Development of a HIL Platform for Dynamic Systems Simulation: Application for Model Simulation and PI Controller Design for Idle Speed Control

Marcus Vinicius de Lima Rebolo
Escola Politécnica de Universidade de São Paulo (EPUSP) - Sistemas Eletrônicos, Brasil

Flavio Almeida de Magalhães Cipparrone, Armando Antônio Maria Laganá, Wellington Henrique Nogueira
Escola Politécnica de Universidade de São Paulo (EPUSP) - Sistemas Eletrônicos, Brasil

Alex Sandro de Souza Jbelle, William Miranda Gusmão
Faculdade de Tecnologia de Sorocaba (FATEC SO) – Tecnologia em Eletrônica Automotiva, Brasil

ABSTRACT

This work presents the development of a Hardware-in-the-Loop (HIL) platform for automotive dynamic system simulations, focusing on the idle speed control of a Volkswagen EA111 BPA 1.6L flex-fuel engine. The HIL generates the necessary sensor signals and reads actuation commands, while a tricore-based ECU (Electronic Control Unit), connected to the HIL, runs the complete engine management routines, including a PI controller (Proportional-Integral) for idle speed and ignition/injection synchronization. Thus, we demonstrate the importance of the HIL platform for automotive software development and testing, enabling the validation of algorithms and control strategies in a controlled environment. The idle speed model was obtained by applying step perturbations to the ignition angle, isolating the influences of air and fuel, and was subsequently embedded into the HIL to replicate the engine's actual dynamics. Finally, the controller's performance was evaluated both on the HIL and the real vehicle, with performance metrics such as the Integral of Absolute Error (IAE) and the Normalized Root Mean Square Error (NRMSE) used to compare the results. The strong agreement between both responses reinforces the effectiveness and reliability of the proposed method.

INTRODUCTION

Idle speed control is a critical function for fuel economy, emissions, and overall drivability. Keeping the speed as low as feasible while maintaining stability minimizes fuel consumption [1]. The challenge is that the load on the crankshaft changes constantly: the air conditioning compressor switches on, the alternator demands extra torque, or the driver turns on lights and fans. Each additional load/disturbance consumes the engine torque and tends to reduce speed, and the idle speed controller must react quickly to keep the target speed with minimal effect on driver perception.

Two actuator channels dominate idle control, throttle position and ignition timing. The throttle regulates the inducted air mass, whereas spark advance changes combustion phasing. For fast disturbances the controller primarily adjusts ignition advance, while the throttle provides a slower trim [2]. Engine temperature is also a significant factor. When the engine is cold, it requires a higher idle speed target due to more viscous lubricant, internal parts that have not fully expanded yet and the need to quickly heat the catalytic converter. Once the engine and catalyst reach operating temperature, the target speed can be lowered to save fuel [3].

Developing and calibrating such control loops directly on a vehicle is costly, time-consuming, and sometimes risky. HIL techniques have therefore become mainstream in the automotive industry [4]. In an HIL setup, the ECU software runs in real time, but the engine's sensors and actuators are replaced by electronically generated signals from a simulator. Engineers can test faults, extreme conditions, and new strategies without mechanical risk and at a fraction of the cost of full physical prototypes.

This paper leverages that approach by modeling the idle dynamics of a Volkswagen EA111 1.6 L flex fuel engine, designing a PI idle speed controller, and validating its performance both on a dedicated HIL bench and on the actual vehicle.

HIL SETUP

The custom HIL platform replicates in real time the main electrical signals of a Volkswagen EA111 BPA 1.6 L engine.

This hardware is built around an STM32 microcontroller running C firmware that emulates the engine's electrical signals:

- Static sensors: coolant temperature, accelerator pedal position, pressure, among other analog inputs. They are set via potentiometers.

- Dynamic sensors: crankshaft and camshaft (60-2 tone wheel pattern). They are produced in real time by firmware.

Real actuators (fuel injectors, ignition coils, electronic throttle) can be plugged in, so the ECU sees the correct load impedance. Some ECUs will not operate without it. Figure 1 shows the full HIL setup.

Figure 1. The full HIL setup (with actuators).

The system inputs and outputs can be checked in Table 1. It is important to point out that the outputs are simulated by the HIL and the inputs are controlled by the ECU.

Table 1. Inputs and outputs of the HIL developed in the study

Inputs	Outputs
Lambda heater	Line 15 (keyed power)
Throttle PWM	Pedal sensor (2 total)
Ignition signals (2 total)	TPS signal (Throttle Position Sensor) – (2 total)
Injection signals (4 total)	Pressure sensor
Fuel pump relay	Intake air temperature sensor
Cold start fuel pump relay	Engine coolant temperature sensor
Canister purge relay	Knock sensor
CAN communication	Clutch pedal
K-Line communication	Brake pedal (2 total)
UART communication	Alternator
	Engine speed signal (crankshaft signal)
	Camshaft position signal
	Battery 12V

Figure 2 labels the custom controller board, where:

1. Power connector.

2. STM32 MCU.

3. Potentiometers (static sensors).

4. Converters, drivers and relays.

5. Headers to ECU harness.

6. Load outputs (coils, injectors and throttle body).

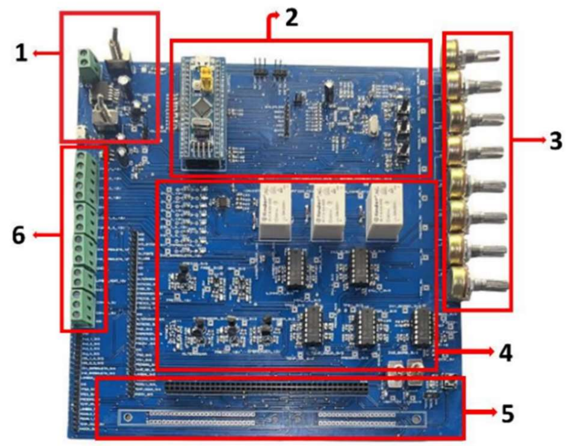

Figure 2. Main board of the HIL.

A graphical interface was developed to mediate communication between the user and the system, making it possible to define simulation values and view all actuation and simulation data in real time. This graphical interface was created using the Python programming language, thus it is possible to manipulate the rotation value and some actuators such as the brake and the pedal. It is also possible to view the values of the statically simulated sensors. In Figure 3 you can see the graphical interface used in the system.

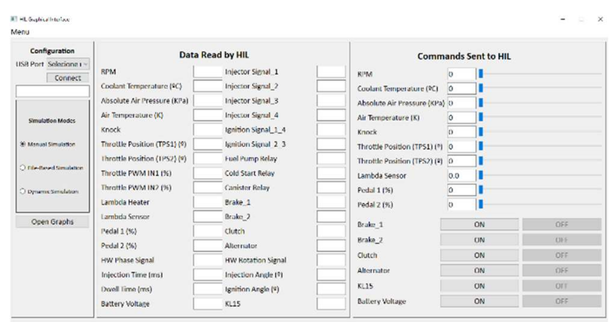

Figure 3. HIL Graphical Interface.

In the interface, three excitation modes are available: Manual, File-Based, and Dynamic. In manual mode the target speed is entered directly on the screen while the remaining analog sensors (pedal, pressure, temperature) are set with the hardware potentiometers. In file-based simulation mode the user uploads a previously recorded log, the GUI streams the file, and the firmware reproduces every signal in the exact time sequence captured on the real vehicle. In Dynamic mode the firmware runs embedded plant models. For example, the idle-speed model discussed later, so that realistic transients are generated mathematically as a function of the commands issued from the GUI.

Figure 4 shows that the HIL has replicated a cold start and idle speed from a real EA111, with the simulated crank trace matching the physical engine point for point [5].

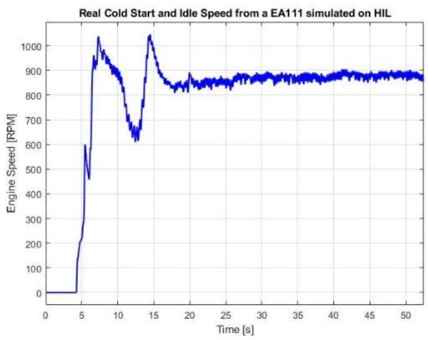

Figure 4. RPM recorded in the real vehicle and simulated in the HIL.

To verify the correct synchronization behavior of the ECU, a logic analyzer was used to capture the real-time waveforms of the crankshaft and camshaft signals generated by the HIL, as well as the resulting ignition and injection commands controlled by the ECU.

As shown in Figure 5, the ECU correctly interprets the 60-2 crankshaft pattern and camshaft signal to execute sequential fuel injection and spark ignition events. This validates that the HIL system generates accurate signal timing and that the ECU responds accordingly, completing the closed loop in real time.

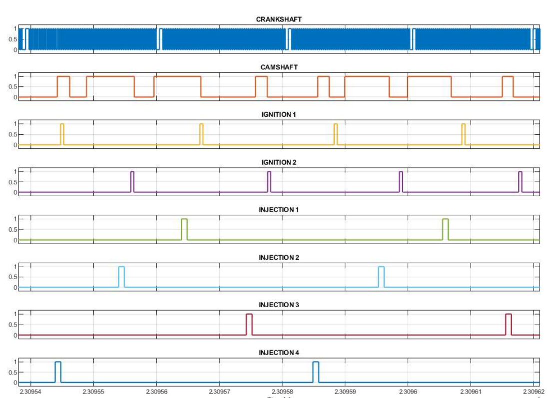

Figure 5. Acquisition of crankshaft, camshaft, ignition and injection waveforms during ECU synchronization using the HIL.

ECU SETUP

Figure 6 shows the ECU used in the study is based on an academic board described in [6], originally fitted with four PIC18F46K22 MCUs.

Figure 6. ECU fitted with four PIC MCUs

To meet current automotive requirements, the PICs were replaced by an Infineon AURIX TC297B Tricore evaluation kit. A dedicated "mezzanine" board links the kit to the original baseboard, preserving all injector, coil, throttle, and relay drivers (Figure 7).

Figure 7. ECU fitted with AURIX TC297B evaluation kit upgrade.

The firmware was structured with a Basic Software (BSW) layer in C for peripheral configuration, timing, and fuel/spark generation (synchronization). The Application Software (ASW) layer contains modules for high-level control (idle, torque, diagnostics), developed using Model-Based Design (MBD) in Matlab/Simulink.

COMPLETED SETUP

The final setup connecting the ECU to the real vehicle is shown in Figure 8, while the setup with the HIL system is illustrated in Figure 9.

Figure 8. Final setup connecting the ECU to the real vehicle.

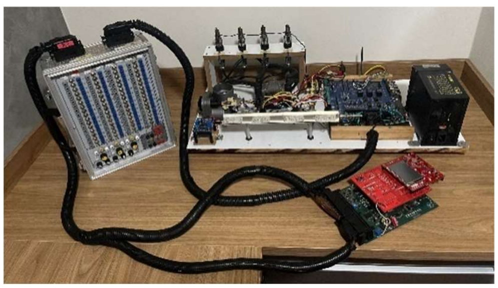

Figure 9. Final setup connecting the ECU to the HIL.

IDLE SPEED PLANT IDENTIFICATION

The mathematical model was identified using the Process Reaction Curve (PRC) method, as proposed by [3]. The only actuator excited during the experiment was the ignition advance, which was stepped from –5° to +21°. To ensure that the torque variation came exclusively from the ignition advance, all other influencing variables were kept constant. The throttle was fixed at 5% opened, engine coolant temperature stabilized at 90°C and the fueling system was calibrated to maintain λ = 1 (stoichiometric mixture) throughout the entire speed range.

Maintaining λ = 1 is essential because, in spark-ignition engines, the air-fuel ratio has a strong influence on torque production. Rich mixtures (λ < 1) increase flame speed and torque, while lean mixtures (λ > 1) reduce combustion efficiency and stability [7]. Any variation in λ or airflow would mask the isolated effect of ignition advance.

Therefore, with the air path and fueling locked, the change in engine speed can be attributed exclusively to ignition advance timing.

The step response was recorded and is shown in Figure 10, where the ignition advance (input) and engine speed (output) are plotted. The engine speed increased from 825 rpm to 1220 rpm following the step input of 26°.

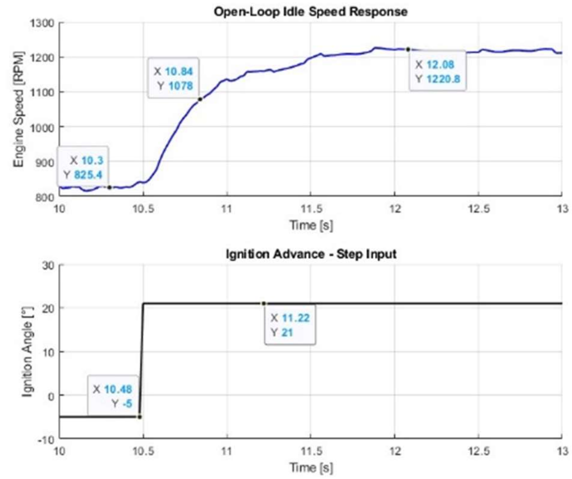

Figure 10. Open-loop idle speed response

Using the classical PRC method:

$$G(s) = \frac{K}{\tau s + 1} \tag{1}$$

The steady-state gain is calculated as:

$$\mathrm{K} = \frac{(1220 - 825)}{(21 - (-5))} = 15.19 \tag{2}$$

The time constant (τ) was obtained by measuring the time to reach 63.2% of the final value:

$$\tau = 10.83 - 10.48 = 0.35 \tag{3}$$

Thus, the first-order transfer function becomes:

$$G(s) = \frac{15.2}{0.35s + 1} \tag{4}$$

To improve the model fit to the measured response, K was adjusted to 15 and τ to 0.45s (Equation 5), resulting in a better curve match (around 86% fit, as evaluated using the “compare()” command on Matlab), as shown in Figure 11.

$$G(s) = \frac{15.}{0.45s + 1} \tag{5}$$

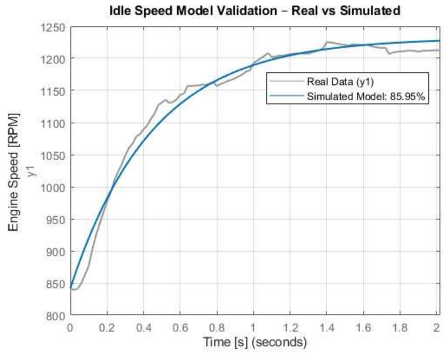

Figure 11. FIT Comparison: Idle Speed Model vs Real Idle Speed Data.

To implement the model in the HIL environment, the continuous transfer function was discretized with the Tustin method using a 20 ms sampling period, reaching the Equation 6:

$$G(z) = \frac{0.3261z + 0.3261}{z - 0.9565} \quad (6)$$

IDLE SPEED CONTROLLER DESIGN

The idle speed controller was designed in the frequency domain using a Proportional-Integral (PI) structure. The performance goals defined for the synthesis focused on achieving a fast rise time and low overshoot, while respecting the physical limitations of the system. The main targets considered were:

- Rise time: 800 milliseconds

- Damping ratio $\zeta = 0.9$

From these specifications, the desired closed-loop poles were calculated and placed in the complex s-plane as:

$$s_{1,2} = -\zeta\omega_n \pm j\omega_d = -6.944 \pm j\,3.363 \quad (7)$$

The PI controller was designed using classical frequency-domain techniques. The zero introduced by the compensator was placed to meet the required phase contribution at the desired closed-loop pole, while the gain was computed using the magnitude condition [8]. This design procedure is illustrated below.

$$\angle C(s_0)G(s_0) = -180° \quad (8)$$

$$\angle C(s_0) = -35.46° \quad (9)$$

$$\angle G(s_0) = -144.54° \quad (10)$$

Assuming a generic PI compensator:

$$C(s) = K\frac{s + a}{s} \quad (11)$$

The zero at $-a$ contributes an angle

$$\varphi = \tan^{-1}\left(\frac{\omega_d}{-\zeta\omega_n}\right) \quad (12)$$

Imposing $\varphi = -35.46°$, leads to:

$$a = 2.21\,rad/s \quad (13)$$

With the angle requirement fulfilled, the magnitude condition was applied to compute the controller gain, resulting in:

$$K = \frac{1}{|C(s_0)G(s_0)|} = 0.231 \quad (14)$$

Thus, the controller in continuous-time form is described by the following transfer function:

$$C(s) = 0.231\frac{(s + 2.21)}{s} \quad (15)$$

With controller gains set to Kp = 0.231 and Ki = 0.511, the step response of the compensated system is illustrated in Figure 12, based on the closed-loop simulation implemented in Simulink, as depicted in Figure 13. The system was tested with an initial engine speed of 825 RPM, and a reference step to a 1200 RPM setpoint was applied at 1s. These results demonstrate the dynamic behavior of the idle speed control under a step change in reference input.

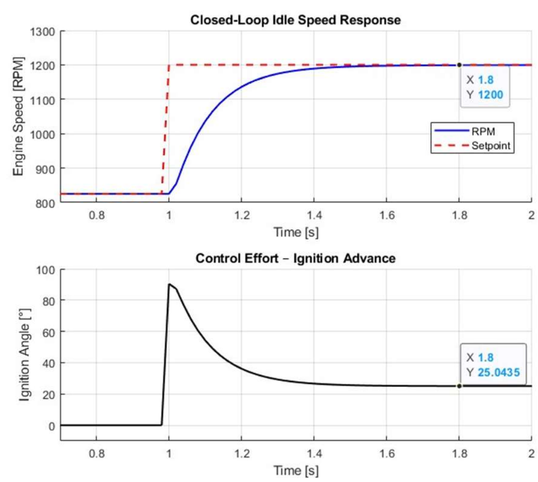

Figure 12. Closed-loop idle speed response.

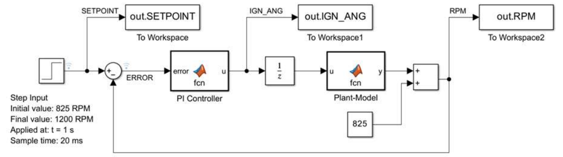

Figure 13. Simulink diagram of the idle speed control loop.

RESULTS

To validate the proposed control strategy and the fidelity of the HIL platform, both systems were tested under identical conditions: an idle speed reference step from 900 RPM to 1200 RPM. The real vehicle was configured to match the same operating conditions used during model identification, including fixed throttle position, engine coolant temperature stabilized at 90°C and stoichiometric air-fuel ratio. The closed-loop response of the real vehicle and the HIL simulation are compared in Figure 14. The results show close agreement in rising time, steady-state value, and overall transient behavior. It is important to point out that the ignition advance was saturated between –10° and +35°, reflecting physical limitations of the engine and ignition system.

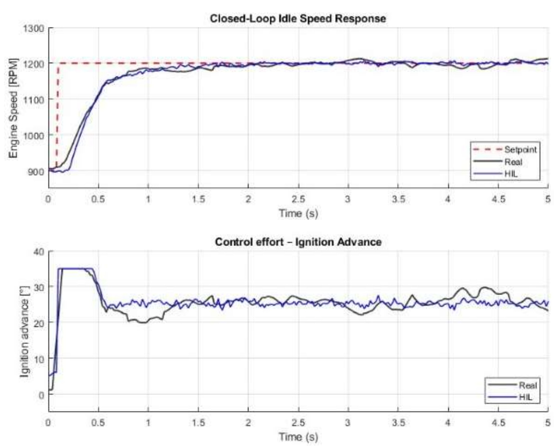

Figure 14. Comparison of real vehicle and HIL response

For a quantitative assessment, the filtered signals were analyzed over a 5-second observation window. The Integral of Absolute Error (IAE) between the real and HIL engine speed responses was 37.20 RPMs, while the IAE for the ignition advance control effort was 8.76°. The IAE metric quantifies the cumulative absolute deviation between two signals throughout the observation period. While specific percentage-based thresholds for IAE are less common in the literature due to its dependence on the signal's magnitude, units, and observation duration, lower IAE values consistently indicate better alignment between the compared responses [9]. In the context of the engine's operational range and transient dynamics, these IAE values reflect close tracking performance.

In addition, the Normalized Root Mean Square Error (NRMSE) for the engine speed response was 3.41% and for the ignition advance control effort was 7.19%, demonstrating a high level of correlation between the simulated and real systems. Performance thresholds for NRMSE vary depending on the application, in empirical modeling literature, values below 10% are often considered excellent [10].

These results confirm that the HIL platform accurately reproduces the closed-loop dynamic behavior of the engine, supporting the validation and calibration of idle speed control strategies before applying them to the physical vehicle.

To further assess controller robustness, a load disturbance was introduced by triggering the radiator fan while the vehicle was in idle mode. Although the disturbance magnitude was not measured directly, its effect on engine speed is clearly visible. The activation of the fan imposed an additional torque load on the engine, leading to a momentary drop in RPM. As shown in Figure 15, the controller quickly compensated for this disturbance through ignition advance adjustments, restoring the idle speed to the target value. This response demonstrates the system's ability to reject real-world disturbances and maintain idle stability.

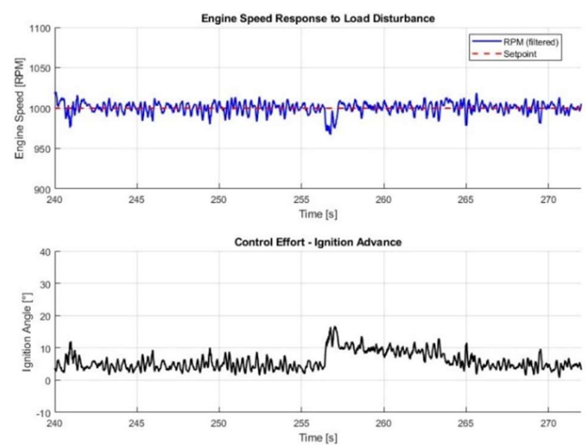

Figure 15. Idle speed disturbance rejection during fan activation.

Additionally, a sequence of idle speed reference steps with different amplitudes was applied to the real system. The responses, illustrated in Figure 16, demonstrate consistent performance across varying conditions. The controller maintained stable behavior and low overshoot in all cases. These tests reinforce the reliability and versatility of the proposed control strategy and the HIL platform for development and validation purposes.

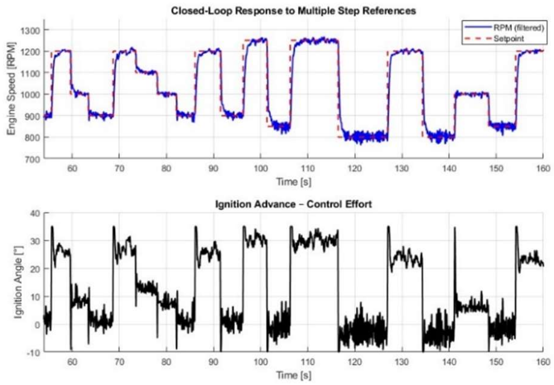

Figure 16. Closed-loop response to multiple idle speed reference steps in the real plant.

CONCLUSION

This work presented the development and validation of a HIL platform designed to simulate the idle speed behavior of an internal combustion engine and support the implementation of advanced control strategies. A mathematical model of the idle speed dynamics was identified and embedded into the HIL system, allowing real-time interaction with a tricore-based ECU executing a PI controller.

The control strategy was successfully validated both in the HIL and in the actual vehicle, with close agreement observed between the two environments. Quantitative metrics, including IAE and NRMSE, confirmed the accuracy and effectiveness of the proposed control solution. Additionally, the system demonstrated robustness to real-world disturbances such as the activation of the radiator fan.

The results reinforce the capability of the proposed HIL platform as a reliable and efficient tool for the development, testing, and validation of engine control algorithms prior to in-vehicle application, reducing development time and risk while increasing confidence in software performance.

FUTURE WORK

The HIL platform developed in this study provides a robust foundation for a range of future research efforts. Its architecture is inherently scalable, opening possibilities for integrating models of other critical automotive subsystems, such as cold start, throttle body, or lambda sensor. This would allow for more comprehensive vehicle-level simulations and the development and testing of integrated control strategies. Furthermore, the platform can serve as a valuable tool to advance the study of modeling and simulation techniques for internal combustion engines, enabling deeper investigations into complex phenomena and the validation of higher fidelity engine models under diverse operating conditions.

Specific improvements to the current idle speed control application are also envisioned. The crankshaft position signal, essential for RPM calculation, exhibited some noise during experimental validation. Therefore, a promising avenue for future work involves the implementation of a digital filter within the ECU firmware for the raw RPM signal. By improving the quality of the speed measurement, it is anticipated that the performance of the idle speed controller could be further enhanced, potentially leading to even lower IAE and NRMSE values and a more robust response to disturbances. This refinement would also contribute to a more accurate validation of subsequent control algorithms developed on the HIL platform.

REFERENCES

[1] KIENCKE, U.; NIELSEN, L. **Automotive Control Systems:** For Engine, Driveline and Vehicle (2nd ed.). Springer-Verlag, Berlin, Heidelberg, 2005. ISBN 3-540-23139-0.

[2] ASHOK, B.; ASHOK, S. D.; KUMAR, C. R. **A review on control system architecture of a SI engine management system.** Annual Reviews in Control, Vellore, India, 2016. https://doi.org/10.1016/J.ARCONTROL.2016.04.005

[3] MANIVANNAN, P.V., SINGAPERUMAL, M. & RAMESH, A. **Development of an idle speed engine model using in-cylinder pressure data and an idle speed controller for a small capacity port fuel injected SI engine**. Int.J Automot. Technol. 2011. https://doi.org/10.1007/s12239-011-0002-3.

[4] ZANDER, J.; SCHIEFERDECKER, I. K.; MOSTERMAN, P. **Model-based testing for embedded systems** (1st ed.). CRC Press, 2011. ISBN 978-14-3981-845-9

[5] NOGUEIRA, W.H. **Gerenciamento de motor à combustão**: projeto de controle PID com compensador feedforward para válvula borboleta. Dissertação (Mestrado em Engenharia de Sistemas) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2025. doi:10.11606/D.3.2025.tde-24042025-073259.

[6] JUNIOR, E. I. A.; JATO, F.; HIROKI, G. B. **Desenvolvimento de uma unidade de gerenciamento eletrônico para motor volkswagen 1.6l**. Trabalho de Conclusão de Curso (Graduação). Faculdade de Tecnologia de Santo André, Santo André, p. 127, 2016.

[7] HEYWOOD, J. B. **Internal Combustion Engine Fundamentals** (2nd. ed.). McGraw-Hill Companies, 2018. ISBN 978-12-6011-610-6.

[8] OGATA, K. **Engenharia de controle moderno** (5th. ed.). São Paulo. Pearson Prentice Hall, 2010. ISBN 978-85-4301-375-6.

[9] MOUSAKAZEMI, S. M. H. **Comparison of the error-integral performance indexes in a GA-tuned PID controlling system of a PWR-type nuclear reactor point-kinetics model.** Progress in Nuclear Energy, vol. 132, p. 103604, 2021, doi: 10.1016/j.pnucene.2020.103604.

[10] CHERIFI, L.; AMMI, Y.; HANINI, S.; HENTABLI, M.; BELKACEM, O.; HARMAND, J. **A Novel Empirical Fractional Approach for Modeling the Clogging of Membrane Filtration During Protein Microfiltration**. Membranes, vol 15, 2025. https://doi.org/10.3390/membranes15040099.