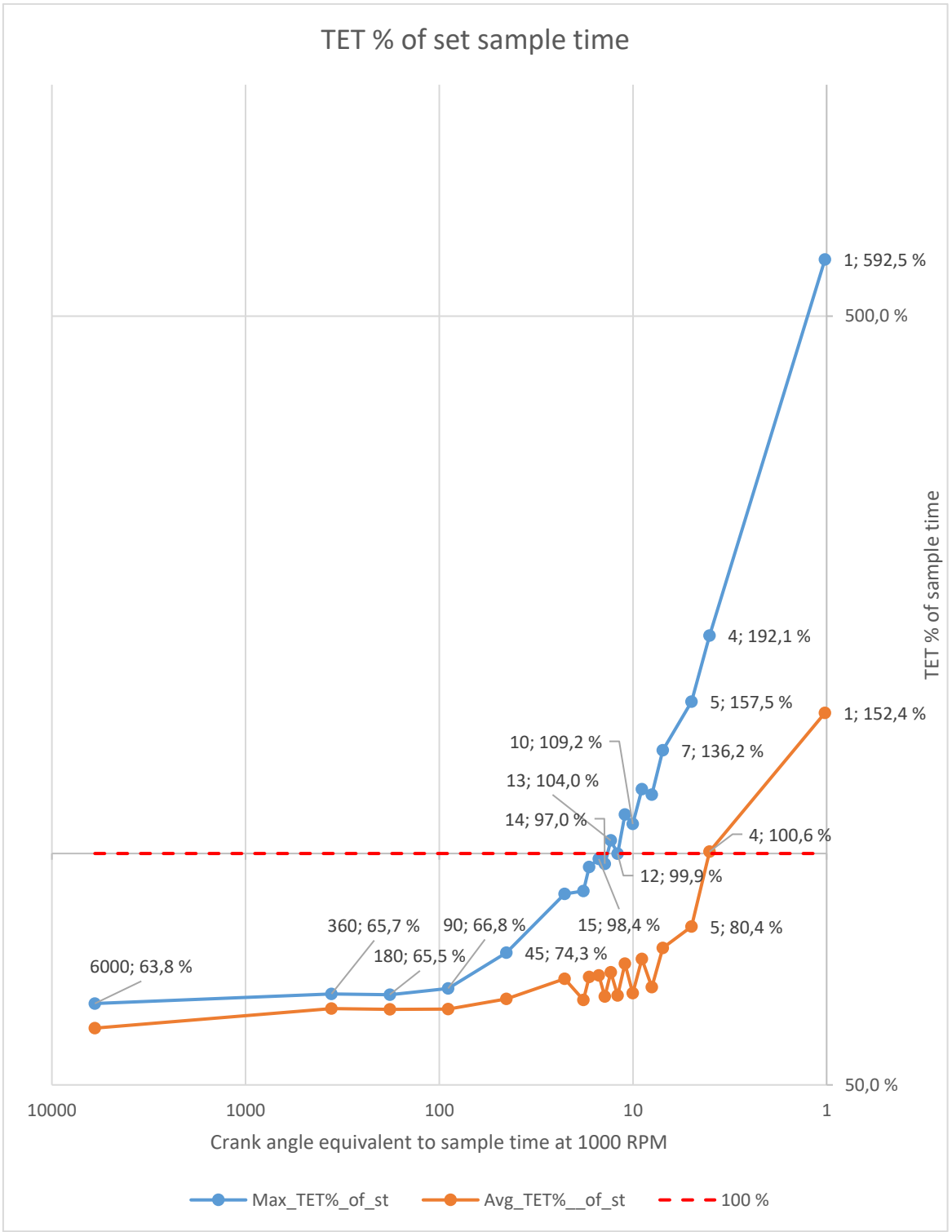

Figure 47. Task execution times of engine model in relation to sample times.

The simulations of the combined hybrid power plant model were carried out in a similar fashion. However, as the simulation results of the engine model and the largest tested

time steps could be used for guidance, less simulations were needed to find the limit. As with the isolated engine mode, Figure 48 shows that even with only 7 data points, the second order curve pattern is clear. Again presented as a function of time in the unit of CAD and on a logarithmic scale, the graph shows that the last tested sample time that fulfilled the hard real-time requirements was 18 CAD. The electrical system, which was running at a time step of 50 µs, posed no threat to the real-time capability, as the TETs consistently stayed comfortably below the time step even at lowest tested engine model sample times. The spikes at the beginning of each engine cycle could also been seen in the TET measurement of the engine model sample time in the simulations of the combined mode. Presenting these results separately was waived, as they bring no new information as such.

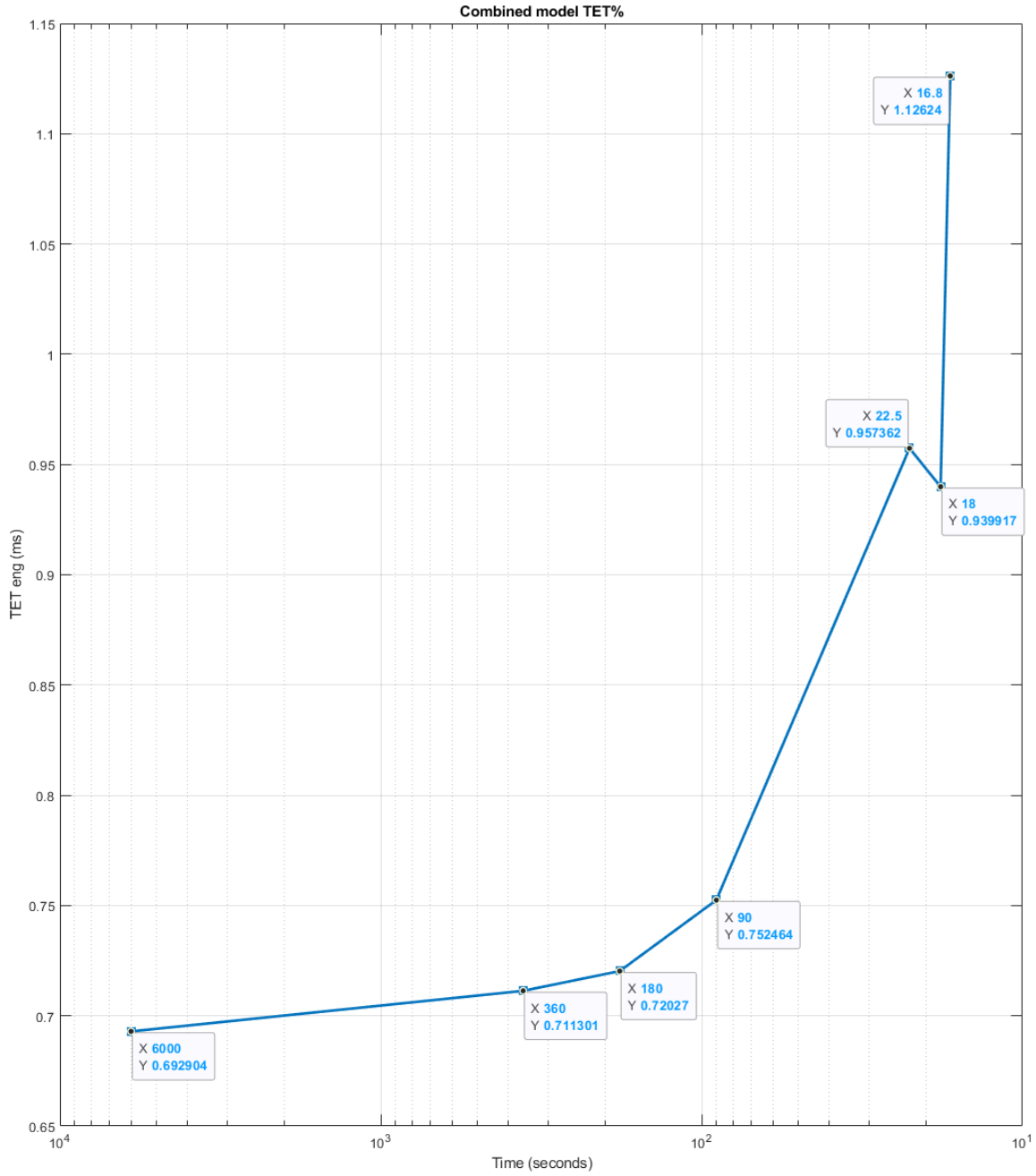

Figure 48. Maximum task execution times of engine model as a part of a power plant model in relation to sample times.

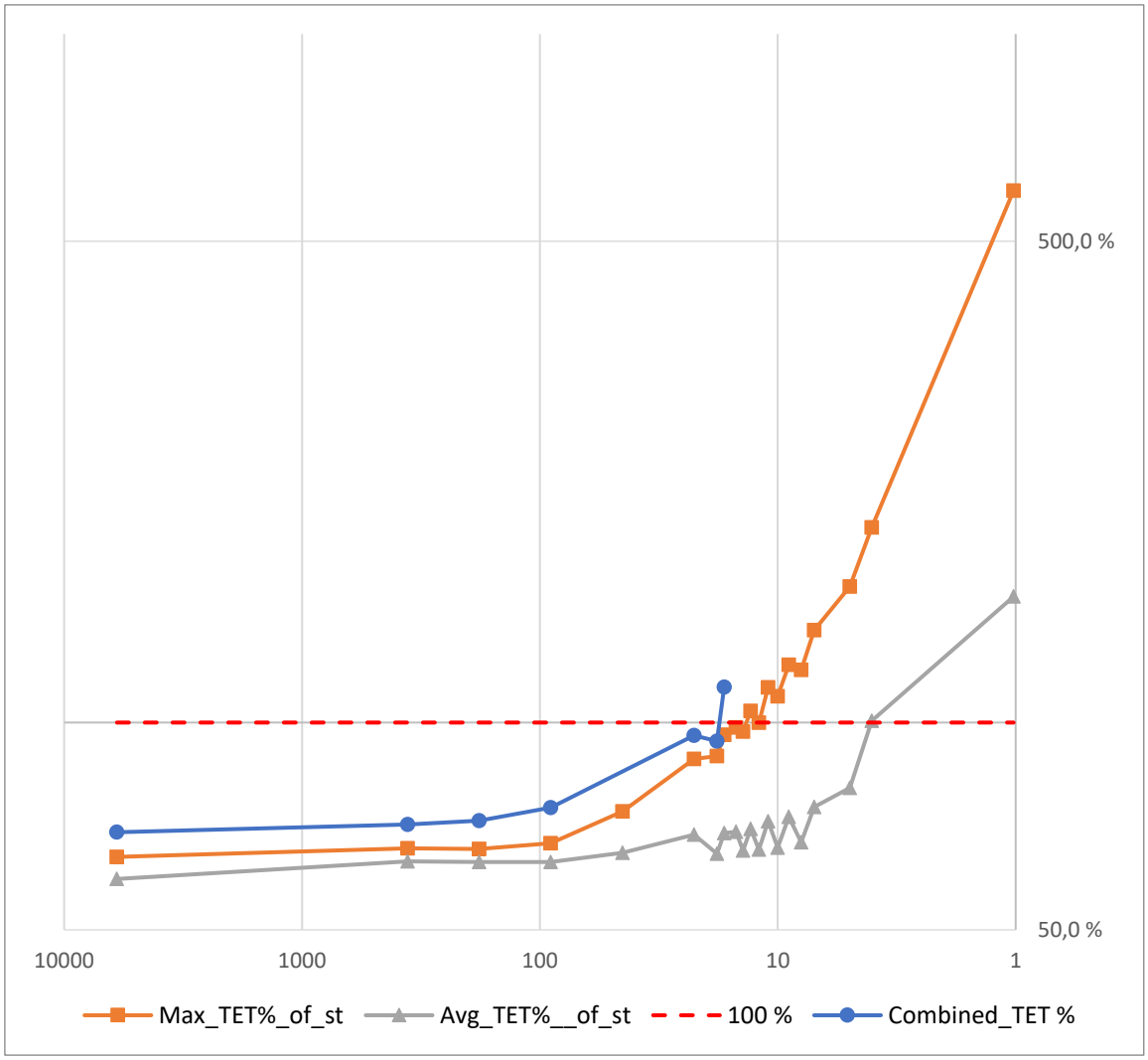

Figure 49. Engine model TET fractions, both isolated and connected to power plant model.

In Figure 49, the average and max TET of the isolated engine model as the max TET of the engine model when connected to the power plant model are shown. This comparison shows that introducing the electrical system to the simulation does have a clear effect on the engine model's performance. This is expected behaviour, as the models are running on shared resources. However, the moderate difference clearly shows that the main computational load of the combined power plant model is caused by the engine model.

These results surpass those of Söderäng et al. (2022) significantly, as the lowest sample time without overruns was reduced from 1 s to 3 ms. In future research, a key target should be bringing the maximum occurred TET within a simulation period closer to the average TET over the same period. The leads presented in this analysis point to the direction of beginning-of-cycle initialisation within the engine model.

Another question that inevitably arises from the graphs is the sawtooth-like pattern of the TET fractions at double-digit CADs. This pattern shows that the computational load induced by the models is not the only thing dictating real-time capability. The alternating pattern rather suggests that at certain sample times, the inter-model communication takes up additional resources.

In summary, the analysis has shown that while the isolated engine model can achieve hard real-time performance down to a sample time of 14 CAD, the combined hybrid power plant model is limited to 18 CAD. The primary computational load stems from the engine model, with the electrical system introducing additional complexities. This shows that a 1D flow solution can well be used in hybrid powertrain simulations without inevitably losing meaningful real-time capability.

4.3 System simulation

After compiling the FMUs, they were tested to verify that model accuracy has not suffered in the process. The results of this comparison can be found in Figures 50 and 51. Note, that Figure 50 presents the difference as a results plot with tolerance boundaries, whereas Figure 51 presents the difference as total deviation from original results.

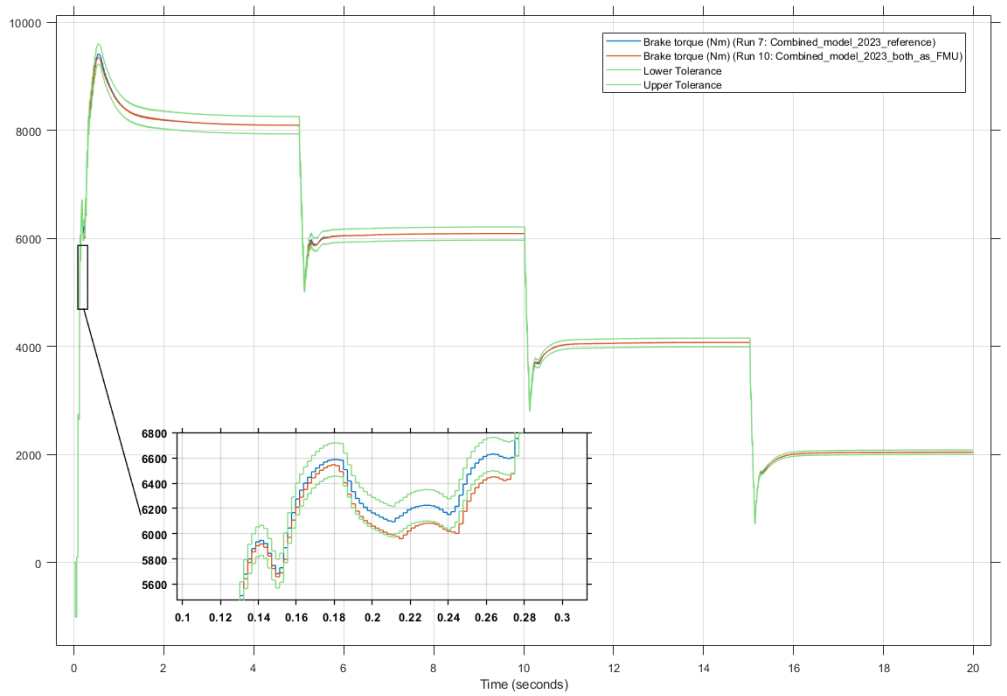

Figure 50. FMU deviation in engine torque.

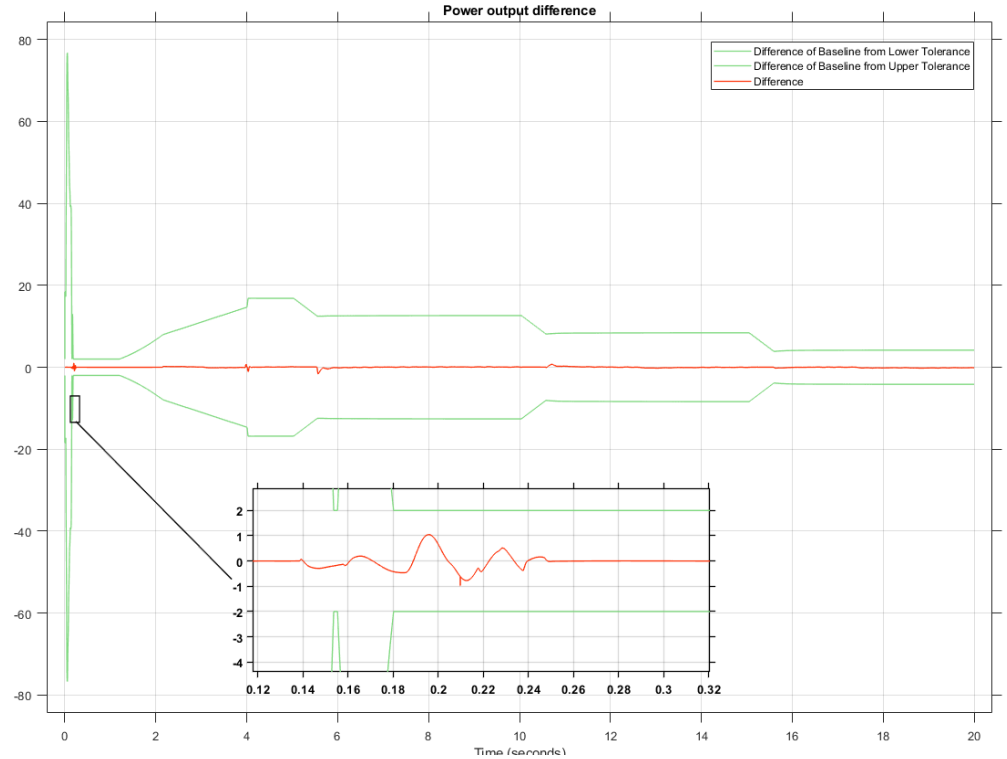

Figure 51. FMU deviation in total electrical power output.

The only time the 2 % relative and 2-unit total tolerance were violated was in the ramp-up period and specifically in the engine torque. With it being the input for the electrical system, the largest deviations in power output were expectedly found at the same time steps.

As the models fulfilled the expectations set for the system-level simulations, the FMUs were then run in a model-based system engineering (MBSE) tool, particularly Catia Magic by Dassault Systémes.

For the demonstration, the simulation was restricted to an open-loop setting, where the input structure was identical to the simulation model presented in Figure 20. This yielded an expected outcome, where a visual sanity check shows that the outputs react to changes in the inputs in a logical manner (Figure 52).

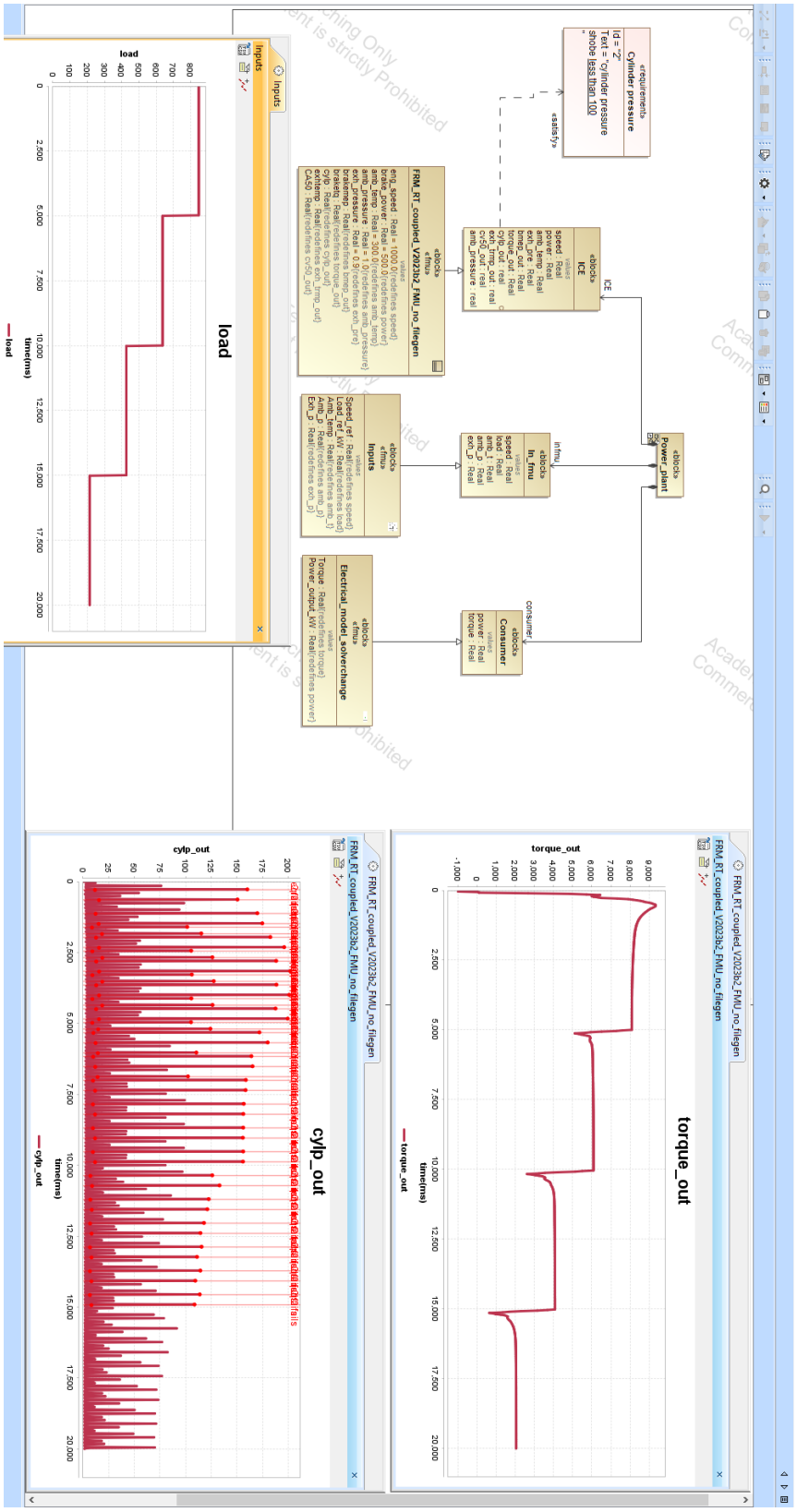

Figure 52. Overview of Catia Magic dashboard, where the previously used 20 s simulation period has been recreated.

Due to software limitations, the simulations were carried out using a sample time of 21 ms for the engine model. The process of validating model outputs is still underway, but preliminary results suggest that the model accuracy stays at a level that justifies its use over a simpler but computationally less demanding model.

Even though real-time operation was not a particular target of the demonstration, the system model was found to execute close to real time at a sample time of 2,1 ms when run on AVL:s Model connect co-simulation software. Recreating the 20 s simulation matching the power request profile of Söderäng et al. (2022) led to an average real-time factor of 1,6 over the simulation period, as is evident from Figure 53.

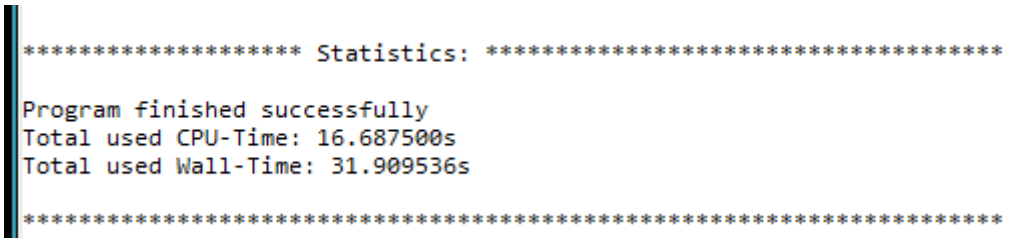

Figure 53. Statistics of a 20 s simulation period in Model connect.

As with Catia Magic, the validation process is underway, but preliminary results suggest that running the simulations in an FMI environment poses no systematic threat to model accuracy.

5 Discussion

RQ1: What computational restrictions does a Speedgoat Mobile target machine set for running the fast-running engine model and hybrid power plant model?

The primary objective of this thesis was to explore the capabilities of a newly acquired real-time target machine in simulating a hybrid power plant model. Specifically, the research aimed to determine the computational limits of running such a model on the upgraded target machine, pushing the system to its limits by progressively reducing the sample times until task execution time (TET) overruns occurred. Through detailed testing and analysis, the limits of this system were found, and some identifiers for predicting performance of next system iterations were established.

The testing strategy involved a series of experiments with both the isolated engine model and the combined hybrid power plant model. A key finding was that reducing the engine model sample time significantly impacted the TET's relation to the corresponding sample time; the computational demand does not follow the sample time in a linear fashion. The results demonstrated that the lowest sample times without overruns corresponded to 14 degrees crank angle degrees (CAD) for the isolated engine model and 18 CAD for the combined hybrid power plant model. Differences in the engine model average and maximum TET values underscore the importance of distinguishing between the indicators when evaluating the real-time capabilities of models.

Simulations revealed that TETs are not uniform throughout the simulation period, particularly when operated at sample times below an equivalent of a full engine cycle at 1000 RPM. This finding correlates strongly with the work of Wurzenberger et al. (2013), which also identified significant spikes in TET at the beginning of each engine cycle. As hybrid powertrain control development requires models capable of hard real-time performance, understanding these dynamics is essential.

Analysis of the combined hybrid power plant model demonstrated that introducing the electrical system to the simulation had a noticeable effect on the engine model's performance. This was anticipated due to the shared computational resources between the models. However, it was found that the primary computational load was still caused by the engine model, with the electrical system introducing additional complexities but not posing a significant threat to real-time capability.

These results significantly surpass those of Söderäng et al. (2022), as the lowest sample time without overruns was reduced from their 1 second to 3 milliseconds. This improvement is coined for the upgraded real-time target machine. The ability to achieve such low sample times without overruns indicates that more detailed and accurate simulations can be conducted, facilitating better control and optimisation of hybrid power plant systems.

RQ2: What are the necessary steps for evolving the model into a complete digital twin?

The development of a digital twin of the Vaasa Energy Lab powertrain, which requires high-fidelity models capable of operating in real-time, benefits from the improvements in computational performance demonstrated in this thesis. Achieving real-time performance is crucial for digital twins to accurately mirror physical counterparts and support effective control strategy development.

As the Speedgoat target machine is equipped with Modbus capabilities, it is compatible with the Energy Lab infrastructure. Thus, it can fulfil its role as presented in Figure 54.

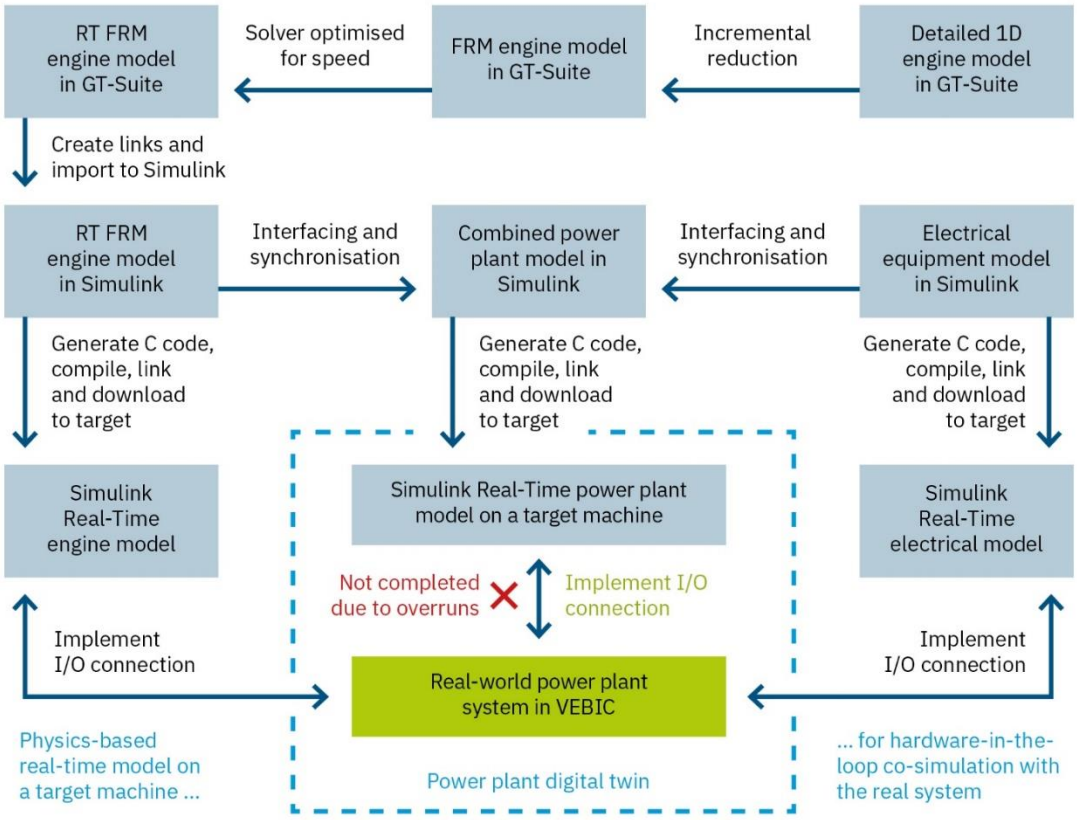

Figure 54. The process of creating a physics-based digital twin of a hybrid power plant (Söderäng et al., 2022).

As the HiL capability in terms of simulation speed was demonstrated within the scope of RQ1, the remaining work lies in integrating the Speedgoat into the Energy Lab. In practice, this means establishing Modbus communication between the target machine and the Energy Lab by physically connecting the devices and parametrising the Speedgoat to match the particular communication setup. Through these steps, the University of Vaasa will be in hold of all the necessary components to further evolve the HiL setup into a digital twin.

RQ3: What is the trade-off between simulation speed and accuracy while meeting real-time requirements and maximising sub-model fidelity?

The detailed engine model has been modified and adapted several times over the course of research activities in the University of Vaasa. At each stage, model accuracy has been re-validated, and so was the case for this work.

The real-time model was deployed on a Speedgoat real-time target machine and results from each tested sample time were compared to those of the detailed model, in the same scope as was done in the work prior to this. The results showed an expected outcome, where smaller sample times consistently forecasted better model accuracy. Some particularities regarding certain combinations of engine speed and sample time were identified and addressed.

Considering the research question, the trade-off between simulation speed and accuracy became clear. With the BMEP and exhaust gas temperature values, the effects of increasing the sample time were modest. In contrast to this, the effects on in-cylinder pressure recreation capability were even clearer. There, smaller sample times directly preceded better reproduction accuracy. Furthermore, at sample times over an engine cycle time, sensible and comparable results could not be produced. The significance of accurate in-cylinder pressure recreation becomes even more important as novel combustion technologies take over the field.

Even though the main trade-offs were identified and generally satisfactory results were obtained by using the lowest feasible sample time of 2,8 ms, there is room for improvement. The results suggest that further streamlining the model to once again reach lower sample times would lead to reaching the target tolerances even for in-cylinder pressure. These results also confirm that to enable higher-fidelity system-level XiL simulations, it is absolutely worthwhile to further pursue improvements on model accuracy, while still keeping the system real-time capable.

RQ4: What benefits does the improved tool provide for system-level simulations?

In the system-level modelling landscape systems are built out of existing or purpose-made sub-models, which often justifies use of non-predictive models or simple data-driven approaches for engine models. This work has shown that waiving model predictivity is not a pre-requisite but rather a shortcut. As the baseline for model fidelity has been set to match real-time requirements on the particular system, it acts as a landmark for future applications.

By recreating the system as interconnected FMUs, it was shown that its viability for system-level simulations is not dependent on host software. To further emphasise this, a reference simulation was run on Catia Magic and Model.connect MBSE tools. The simulation speed was found to be impressive especially in Model.connect, where an RTF of 1,6 could be reached without any additional optimisation measures.

The benefits of using a physics-based predictive engine model in system-level simulations are substantial. For one, repurposing the model to fit a new use-case scenario is a matter of adjusting the outputs. This is a direct result of the model being physics-based and fully predictive and thus not reliant on existing data, which considerably streamlines the repurposing process.

In the CASEMATE project, this quality is capitalised by using the model as a part of a system-level model of a hybrid powertrain. The model has proven invaluable in its flexibility, which enables easily modifying the model to fit e.g. different power requirement and available fuel scenarios. The results of this thesis in this regard suggest that the era of settling for coarse engine models in system-level simulations has ended.

6 Conclusions and outlook

To conclude, the main findings of this thesis are as follows:

- Reducing the sample time of the engine model significantly affects the task execution time (TET), with computational demand not scaling linearly with sample time.
- The lowest sample times without overruns were 14 crank angle degrees (CAD) for the isolated engine model and 18 CAD for the combined hybrid power plant model, showcasing the system's capability for high-resolution simulations .
- TETs are non-uniform throughout the simulation, particularly at sample times below an equivalent of a full engine cycle at 1000 RPM, with significant spikes occurring at the beginning of each engine cycle.
- The engine model remains the primary computational load in the combined hybrid power plant model, with the electrical system introducing additional complexities but not significantly impacting real-time performance.
- The Speedgoat target machine's connectivity and computational capabilities enable the Vaasa Energy Lab to further evolve the HiL-capable system into a digital twin.
- Smaller sample times consistently resulted in better model accuracy, particularly for in-cylinder pressure recreation. This highlights the importance of using the lowest possible sample times to achieve accurate simulations.
- Using a physics-based predictive engine model in system-level simulations provides substantial benefits, such as the flexibility to repurpose the model for different use-case scenarios.

Future research should focus on further reducing the maximum TET within a simulation period to bring it closer to the average TET. This could involve investigating the initialisation processes taking place at the beginning of each engine cycle, which appear to contribute significantly to the observed TET spikes. Improving the lowest feasible sample

time is expected to yield better model accuracy, even to the point of reaching the threshold of 5 % maximum deviation of detailed model.

Since the hybrid power plant model was last adjusted, the equipment of Vaasa Energy Lab has undergone changes, as one engine cylinder has been fitted with reaction-controlled compression ignition capability. This renders the engine model in need of updating, as the definition of a digital twin dictates that the physical and virtual systems must indeed match each other. An additional endeavour in this regard is the planned acquisition of a battery energy storage system (BESS) to match its virtual counterpart. Thus, some work is left to be done before the system can be judiciously titled a digital twin.

The practical applications and industry impact of this research are evident. By understanding the computational limits and optimising sample times, control strategies can be designed to operate within these constraints, ensuring reliable real-time performance. Following the model-based design (MBD) workflow, the resulting systems can act as drivers for streamlined product development. More importantly, these advancements are yet another step towards more sustainable and better optimised energy production. The models and simulations used in this research are ready for real-world implementation in various operational settings. Potential scenarios for deployment include power plants, marine vessels, and industrial applications where hybrid powertrains are used.

References

Aarenstrup, R. (2015). *Managing Model-Based Design*. https://se.mathworks.com/campaigns/offers/managing-model-based-design.html

AIAA. (1998). *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations (AIAA G-077-1998(2002))*. American Institute of Aeronautics and Astronautics, Inc. https://doi.org/10.2514/4.472855.001

Bondarenko, O., & Fukuda, T. (2020). Development of a diesel engine's digital twin for predicting propulsion system dynamics. *Energy*, *196*, 117126. https://doi.org/10.1016/j.energy.2020.117126

Bozza, F., De Bellis, V., & Dulbecco, A. (2020). Advanced 0D and QuasiD Thermodynamic Combustion Models for SI and CI Engines. In *1D and Multi-D Modeling Techniques for IC Engine Simulation*. SAE International. https://www.sae.org/publications/books/content/r-469/

Chalal, L., Saadane, A., & Rachid, A. (2023). Unified Environment for Real Time Control of Hybrid Energy System Using Digital Twin and IoT Approach. *Sensors*, *23*(12), Article 12. https://doi.org/10.3390/s23125646

Coraddu, A., Kalikatzarakis, M., Oneto, L., Meijn, G., Godjevac, M., & Geertsma, R. (2018, October 2). *Ship diesel engine performance modelling with combined physical and machine learning approach*. https://doi.org/10.24868/issn.2631-8741.2018.011

Coraddu, A., Kalikatzarakis, M., Theotokatos, G., Geertsma, R., & Oneto, L. (2022). Physical and Data-Driven Models Hybridisation for Modelling the Dynamic State of a Four-Stroke Marine Diesel Engine. In A. K. Agarwal, D. Kumar, N. Sharma, & U. Sonawane (Eds.), *Engine Modeling and Simulation* (pp. 145–193). Springer. https://doi.org/10.1007/978-981-16-8618-4_6

Coraddu, A., Oneto, L., Cipollini, F., Kalikatzarakis, M., Meijn, G.-J., & Geertsma, R. (2022). Physical, data-driven and hybrid approaches to model engine exhaust gas temperatures in operational conditions. *Ships and Offshore Structures*, *17*(6), 1360–1381. https://doi.org/10.1080/17445302.2021.1920095

CPU-World: CPU chart of modern Intel and AMD microprocessors. (n.d.). Retrieved 2 January 2024, from https://www.cpu-world.com/cgi-bin/CPU_Chart.pl

Crespi, N., Drobot, A. T., & Minerva, R. (2023). The Digital Twin: What and Why? In N. Crespi, A. T. Drobot, & R. Minerva (Eds.), *The Digital Twin* (pp. 3–20). Springer International Publishing. https://doi.org/10.1007/978-3-031-21343-4_1

Dillaber, E., Kendrick, L., Jin, W., & Reddy, V. (2010). *Pragmatic Strategies for Adopting Model-Based Design for Embedded Applications*. https://doi.org/10.4271/2010-01-0935

do Amaral, J. V. S., dos Santos, C. H., Montevechi, J. A. B., & de Queiroz, A. R. (2023). Energy Digital Twin applications: A review. *Renewable and Sustainable Energy Reviews*, *188*, 113891. https://doi.org/10.1016/j.rser.2023.113891

Domínguez-García, J. L., Gomis-Bellmunt, O., Trilla-Romero, L., & Junyent-Ferré, A. (2012). Indirect vector control of a squirrel cage induction generator wind turbine. *Computers & Mathematics with Applications*, *64*(2), 102–114. https://doi.org/10.1016/j.camwa.2012.01.021

Functional Mock-up Interface Specification. (n.d.). Retrieved 24 May 2024, from https://fmi-standard.org/docs/3.0/

Gamma Technologies. (2023). *Engine performance tutorials*.

Garre, C., Mundo, D., Gubitosa, M., & Toso, A. (2014). *Performance Comparison of Real-Time and General-Purpose Operating Systems in Parallel Physical Simulation with High Computational Cost* (SAE Technical Paper 2014-01–0200). SAE International. https://doi.org/10.4271/2014-01-0200

Gelernter, D. (1991). *Mirror Worlds: Or The Day Software Puts the Universe in a Shoebox... How it Will Happen and What it Will Mean*. Oxford University Press.

Ghenai, C., Husein, L. A., Al Nahlawi, M., Hamid, A. K., & Bettayeb, M. (2022). Recent trends of digital twin technologies in the energy sector: A comprehensive review. *Sustainable Energy Technologies and Assessments*, *54*, 102837. https://doi.org/10.1016/j.seta.2022.102837

Glaessgen, E., & Stargel, D. (2012). The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural*

Dynamics and Materials Conference. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2012-1818

Grieves, M. W. (2002). Conceptual ideal for PLM.

Grieves, M. W. (2023). Digital Twins: Past, Present, and Future. In N. Crespi, A. T. Drobot, & R. Minerva (Eds.), *The Digital Twin* (pp. 97–121). Springer International Publishing. https://doi.org/10.1007/978-3-031-21343-4_4

Grill, M., Bargende, M., Rether, D., & Schmid, A. (2010). *Quasi-dimensional and Empirical Modeling of Compression-Ignition Engine Combustion and Emissions*. 2010-01–0151. https://doi.org/10.4271/2010-01-0151

Grimmelius, H. (2003). Simulation models in marine engineering: From training to concept exploration. *COMPIT '03*, 502–515.

Hautala, S., Mikulski, M., Söderäng, E., Storm, X., & Niemi, S. (2022). 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*, 14680874221106168. https://doi.org/10.1177/14680874221106168

Heluany, J. B., & Gkioulos, V. (2024). A review on digital twins for power generation and distribution. *International Journal of Information Security*, *23*(2), 1171–1195. https://doi.org/10.1007/s10207-023-00784-x

Heywood, J. B. (Ed.). (2018). *Internal Combustion Engine Fundamentals* (2nd Edition). McGraw-Hill Education. https://www.accessengineeringlibrary.com/content/book/9781260116106

Hu, D., Wang, H., Yang, C., Wang, B., Duan, B., Wang, Y., & Li, H. (2024). Construction of digital twin model of engine in-cylinder combustion based on data-driven. *Energy*, *293*, 130543. https://doi.org/10.1016/j.energy.2024.130543

Isermann, R. (2003). *Mechatronic Systems*. Springer. https://doi.org/10.1007/1-84628-259-4

Isermann, R. (2014a). *Engine Modeling and Control: Modeling and Electronic Management of Internal Combustion Engines*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-39934-3

Isermann, R. (2014b). General Combustion Engine Models. In R. Isermann (Ed.), *Engine Modeling and Control: Modeling and Electronic Management of Internal Combustion Engines* (pp. 133–271). Springer. https://doi.org/10.1007/978-3-642-39934-3_4

Isermann, R., Schaffnit, J., & Sinsel, S. (1999). Hardware-in-the-loop simulation for the design and testing of engine-control systems. *Control Engineering Practice*, *7*(5), 643–653. https://doi.org/10.1016/S0967-0661(98)00205-6

Jones, D., Snider, C., Nassehi, A., Yon, J., & Hicks, B. (2020). Characterising the Digital Twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, *29*, 36–52. https://doi.org/10.1016/j.cirpj.2020.02.002

Kavi, K., Akl, R., & Hurson, A. (2009). *Real-Time Systems: An Introduction and the State-of-the-Art*. https://doi.org/10.1002/9780470050118.ecse344

Kelemenová, T., Kelemen, M., Miková, Ľ., Maxim, V., Prada, E., Lipták, T., & Menda, F. (2013). Model Based Design and HIL Simulations. *American Journal of Mechanical Engineering*, *1*, 276–281. https://doi.org/10.12691/ajme-1-7-25

Keskin, M. T., Grill, M., Chiodi, M., & Bargende, M. (2020). Virtual Engine Development: 1D- and 3D-CFD up to Full Engine Simulation. In *1D and Multi-D Modeling Techniques for IC Engine Simulation*. https://ieeexplore.ieee.org/document/9127608

Knaus, O., & Wurzenberger, J. C. (2020). System Simulation in Automotive Industry. In H. Hick, K. Küpper, & H. Sorger (Eds.), *Systems Engineering for Automotive Powertrain Development* (pp. 1–34). Springer International Publishing. https://doi.org/10.1007/978-3-319-68847-3_34-1

Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, *51*(11), 1016–1022. https://doi.org/10.1016/j.ifacol.2018.08.474

Laurén, M., Goswami, G., Tupitsina, A., Jaiswal, S., Lindh, T., & Sopanen, J. (2022). General-Purpose and Scalable Internal-Combustion Engine Model for Energy-Efficiency Studies. *Machines*, *10*(1), Article 1. https://doi.org/10.3390/machines10010026

Liu, J. W. S. (2000). *Real-Time Systems*. Prentice Hall.

Lucchini, T., & Wright, Y. (2020). 3D-CFD Combustion Models for SI and CI Engines. In *1D and Multi-D Modeling Techniques for IC Engine Simulation*. SAE International. https://www.sae.org/publications/books/content/r-469/

Madusanka, N. S., Fan, Y., Yang, S., & Xiang, X. (2023). Digital Twin in the Maritime Domain: A Review and Emerging Trends. *Journal of Marine Science and Engineering*, *11*(5), Article 5. https://doi.org/10.3390/jmse11051021

Mauro, F., & Kana, A. A. (2023). Digital twin for ship life-cycle: A critical systematic review. *Ocean Engineering*, *269*, 113479. https://doi.org/10.1016/j.oceaneng.2022.113479

Minetti, M., Bonfiglio, A., Benfatto, I., & Yulong, Y. (2023). Strategies for Real-Time Simulation of Central Solenoid ITER Power Supply Digital Twin. *Energies*, *16*(13), Article 13. https://doi.org/10.3390/en16135107

Mirfendreski, A., Schmid, A., Grill, M., & Bargende, M. (2016). *Presenting a Fourier-Based Air Path Model for Real-Time Capable Engine Simulation Enhanced by a Semi-Physical NO-Emission Model with a High Degree of Predictability*. 2016-01–2231. https://doi.org/10.4271/2016-01-2231

Mishra, C., & Subbarao, P. M. V. (2021). *A Comparative Study of Physics Based Grey Box and Neural Network Trained Black Box Dynamic Models in an RCCI Engine Control Parameter Prediction* (SAE Technical Paper 2021-01–0178). SAE International. https://doi.org/10.4271/2021-01-0178

Mobile real-time target machine | Speedgoat. (n.d.). Retrieved 24 January 2024, from https://www.speedgoat.com/products-services/real-time-target-machines/mobile-real-time-target-machine

Onorati, A., & Montenegro, G. (2020). *1D and Multi-D Modeling Techniques for IC Engine Simulation*. SAE International. https://www.sae.org/publications/books/content/r-469/

Paquin, J.-N., Bélanger, J., Chakraborty, S., Jalili-Marandi, V., Song, X., Westermann, D., & Simões, M. G. (2021). Real-time simulation applications for future power systems and smart grids. In *Artificial Intelligence for Smarter Power Systems: Fuzzy logic*

and neural networks (pp. 9–64). IET Digital Library. https://doi.org/10.1049/PBPO161E_ch2

Perabo, F., Park, D., Zadeh, M. K., Smogeli, Ø., & Jamt, L. (2020). Digital Twin Modelling of Ship Power and Propulsion Systems: Application of the Open Simulation Platform (OSP). *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, 1265–1270. https://doi.org/10.1109/ISIE45063.2020.9152218

Plummer, A. (2006). Model-in-the-Loop Testing. *Proceedings of The Institution of Mechanical Engineers Part I-Journal of Systems and Control Engineering - PROC INST MECH ENG I-J SYST C*, *220*, 183–199. https://doi.org/10.1243/09596518JSCE207

QNX. (n.d.). *Scheduling policies*. Retrieved 22 February 2024, from https://www.qnx.com/developers/docs/7.1/#com.qnx.doc.neutrino.prog/topic/overview_SCHEDS.html

Shafto, M., Conroy, M., Doyle, R., Glaessgen, E., Kemp, C., LeMoigne, J., & Wang, L. (2010). *Modeling, Simulation, Information Technology and Processing Roadmap*.

Singh, M., Fuenmayor, E., Hinchy, E. P., Qiao, Y., Murray, N., & Devine, D. (2021). Digital Twin: Origin to Future. *Applied System Innovation*, *4*(2), Article 2. https://doi.org/10.3390/asi4020036

Smith, P., Prabhu, S., & Friedman, J. (2007). *Best Practices for Establishing a Model-Based Design Culture*. https://doi.org/10.4271/2007-01-0777

Söderäng, E., Hautala, S., Mikulski, M., Storm, X., & Niemi, S. (2022). 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*, *266*, 115793. https://doi.org/10.1016/j.enconman.2022.115793

Speedgoat and Simulink Real-Time Workflow | Speedgoat. (n.d.). Retrieved 31 August 2023, from https://www.speedgoat.com/solutions/simulink-real-time-workflow

Steindl, G., Kastner, W., & Stangl, V. (2019). *Comparison of Data-Driven Thermal Building Models for Model Predictive Control*. *7*, 730–742. https://doi.org/10.13044/j.sdewes.d7.0286

Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., & Sui, F. (2018). Digital twin-driven product design, manufacturing and service with big data. *The International Journal of*

Advanced Manufacturing Technology, *94*(9), 3563–3576. https://doi.org/10.1007/s00170-017-0233-1

Wärtsilä Finland Oy. (2020). *Wärtsilä 20 Product Guide*.

Wei Li, Joos, G., & Belanger, J. (2010). Real-Time Simulation of a Wind Turbine Generator Coupled With a Battery Supercapacitor Energy Storage System. *IEEE Transactions on Industrial Electronics*, *57*(4), 1137–1145. https://doi.org/10.1109/TIE.2009.2037103

Wright, L., & Davidson, S. (2020). How to tell the difference between a model and a digital twin. *Advanced Modeling and Simulation in Engineering Sciences*, *7*(1), 13. https://doi.org/10.1186/s40323-020-00147-4

Wu, H., & Li, M.-F. (2016, April 5). *A Hardware-in-the-Loop (HIL) Bench Test of a GT-Power Fast Running Model for Rapid Control Prototyping (RCP) Verification*. https://doi.org/10.4271/2016-01-0549

Wurzenberger, J., Heinzle, R., Deregnaucourt, M.-V., & Katrašnik, T. (2013). A Comprehensive Study on Different System Level Engine Simulation Models. *SAE Technical Papers*, *2*. https://doi.org/10.4271/2013-01-1116

Xia, F., Griefnow, P., Klein, S., Tharmakulasingam, R., Balazs, A., Thewes, M., & Andert, J. (2017, October 25). *Crank Angle Resolved Real-Time Engine Modeling for HiL Based Component Testing*.