Transient engine simulations using the stochastic reactor model for driving cycle performance studies

Reddy Babu Siddareddy, MSc

Extended Abstract

Academic Supervisor
Prof. dr hab. inż. Wojciech Adamczyk

Industry Supervisor
dr inż. Michał Pasternak

September 2025

1 Scope and objectives

Optimising existing Internal Combustion (IC) engines remains a key engineering priority in terms of fuel economy, utilisation of alternative fuels, and exhaust emissions reduction as emission standards continue to become stringent worldwide. The optimisation process requires extensive testing to evaluate a variety of engine operating strategies. These strategies must be relevant for engine operation under actual transient driving conditions, such as those mimicked by the Worldwide harmonised Light vehicles Test Procedure (WLTP) [1] or the Non-Road Transient Cycle (NRTC) [2]. When performed on test benches only, the development and optimisation of IC engines are costly and time-consuming, creating a need for reliable simulation tools to support this process.

Accurate modelling and simulation of time-varying engine operating conditions during WLTP or NRTC require robust methods for simulating combustion and emissions formation during these real-world scenarios. Such methods must be computationally efficient to allow for effective application under transient conditions. On the other hand, they must be accurate enough to predict pollutant formation, as it is needed for WLTP or NRTC procedures. The Stochastic Reactor Model (SRM) of engine in-cylinder processes appears to be a suitable modelling approach for fulfilling this objective. In the context of transient engine simulations, it is of interest not only to simulate engine in-cylinder processes alone but also to simulate the behaviour of the complete engine powertrain, with exhaust aftertreatment, to allow for developing engine control strategies or simulations that closely mimic conditions during actual vehicle testing for certification purposes. Such a capability can be reached by integrating the SRM with Model-in-the-Loop (MiL) or Hardware-in-the-Loop (HiL) simulation platforms.

MiL refers to a model representation of physical systems in a software form. With regard to combustion engines, MiL testing typically involves validating engine control algorithms against detailed engine simulation models before any physical hardware is introduced. This method is important for transient engine simulations, since it allows control strategies to be developed and refined under transient operating conditions, all within a safe and fully virtual environment [3, 4]. In turn, HiL refers to a system representing physical hardware, the Electronic Control Unit (ECU) and software models. With regard to combustion engines, HiL testing is a validation method where real engine control hardware is tested against simulation models running in real-time. For transient engine simulations, HiL provides a way of testing engine hardware and finding strengths and vulnerabilities when operating under transient conditions, without the need for a fully assembled test bench, ensuring that control units perform reliably under realistic operating conditions [5, 6].

The use of SRM as an integrated toolchain with MiL/HiL platforms appears promising because of the expected flexibility to work with different engines and fuels and in co-simulation with models for other components of vehicle systems. Furthermore, the 0D formulation of the SRM enables a low computational cost while ensuring high accuracy of predicting pollutant formation due to the application of detailed reaction kinetics. Therefore, such an integrated toolchain should contribute to the development of cleaner and more efficient IC engine technologies, which was the motivation behind this work.

The SRM has been proven to be an efficient approach for simulating IC engines under steadystate conditions [7, 8, 9, 10], but so far, it has not been applied for simulations under transient operating conditions like those in WLTP or NRTC. It also has not conformed to Functional Mock-up Interface (FMI) and Functional Mock-up Unit(FMU) standards, which are essential for deploying them in environments like HiL and MiL. This interoperability gap hampers the effectiveness of SRM in real-time simulations as demanded by HiL platforms. Addressing these issues is crucial to unlocking the full potential of the SRM for comprehensive simulations under transient conditions. Regarding these gaps, the objective of this work is to develop a simulation toolchain that enables transient engine simulations based on the SRM technology. The defined objective is graphically presented in Figure 1.

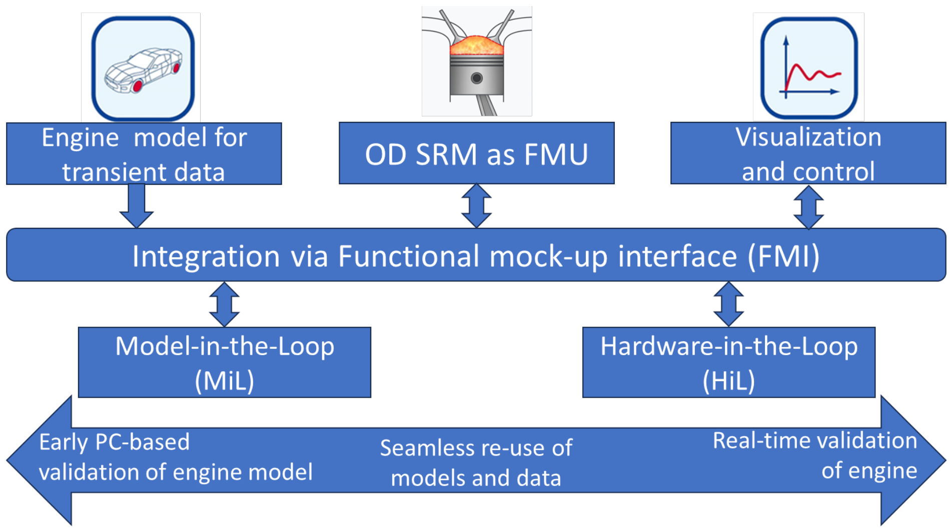

FIGURE 1: Concept of an integrated toolchain featuring 0D SRM, MiL/HiL, and FMI/FMU technology for transient driving cycle simulation studies.

Specifically, the work focuses on developing an FMU via FMI tailored to the SRM to enable integration into external simulation environments, co-simulation platforms, and real-time testing setups such as HiL and MiL. To accomplish this task, the existing SRM codebase has been updated to accommodate transient simulations by incorporating time-varied engine parameters such as speed, load, injection strategy, and valve timing that vary during transient operation. Furthermore, a framework was needed to generate an FMU for the SRM using the FMI standard and implementing an FMI-compliant interface. Finally, the framework had to be streamlined to minimise computational costs, ensuring real-time feasibility on stand-alone, MiL and HiL platforms while retaining combustion and emissions prediction capabilities of the SRM for Compression-Ignition (CI) and Spark-Ignition (SI) engines.

Overall, the framework sketched in Figure 1 is designed for portability and adaptability in MiL and HiL environments, providing standard FMI/FMU interfaces for seamless integration into realtime workflows. This approach bridges the gap between physical accuracy and computational efficiency, making SRM simulations of engine performance and exhaust emissions practical for contemporary engine testing, calibration, and virtual validation applications under transient conditions.

2 Methods and tools

The toolchain proposed in this work integrates a 0D SRM of engine in-cylinder processes, with tabulated chemistry for combustion and emission formation, into MiL and HiL platforms using the FMI standard. By encapsulating the SRM within FMU, the framework enables cross-platform simulations, bridging the gap between physical modelling accuracy and practical applicability in engine development [11, 12, 13].

2.1 The Stochastic Reactor Model (SRM)

The SRM is a zero-dimensional model of physical and chemical in-cylinder processes in IC engines. It regards the gas within the cylinder as a collection of notional particles. The particles are allowed to interact by mixing with each other and exchanging heat with the walls of the cylinder. Each such notional particle is characterised by temperature, mass and species concentration. Thus, in a sense, each particle can be understood as a realisation of the flow at a certain time and location in the actual combustion chamber. The particles are subjected to chemical reactions according to the applied reaction mechanism. All the particle-based processes are calculated sequentially using an operator splitting loop method. Mixing between particles is a modelled process, using different types of mixing models. The common feature of these models is the necessity of modelling the mixing time scale, which governs the frequency at which particles mix with each other and decides finally about mixture inhomogeneity in temperature and composition that in turn influences the combustion process and emissions formation [14]. A schematic visualisation of the SRM concept is shown in Figure 2.

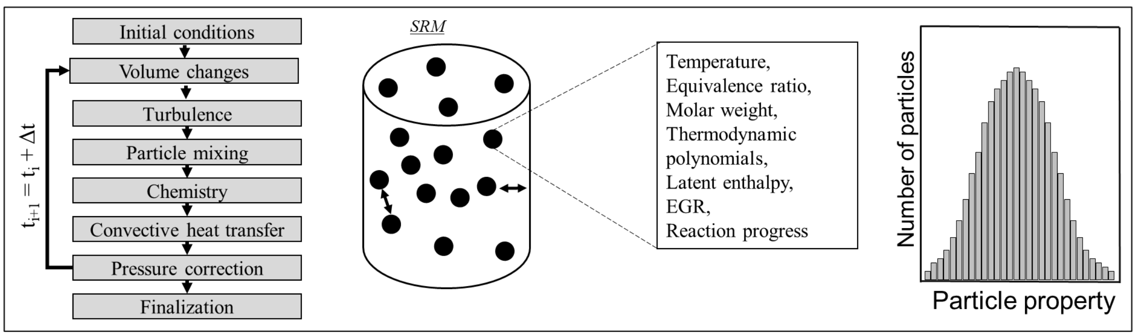

FIGURE 2: Schematic visualisation of the SRM concept for IC engines [15].

In this work, the coalescence-dispersion type mixing model is used with stochastic-based selection of particles for mixing. The history of mixing time scales is obtained from the turbulence model as the ratio between the turbulent kinetic energy k and its dissipation ϵ. Combustion and emissions formation are calculated based on he detailed reaction kinetics mechanisms, which are used in the form of tabulated chemistry employing combustion progress variable (CPV). The use of tabulated chemistry helps to reduce the computational cost to milliseconds for a complete engine working cycle without deteriorating the quality of results. The SRM can simulate premixed combustion with flame propagation that is relevant to SI engines, and non-premixed combustion relevant to CI engines [14].

2.2 FMU and SRM integration

In this work, the SRM is compiled as an FMU to enable integration with MiL/HiL environments. Co-simulation is employed, where the SRM runs as an independent solver communicating with an external simulation master (e.g., MATLAB/Simulink or dSPACE ASM) through standardised signal exchange. This approach allows modular linking of the SRM with other engine subsystems while preserving real-time operation and computational efficiency. The FMU acts as a container for the SRM solver and is referred to as FMU/SRM.

The FMU is built according to the FMI 2.0 co-simulation standard. In this setup, the SRM executes independently, taking input signals and sending and output variables at fixed communication intervals. Input signals include engine speed, valve timing, intake/exhaust pressures and temperatures, EGR rate, fuel mass per injection, and air-fuel ratio, as well as control signals for valve timing and injection strategy. Output signals comprise cycle-resolved engine metrics such as peak pressure, IMEP, torque, and exhaust emissions.

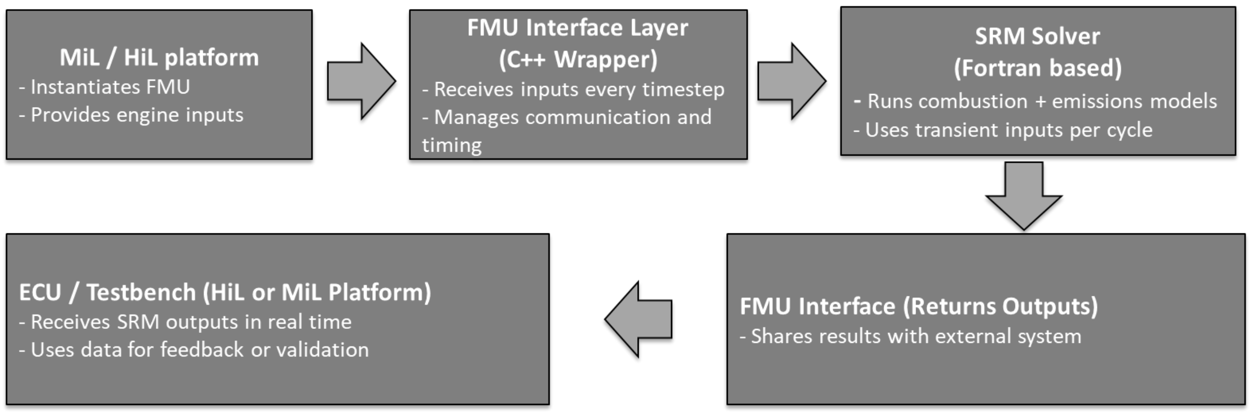

FIGURE 3: Simulation workflow with FMU-based SRM.

The FMU is created by pairing the SRM solver code with an FMI-compliant XML manifest and a dynamic library. A co-simulation master manages time synchronization and solver step alignment, ensuring accurate communication with external tools (Figure 3). The FMU is implemented in C++ as a wrapper around the Fortran-based SRM solver. The wrapper provides standard FMI 2.0 interfaces for initialization, input/output handling, time-stepping, and error reporting. The Fortran solver executes the combustion cycle computations and returns engine performance and emissions results. Data buffering ensures that if the SRM lags during a cycle, the most recent outputs are provided to maintain continuity. This modular setup allows testing different SRM solvers without rebuilding the FMU.

2.3 ANN-based meta-model

To further reduce computational costs below real-time, an artificial neural network (ANN) based meta-model of the driving cycle simulator was developed. The ANN is trained using data generated by the physics-based simulator employing the SRM of the driving, making this development preliminary. A three-layer network is implemented in Python using Keras, TensorFlow, and scikit-learn [16, 17, 18], with 128 neurons in the input layer, 64 in the hidden layer, and an output layer matching the number of predicted quantities. ReLU activation introduces non-linearity, and the Adam optimizer [19] performs stochastic gradient descent. The workflow involves training the network on transient engine data, predicting engine-out results, and iteratively updating the model until the prediction error falls below an assumed threshold of 5%.

2.4 Simulation toolchain

The overall toolchain proposed in this work, along with its components, is schematically shown in Figure 4. This unified simulation framework enables real-time transient engine simulations by combining the SRM, tabulated chemistry, FMU interfaces, an ANN-based surrogate model, and their integration within MiL and HiL testing platforms.

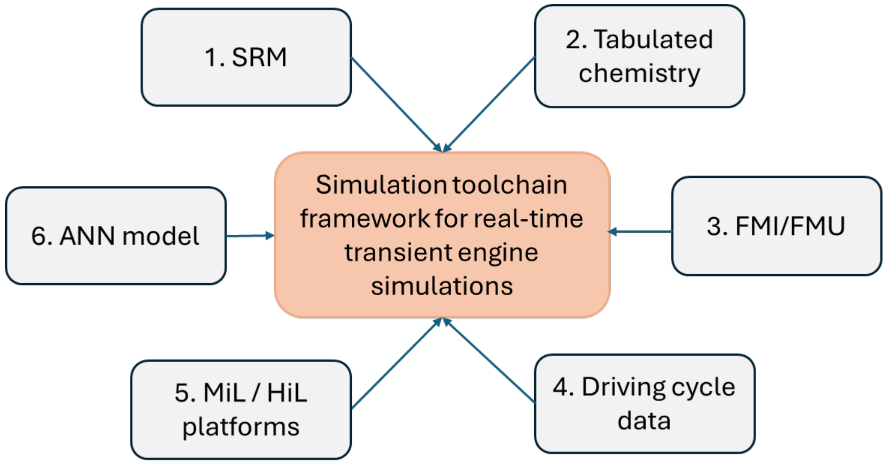

FIGURE 4: Components of the developed toolchain for transient engine simulation.

TABLE 1: Mapping of simulation toolchain elements to technical objectives.

Diagram element	**Role in in the toolchain**
SRM	0D model of engine in-cylinder processes
Tabulated chemistry	Modelling of combustion and pollutant Fuel surrogate formulation
FMI/FMU	Interfacing, modularity and system-wide compatibility
Driving cycle data	Inputs to run engine transient driving cycle
MiL/HiL platforms	Integrity with 3rd party testing platform
ANN meta-model	A meta-model for computational cost reduction

The framework is modular, allowing each component to operate independently while communicating through the FMU interface. Driving cycle data provides transient inputs, which are processed by the SRM, while the FMU ensures synchronisation and portability across simulation platforms. At the core, transient engine simulations are coordinated by the FMU interface, which integrates all components into a unified, scalable, and portable simulation environment. This toolchain provides a methodical approach for physics-based engine modelling suitable for real-time and system-level applications.

3 Results

The developed toolchain employing the SRM was applied in both stand-alone mode and integrated MiL/HiL environments for CI and SI engines under transient driving conditions of WLTP and NRTC. The content of this chapter is based on the published results [15, 14, 20, 21]. It begins with the integration of the SRM-based engine model via FMU/FMI technology into MiL and HiL platforms, forming a complete physics-based driving cycle simulator. Next, predictions of engine performance and exhaust emissions for an on-road CI engine under the WLTP cycle are presented, including driving-cycle-to-driving-cycle variability of exhaust emissions. Then, SI engine simulations during cold-start phases and co-simulation of the engine and catalyst are presented. Finally, results from the ANN-based meta-model of the SRM-driven simulator are presented.

3.1 MiL/HiL platforms-based engine simulations

This section demonstrates the toolchain's real-time performance, functionality and integrity within MiL and HiL platforms during a standardised driving cycle from the WLTP. The engine model built in the SRM and driving cycle data utilised in this study are from the dSPACE ASM package. The engine is an SI engine fuelled with gasoline and employs a port fuel injection system. Figure 5 shows the real-time MiL/HiL execution in dSPACE ControlDesk, which records engine-out emissions and key performance parameters during a transient WLTP cycle. In turn, Figure 6 presents simulated results for maximum in-cylinder pressure and the cumulative emissions of CO_2, CO and uHC, over a 1800s WLTP driving cycle, using both MiL and HiL frameworks.

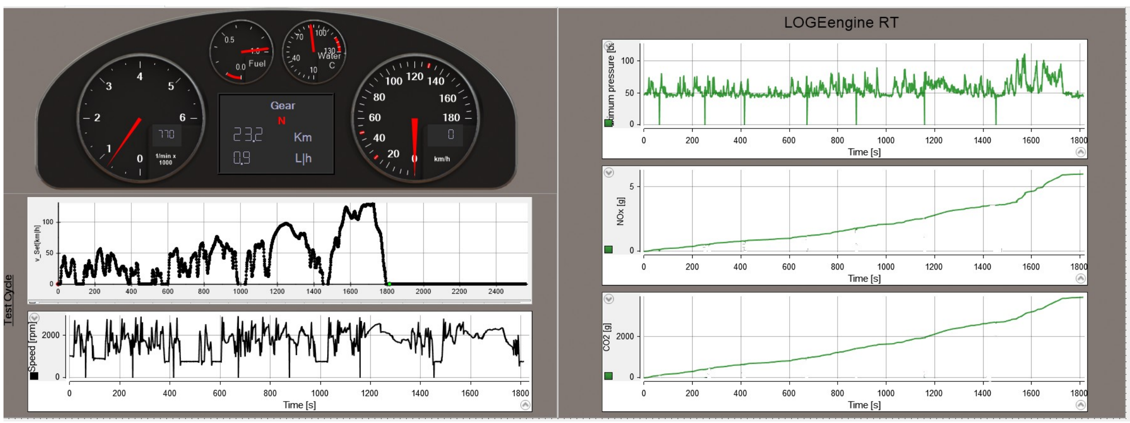

FIGURE 5: Dashboard view of the MiL/HiL simulations in dSPACE configurationDesk [6, 22].

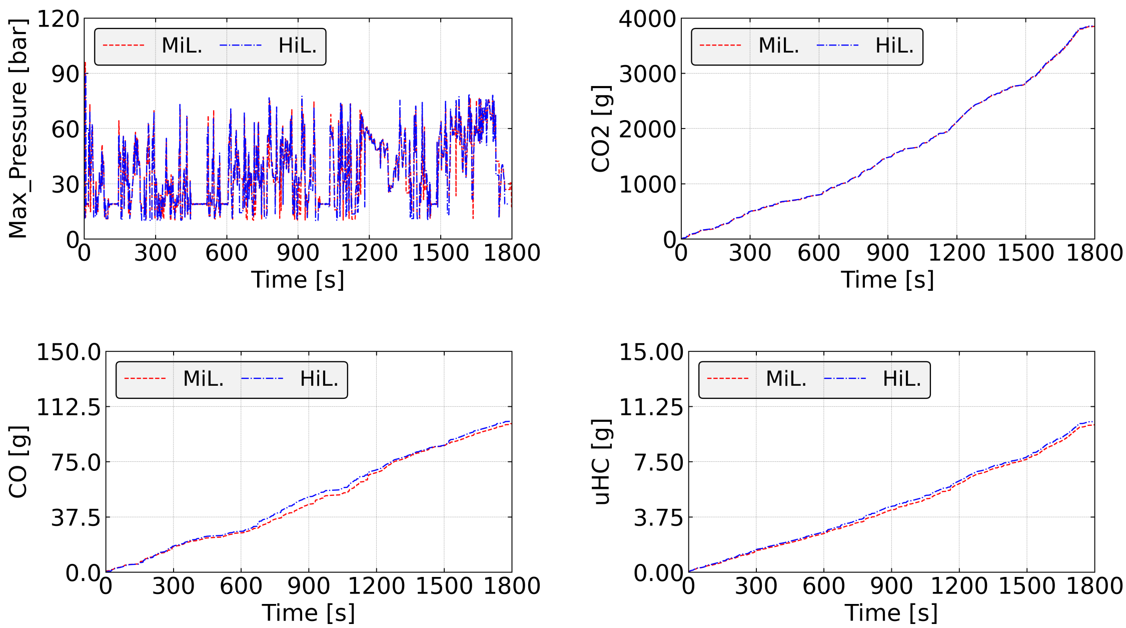

FIGURE 6: Results from MiL and HiL simulations employing the SRM.

The results confirm the flexibility and reliability of the developed toolchain across simulation environments, with consistent outputs observed between MiL and HiL. Minor discrepancies observed in cumulative emissions are attributed to the stochastic cycle-to-cycle variation that results in driving-cycle to driving-cycle variations, and here we simulated a single realisation of the driving cycle.

3.2 CI engine simulations under WLTP driving cycle

The developed toolchain was applied to CI engines to simulate driving-cycle to driving-cycle variability in exhaust emissions, including NO_x, soot, unburned hydrocarbons, and CO from a diesel fuelled engine.

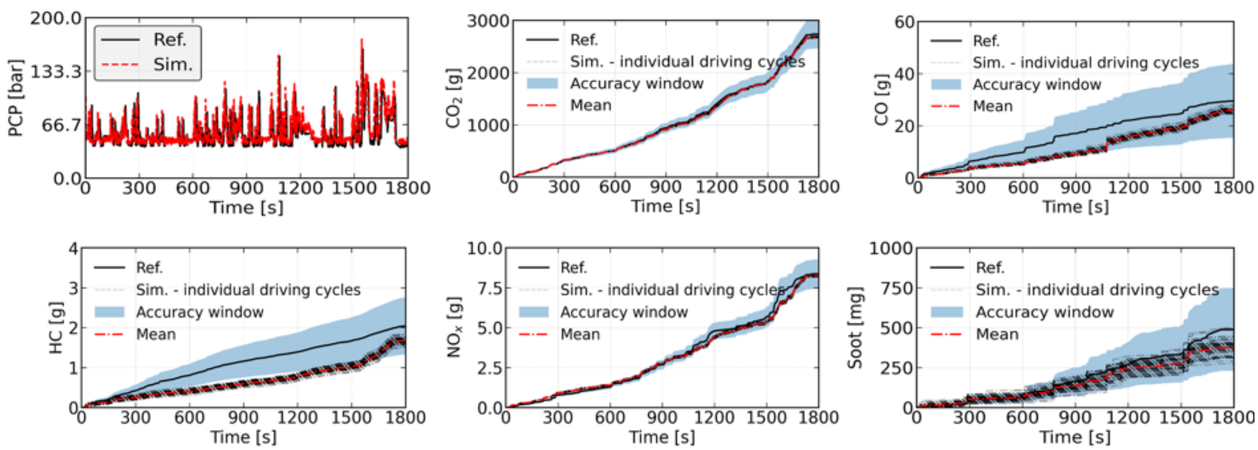

FIGURE 7: Maximum in-cylinder pressure and cumulative histories of exhaust emissions from consecutive fifty driving-cycles and their average history compared with reference data and plotted over the accuracy window of test-bench measurement.

The trained at steady-state conditions model was applied to simulate the transient engine behaviour under the WLTP cycle. Combustion metrics maintain good alignment with reference data. Emissions trends are well captured. CO_2 and CO match closely with experimental counterparts. Similarly, NO_x is also accurately predicted. Some discrepancy between the simulated and reference data is attributed to the effects of activating low-pressure EGR in these phases of the cycle. HC and soot are underestimated during the majority of the cycle, but their end-cumulative values are found within the determined accuracy windows based on the work reported in [23].

To quantify the stochastic variability of the obtained results from transient simulations, 50 consecutive runs of the WLTP cycle were conducted using identical engine data. Results show that a stable mean for CO_2, CO, and NO_x can be achieved with 15 simulations, while HC and soot require 25 and 35 simulations, respectively. Simulated pollutants fall within experimentally defined accuracy windows, confirming the model's reliability for transient emissions prediction with respect to results obtained experimentally. Simulations took 30 minutes for a 30-minute driving cycle, proving the potential to use the method in real-time emissions prediction and virtual calibration workflows.

3.3 Simulation a CNG engine cold start during WLTP and NRTC

The developed toolchain was applied to SI engines to study CNG-fuelled engine behaviour during the cold-start phase of the driving cycles; WLTP for passenger car applications and NRTC for agricultural tractor applications. Investigations examined engine behaviour under varying ambient temperatures, fuel compositions, and air-fuel ratios.

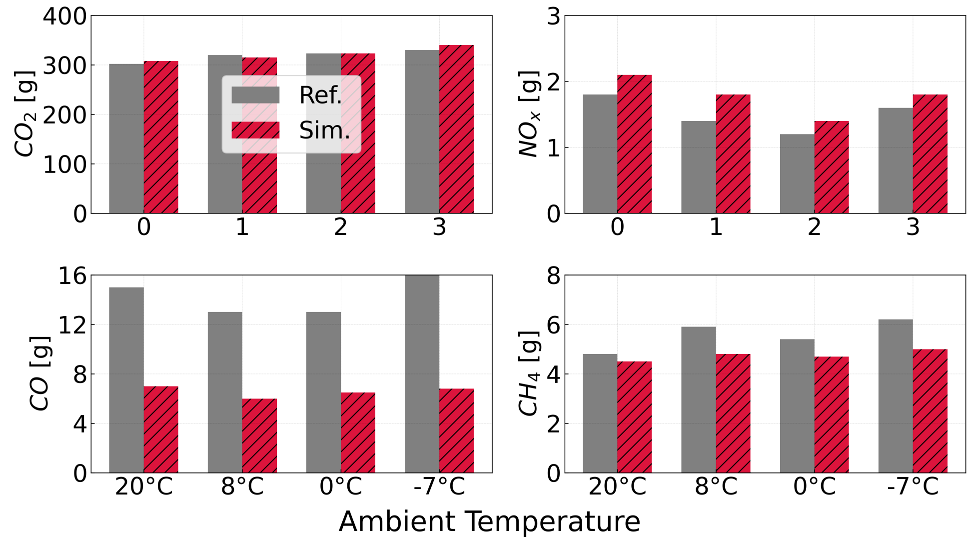

FIGURE 8: Simulated and reference emissions at EVO and for different ambient temperatures during WLTP.

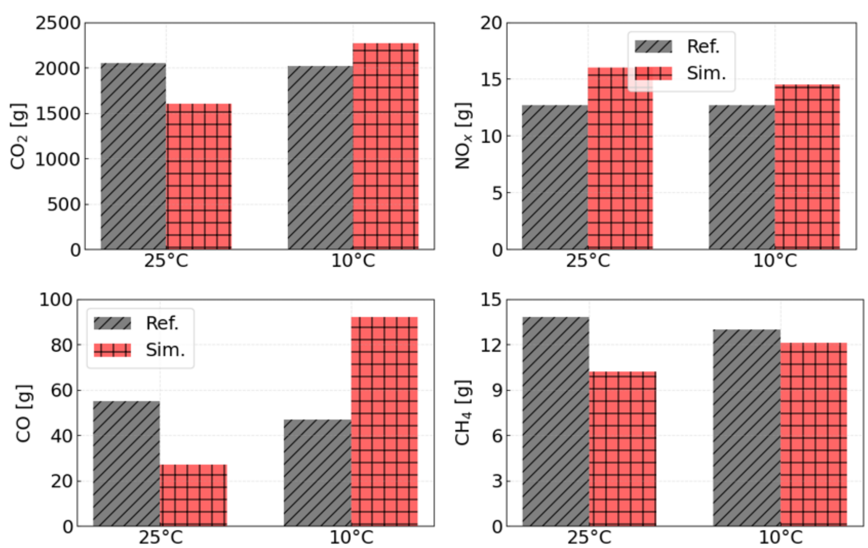

FIGURE 9: Comparison of experiment and simulation cumulative CO_2, CO, NO_x and CH_4 for transient cold start at 25°C, 10°C ambient temperatures during NRTC.

Overall, the model captures the trend of CO_2, NO_x, and CO emissions for different ambient temperatures during WLTP (Figure 8). For CH_4, the model shows a slight increase for decreasing ambient temperatures, but is not able to closely capture the trend of the experimental data. The incorporation of flame-wall quenching could improve the predictability of the engine model at low ambient temperatures. Similar investigations were performed for non-road applications during NRTC (Figure 9). An increase in ambient temperature from 10°C to 25°C resulted in a decrease in total CO_2 produced. Predicted cumulative NO_x is increased as the ambient temperature increases. Cumulative predicted CO is decreased as the ambient temperature decreases. Cumulative predicted CH_4 is decreased with an increase in ambient temperature from 10°C to 25°C, as the temperature increases, consequently, increased fuel consumption, the fuel remaining in exhaust emissions decreases.

3.4 Simulation toolchain with exhaust after treatment systems

The capability of the developed toolchain for co-simulation with aftertreatment systems was verified based on the coupling with an SCR model applied to simulating an ammonia-biodiesel-fuelled engine, resulting in a virtual test bench. Specifically, the work focused on investigating numerically the impact of the properties of SCR catalyst on ammonia reduction under engine-relevant operating conditions. The after-treatment system was modelled using a 1D Catalytic Reactor Model (CRM) representing SCR catalyst [24] and employing the detailed surface reaction kinetics relevant to study SCR. FMI technology was used to compile both the SRM-based engine model and the CRM into FMUs. The FMUs are connected in MATLAB/Simulink. At each simulation time step, the SRM provides the CRM with exhaust gas pressure, temperature, mass flow rate, and species concentrations such as NO, NO_x, NH_3, CO, CO_2, H_2, H_2O, C_3H_6. The CRM returns back-pressure and outlet composition if required for closed-loop operation.

The SRM working within the co-simulation toolchain reproduced in-cylinder pressure with good agreement to experimental data. Similarly, good agreement is observed for exhaust emissions (Figure 10).

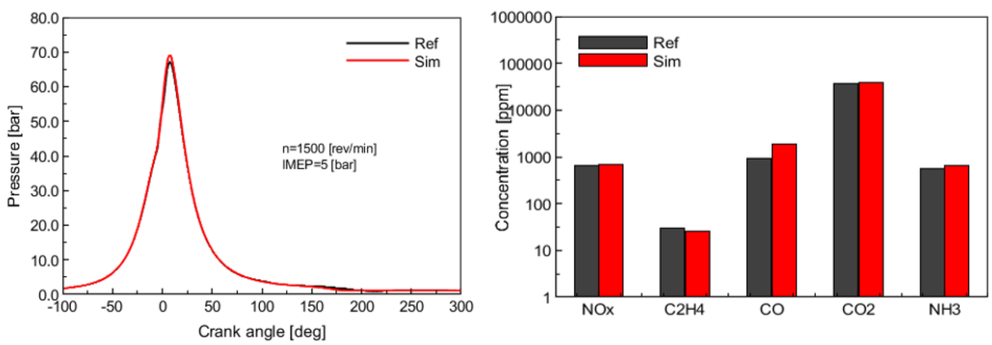

FIGURE 10: Example co-simulation result showing SRM-predicted in-cylinder pressure and emissions and measured data at the selected operating point [14].

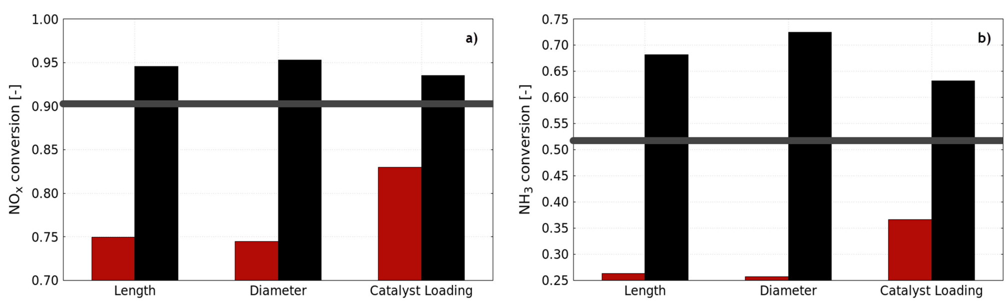

FIGURE 11: Effect of catalyst length, diameter, and loading variation on NO_x and NH_3 conversion [14].

The SCR simulation predicted a baseline NO_x conversion of 90.1% and an NH_3 conversion of 51.6% at the given inlet temperature (550 K) and composition. These values were in line with literature trends for Fe-ZSM-5 catalysts [25]. To illustrate the flexibility of the toolchain, a brief sensitivity analysis was performed by varying catalyst length, diameter and catalyst loading. Increasing the catalyst length by 50% increased NO_x conversion to 94.6%, while reducing the length by 30% reduced conversion significantly. Similar trends were observed for the diameter, increasing by 30% increased NO_x conversion by a few percentage points, while decreasing by 30% reduced conversion significantly. Increasing catalyst loading by 30% improved NO_x conversion by 3.8%

while reducing it by 30% reduced efficiency by 8%. While NO_x reduction was effective, notable ammonia slip was observed. This was attributed partly to competing reactions involving CO and unburned hydrocarbons in the exhaust. In practice, this could be addressed by integrating an ammonia slip catalyst downstream.

3.5 ANN-based simulations of driving cycle

In the final stage of the presented work, a feedforward ANN-based meta-model of the driving cycle simulator, which acts as a surrogate model, has been developed to run beyond real-time and embedded within the toolchain. The performance evaluation showed very good agreement between ANN predictions and SRM outputs. Error metrics, when comparing the ANN-based results with the source SRM-based results, were below 5% across the combustion and emissions metrics together.

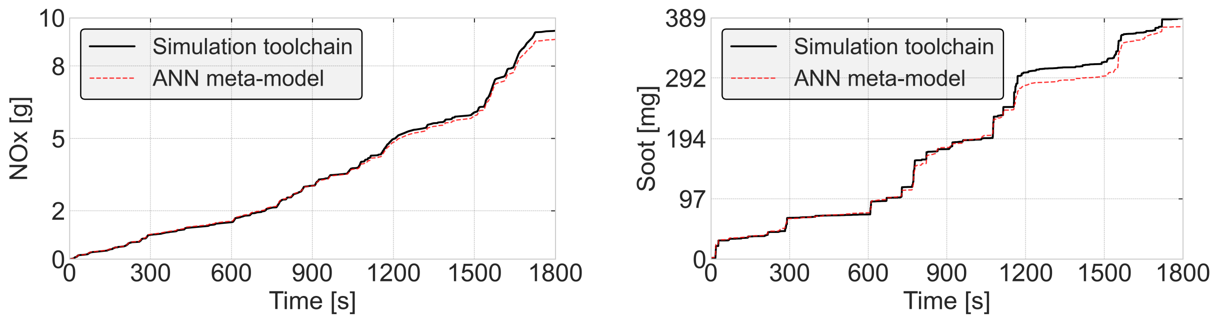

FIGURE 12: Comparison between results obtained with the ANN model and the physical model employing the SRM.

It should be mentioned that in the current workflow, the ANN-based model is trained using the results from the physics-based simulations of the driving cycle that employ the SRM. Hence, the ANN cannot work without the SRM at this stage. The ANN-based model could support simulations targeted at testing other sub-models and their control algorithms if they could be simulated faster than real-time, and not requiring changes in the modelling of the in-cylinder processes. In such circumstances, the ANN, once trained, could eliminate the need to rerun the full simulation toolchain, saving computational resources, as for instance simulating the complete WLTP using the ANN takes below 10s. Nevertheless, because of the aforementioned limits, at this stage, this modelling approach is considered preliminary and requires further development and improvements.

4 Summary

The work introduces a real-time framework for simulating engine performance parameters, exhaust emissions, and fuel effects of SI and CI engines under transient conditions of actual driving cycles, such as WLTP and NRTC. The framework is a unified toolchain that integrates 0D SRM with tabulated chemistry for combustion and pollutant formation into MiL and HiL platforms using the FMI/FMU standard. Such frameworks are of interest to the automotive industry as they help reduce the costs and time involved in costly experimental and prototyping work. The focus has been placed on enabling real-time transient simulations through the seamless integration of the 0D SRM into standardised co-simulation environments, such as MiL/HiL, and cross-platform simulations, features which were not available earlier. This bridges the gap between physical modelling accuracy and practical applicability in engine development workflows for real-time transient simulations.

The existing SRM was adapted to accommodate continuous changes of engine operating parameters such as speed, load, fuel injection strategy, and valve timing. This allowed transient simulations and integration of outer-level software that can support closed-loop control and system-level simulations. An FMI/FMU wrapper compliant with the FMI 2.0 co-simulation standard was developed to interface with legacy Fortran SRM code and platforms such as MATLAB/Simulink, dSPACE ASM, VEOS, and SCALEXIO. The wrapper includes a real-time mode for deterministic execution and synchronised data exchange with external controllers. Port definitions were standardised via an XML file to allow automatic signal mapping and integration. Integration into MiL and HiL environments ensured consistent communication, execution rates, and synchronisation with control systems. The framework was further extended to couple engine processes with exhaust aftertreatment models for co-simulations. Finally, a feedforward ANN-based meta-model, trained on SRM-based driving cycle outputs, was developed, providing ultra-fast surrogate simulations.

The toolchain was applied to CI and SI engines under transient WLTP and NRTC cycles. For CI engines, it predicted driving-cycle-to-cycle variability in NO_x, soot, HC, and CO emissions, showing that multiple runs (15–35 depending on the pollutant) are required for statistical stability and a match to experimental data. For SI engines, the cold-start behaviour of a CNG engine was analysed, revealing strong dependencies of emissions on ambient temperature, fuel composition, and air-fuel ratio. Lower temperatures increased CO_2 and CH_4, while H_2 enrichment enabled compliance with Euro 6 CO_2 limits. Similar trends were observed for a heavy-duty agricultural CNG engine under NRTC cold-start conditions, demonstrating the model's ability to capture temperature-dependent phenomena.

Co-simulation with aftertreatment systems was demonstrated via an SCR model coupled to an ammonia-biodiesel engine, forming a virtual test bench. Sensitivity analysis on catalyst geometry and loading showed their influence on NO_x conversion, while highlighting areas for improvement regarding ammonia slip. The modular framework allows substitution of the SCR with other aftertreatment models, supporting digital twin applications and integrated optimisation of engine aftertreatment systems.

MiL/HiL applicability was verified on a dSPACE platform for a gasoline engine under WLTP. Simulated in-cylinder pressure and emissions agreed across MiL and HiL approaches, with real-time execution maintained without overruns, validating integration accuracy. The ANN-based meta-model reproduced reference results with less than 5% deviation and greatly accelerated simulation time, though it currently relies on prior SRM simulations.

Overall, the work demonstrates a robust, scalable framework for real-time predictive simulations of SI and CI engines under transient conditions. Its FMI/FMU-based integration enables seamless MiL/HiL operation, coupling with aftertreatment models, and potential for digital twin applications, providing a valuable tool and applicable in both research and industrial engine development contexts.

Bibliography

[1] European Union. Worldwide harmonised Light-duty vehicles Test Procedure (WLTP) and Real Driving Emissions (RDE), 2017. EUR-Lex.

[2] TransportPolicy.net. International: Nonroad: Nrtc test. `https://www.transportpolicy.net/standard/international-nonroad-nrtc-test/`, n.d. Accessed July 2025.

[3] Paul D. Walker. Model-based development and calibration of automotive engine control systems. *Control Engineering Practice*, 19(9):962–969, 2011.

[4] Tianjun Zheng, Amir Khajepour, Shuaibing Chen, and Bakhtiar Litkouhi. A model-in-the-loop simulation framework for vehicle control system development. *IEEE Transactions on Vehicular Technology*, 67(5):3885–3899, 2018.

[5] Werner Damm, Bernhard Josko, Jens Oehlerking, and Thomas Peikenkamp. Hardware-in-the-loop and real-time simulation for automotive applications. *ATZ Worldwide*, 110(9):28–33, 2008.

[6] dSPACE. Fmi for pc-based simulation, dspace, 2024.

[7] Tim Franken, Andrea Matrisciano, Rafael Sari, Álvaro Fogué Robles, Javier Monsalve-Serrano, Dario Lopez Pintor, Michal Pasternak, Antonio Garcia, and Fabian Mauss. Modeling of reactivity controlled compression ignition combustion using a stochastic reactor model coupled with detailed chemistry. In *SAE Technical Paper Series*, ICE2021. SAE International, September 2021.

[8] Michal Pasternak, Fabian Mauss, Christian Klauer, and Andrea Matrisciano. Diesel engine performance mapping using a parametrized mixing time model. *International Journal of Engine Research*, 19(2):202–213, July 2017.

[9] Michal Pasternak, Fabian Mauss, Cathleen Perlman, and Harry Lehtiniemi. Aspects of 0d and 3d modeling of soot formation for diesel engines. *Combustion Science and Technology*, 186(10–11):1517–1535, September 2014.

[10] Corinna Netzer, Michal Pasternak, Lars Seidel, Frédéric Ravet, and Fabian Mauss. Computationally efficient prediction of cycle-to-cycle variations in spark-ignition engines. *International Journal of Engine Research*, 21(4):649–663, June 2019.

[11] Andrea Matrisciano, Tim Franken, Laura Catalina Gonzales Mestre, Anders Borg, and Fabian Mauss. Development of a efficient tabulated chemistry solver for internal combustion engine optimization using stochastic reactor models. *Applied Sciences*, 10(24):8979, December 2020.

[12] Alberto Falcone and Alfredo Garro. Distributed co-simulation of complex engineered systems by combining the high level architecture and functional mock-up interface. *Simulation Modelling Practice and Theory*, 8 2019.

[13] Edmund Widl, Florian Judex, Katharina Eder, and Peter Palensky. Fmi-based co-simulation of hybrid closed-loop control system models. In *2015 International Conference on Complex Systems Engineering (ICCSE)*, page 1–6. IEEE, November 2015.

[14] Reddy Babu Siddareddy, Michał Pasternak, Larisa León de Syniawa, Vivien Guenther, Lars Seidel, Fabian Mauss, Grzegorz Przybyła, and Wojciech Adamczyk. Simulations of the scr catalyst in ammonia-biodiesel fuelled ci engine using virtual test bench with detailed chemistry. *Renewable Energy*, 251:123169, October 2025.

[15] Reddy Babu Siddareddy, Tim Franken, Michal Pasternak, Larisa Leon de Syniawa, Johannes Oder, Hermann Rottengruber, and Fabian Mauss. Real-time simulation of cng engine and after-treatment system cold start part 1: Transient engine-out emission prediction using a stochastic reactor model. *SAE International Journal of Advances and Current Practices in Mobility*, 5(6):2421–2443, April 2023.

[16] François Chollet. Keras. https://keras.io, 2015.

[17] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, et al. Tensorflow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/, 2015. Software available from tensorflow.org.

[18] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, and Olivier Grisel. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[20] Reddy Babu Siddareddy, Tim Franken, Larisa Leon de Syniawa, Michal Pasternak, Sascha Prehn, Bert Buchholz, and Fabian Mauss. Simulation of cng engine in agriculture vehicles. part 1: Prediction of cold start engine-out emissions using tabulated chemistry and stochastic reactor model. In *SAE Technical Paper Series*, ICE2023. SAE International, August 2023.

[21] Mario Picerno, Sung-Yong Lee, Michal Pasternak, Reddy Siddareddy, Tim Franken, Fabian Mauss, and Jakob Andert. Real-time emission prediction with detailed chemistry under transient conditions for hardware-in-the-loop simulations. *Energies*, 15:261, 12 2022.

[22] LOGE AB. LOGEsoft, LOGEengine, LOGEcat. http://www.logesoft.com, 2024.

[23] Mario Picerno, Sung-Yong Lee, Joschka Schaub, Markus Ehrly, Federico Millo, Mauro Scassa, and Jakob Andert. Co-simulation of multi-domain engine and its integrated control for transient driving cycles. *IFAC PapersOnLine*, 53-2:13982–13987, 12 2020.

[24] Larisa Leon de Syniawa, Reddy Babu Siddareddy, Johannes Oder, Tim Franken, Vivien Guenther, Hermann Rottengruber, and Fabian Mauss. Real-time simulation of cng engine and after-treatment system cold start. part 2: Tail-pipe emissions prediction using a detailed chemistry based moc model. *SAE Technical Paper Series*, 1, 2023.

[25] Larisa Leon de Syniawa, Reddy Babu Siddareddy, Johannes Oder, Tim Franken, Vivien Guenther, Hermann Rottengruber, and Fabian Mauss. Real-time simulation of cng engine and after-treatment system cold start part 2: Tail-pipe emissions prediction using a detailed chemistry based moc model, 4 2023.