

A Systematic Review of Hardware-in-The-Loop (HiL) Frameworks for Internet of Things Applications

Patnaikuni Dinkar R Patnaik¹ and Dr. Sachin R Gengaje²

^{1,2}Department of Electronics Engineering, Walchand Institute of Technology, Solapur, India

Email: pdrpatnaik@gmail.com, srgengaje@witsolapur.org

ORCID: ¹0000-0002-6737-8273, ²0000-0001-7981-7395

Abstract— Hardware-in-the-Loop (HiL) testing has emerged as a pivotal technique for validating and enhancing the performance of control systems across various industries, notably in automotive and embedded manufacturing sectors. This paper provides an extensive exploration of HiL testing methodologies and their application in the Internet of Things (IoT) domain. HiL testing involves simulating real-world conditions by integrating specialized simulation software and hardware components, facilitating realistic evaluation of controllers' functionality and behaviour. Different variants of HiL, such as Model-in-the-Loop (MiL), Software-in-the-Loop (SiL), Processor-in-the-Loop (PiL), and Hardware-in-the-Loop (HiL), are discussed within the broader context of XiL (Anything-in-the-Loop) models. The paper underscores the significance of HiL testing in optimizing the development process, reducing costs, and minimizing hazards. It elaborates on the utilization of HiL testing to validate intricate control algorithms, simulate fault scenarios, and assess system behaviour under diverse conditions. The integration of HiL testing with IoT applications is examined, highlighting the potential benefits in areas like smart cities, industrial IoT, healthcare, agriculture, and energy management. The study presents a novel fault injection framework within HiL testing, allowing real-time analysis of complex systems' responses to simulated faults. This framework aids in identifying vulnerabilities and evaluating system reliability, thereby enhancing safety and performance. The authors introduce the concept of sensor schema, Observations & Measurements (O&M), and Sensor Web Enablement (SWE) as key components for efficient sensor data integration, sharing, and interoperability. The paper concludes by emphasizing the indispensable role of Hardware-in-the-Loop testing in fostering innovation, safety, and efficiency in the IoT landscape. By offering a comprehensive overview, analysis, and practical insights into the application of HiL testing methodologies, this research contributes to the advancement of robust and reliable IoT systems.

Index Terms— Hardware-in-the-Loop (HiL) testing, Internet of Things (IoT), Model-in-the-Loop (MiL), Software-in-the-Loop (SiL), Processor-in-the-Loop (PiL), Fault injection, Sensor schema, Observations & Measurements (O&M), Sensor Web Enablement (SWE), Standards Development Organization (SDO), Control systems, IoT applications, Simulation, Testing methodologies.

I. INTRODUCTION

Hardware-in-the-Loop (HiL) is a simulation method that is prominent in many industries including automotive and embedded manufacturing sectors [1]. There has been development of different formats of these simulation techniques and are tailored into the form which suits the vendor and the application at its best. These variants of

simulation techniques are referred to as X-in-the-loop models where X stands for anything. The modified formats are often referred to as HiL, PiL, MiL, & SiL and often referred collectively as XiL. MiL stands for Model-in-the-Loop which involves a plant model and a controller is where the engine behavior is modelled using a simulation tool and the controller holds the software model of the control signals and they both when put together becomes an MiL system which is used for testing and validation. Software-in-the-Loop consists of engine model using simulation software and the control signals are also simulated using appropriate simulation tool and then store it in terms of any programming language such as C [1-2] and the entire plant is simulation results are all tested and debugged using software only thereby forming SiL. Similarly, Processor-in-the-Loop has the engine model designed through a simulation tool and the control signals are modelled for the controller and further the hex file of the engine model is burned into the controller and replace the existing engine model thereby forming a PiL. Hardware-in-the-Loop replaces by programming the model file into controller and ultimately replacing the real hardware which is responsible for the proper functioning of the hardware thereby forming the HiL.

A. Hardware-in-the-Loop (HiL) frameworks

Hardware-in-the-Loop (HiL) testing is an advanced testing technique commonly used in the field of control systems and engineering. In this method, actual signals from a controller or embedded system are fed into a specialized test system, often implemented with simulation software and hardware components [3]. This test system is designed to replicate the real-world environment in which the controller is intended to operate.

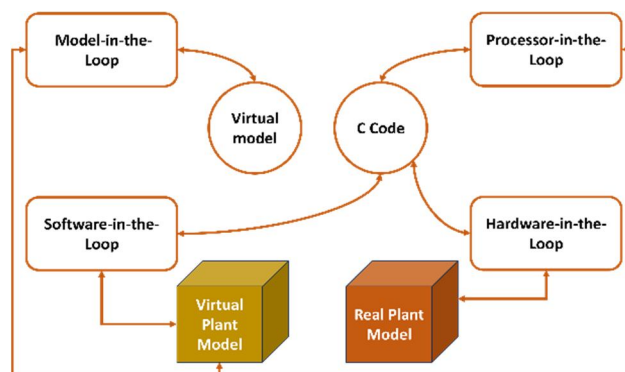


Figure 1: Representation of MiL, SiL, PiL and HiL simulations

The primary objective of HiL testing is to assess and validate the functionality and performance of the controller under realistic conditions, as if it were physically integrated into the final product or system. By connecting the controller to the HiL test system, engineers can subject it to a wide range of simulated scenarios that mirror the operating environment it is designed for. During HiL testing, the controller interacts with the virtual environment created by the test system, receiving and processing simulated sensor inputs and producing appropriate control signals in response. This interaction is almost indistinguishable from how the controller would operate when connected to the actual physical system. One significant advantage of HiL testing is the ability to efficiently perform comprehensive testing without the need for physical prototypes or expensive real-world setups. Instead of conducting time-consuming and costly physical tests, engineers can easily execute thousands of simulations, covering various operational conditions and extreme scenarios. There are numerous vendors who have incorporated these techniques in their industrial processes as a part of simulation, synthesis and for the identification of bugs before launching a fool proof application for a given problem. The main significance of using HiL during development of any hardware system greatly contributes to the industry in terms of making the entire development process more economic, feasible and hazard free [4] where otherwise could be very hazardous to the industry and the involved personnel.

As depicted in Fig.1 to establish a sensor mathematically, firstly the input to output relationship is studied where the input defines the stimuli, and the output defines the response. The underlying manufacturing physics of a sensor largely contributes to the formulation of equations or the mathematical functions representing the sensor relationship for its input and output. Therefore, there is a need to translate the sensor stimuli response relationship into a mathematical model so that it can be framed into a framework or stored in a digital form for a quick reuse and eliminate the dependency of real sensor and most importantly the sensor environment.

The MIL testing phase starts software verification by evaluating standalone or integrated model modules using platforms such as ASCET or MATLAB Simulink. It ensures the harmony of controller-plant interaction, validating

desired control and functionality alignment. This step regulates the controller logic's performance sufficiency [5-7]. After model verification, SIL generates code from the controller model, replacing the controller block. SIL validates embedded software, algorithms, and control loops on a computer platform with plant and controller software models. If SIL detects anomalies, it may prompt iterative adjustments between MIL and SIL, reconfirming MIL's success. Successful MIL and SIL verification led to the subsequent PIL step. PIL, also known as FIL (FPGA-in-the-loop), encompasses executing closed-loop simulations of embedded software, control loops, and system components on real hardware to ensure seamless operation. Any challenges encountered may point to processor capability or necessitate software adjustments, potentially leading back to SIL or MIL phases. Concurrently with control software validation, hardware systems undergo modeling and validation, culminating in HIL testing as the ultimate verification step prior to system integration. Before connecting the embedded processor to physical hardware, HIL testing is performed using platforms such as Typhoon, OPAL-RT, and dSPACE to run simulated plant models in real-time setups. These simulation models are generated by tools like ASCET or MATLAB Simulink. HIL testing encompasses deterministic simulations, establishing actual connections with the processor, and real-time interaction between control software and hardware. It is vital for identifying hidden glitches and is especially critical for safety-critical functions in fields like automotive and aerospace. Integration challenges may necessitate revisiting HIL testing, and in some cases, even earlier stages like MIL, SIL, or PIL testing.

B. Overview of the growing importance of IoT applications

The world of IoT is always changing, and as technology progresses, new ways to use it keep popping up. As IoT becomes more widespread in different areas, it becomes even more important and changes the way we live, work, and interact with our surroundings [8]. It connects things in ways we've never seen before, making our lives better and more efficient.

Also, the pervasive adoption of IoT has deeply permeated today's world, opening numerous opportunities to address challenges and unlock innovative solutions in various domains. IoT's integration has proven beneficial in leveraging data-driven insights, enabling the development of smart cities, optimizing industrial processes through Industrial IoT, revolutionizing healthcare practices, transforming agriculture with precision farming, enhancing energy management, and harnessing the potential of 5G and edge computing. As the technology progresses, it becomes imperative for IoT to continually evolve and proactively embrace emerging trends to remain at the forefront of cutting-edge advancements. Embracing these changes will ensure that IoT continues to stay relevant and impactful in the ever-changing technological landscape.

C. Significance of event based HiL frameworks

HIL testing is commonly used to test the performance and functionality of control systems, such as power electronics, under realistic conditions. It can also be used to test the performance of safety-critical systems, such as those used in the aerospace and automotive industries to ensure their reliability and safety. HIL allows testing of the entire control system, including sensors, actuators, controllers, and communication interfaces in a closed-loop configuration. HIL is a powerful tool for the development and testing of control systems [], allowing for early detection of problems, and reducing the need for expensive and time-consuming physical testing. It can also be used to test and validate the performance of control algorithms, and to identify and correct problems before they occur in the actual system.

There are many types of sensors that are used for the Internet of Things (IoT) depending on the application. Some of the most common types of sensors used in IoT include:

- *Temperature sensors:* These sensors are used to measure the temperature of a given environment or object. They can be used in various applications such as HVAC control, monitoring of refrigerated goods, and monitoring of industrial processes.
- *Humidity sensors:* These sensors are used to measure the humidity of a given environment or object. They can be used in various applications such as indoor air quality monitoring, agriculture, and weather forecasting.
- *Light sensors:* These sensors are used to measure the amount of light present in a given environment or object. They can be used in various applications such as lighting control, monitoring of plant growth, and monitoring of industrial processes.
- *Pressure sensors:* These sensors are used to measure the pressure of a given environment or object. They can be used in various applications such as weather forecasting, monitoring of industrial processes, and monitoring of tire pressure in vehicles.

- *Accelerometer sensors:* These sensors are used to measure the acceleration of an object. They can be used in various applications such as monitoring of vibrations in industrial equipment, monitoring of physical activity in wearables, and navigation in mobile devices.
- *Gyroscopic sensors:* These sensors are used to measure the angular rate of an object. They can be used in various applications such as navigation in mobile devices, monitoring of industrial equipment, and monitoring of physical activity in wearables.
- *GPS sensors:* These sensors are used to determine the location of an object. They can be used in various applications such as tracking of vehicles, monitoring of wildlife, and navigation in mobile devices.
- *Proximity sensors:* These sensors are used to detect the presence of objects. They can be used in various applications such as obstacle detection in robotics, touchless controls, and monitoring of industrial processes.
- *Optical sensors:* These sensors are used to detect light or changes in light. They can be used in various applications such as monitoring of industrial processes, monitoring of traffic, and surveillance.

All the sensors mentioned above are just a few examples of the many different types of sensors that are used in IoT applications. The specific type of sensor used depends on the IoT application.

II. LITERATURE REVIEW

Traditionally, the testing of control systems has primarily involved direct evaluation on physical equipment in the field, within the entire system or on a powerful testbed in a laboratory setting. Although this approach provides high testing fidelity, it comes with substantial drawbacks such as high costs, inefficiencies, and potential safety risks. An attractive alternative to these conventional methods is Hardware-in-the-Loop (HIL) testing. HIL testing offers a compelling solution by replacing the physical plant with a precisely equivalent software model [9-10], which operates in real-time on a simulator equipped with inputs and outputs (I/Os) capable of interfacing with control systems and other equipment. The HIL simulator replicates the behaviour of the actual plant, including its dynamics, sensors, and actuators, facilitating comprehensive closed loop testing without the necessity of testing on real systems. This approach not only ensures accuracy but also provides various additional functionalities, effectively mitigating the shortcomings of traditional testing methods. One of the significant advantages of HIL simulation is its ability to reduce risks, costs, and overall testing time for complex embedded systems. As a result, HIL simulation has become the preferred standard for numerous industries worldwide, serving as an efficient and reliable testing methodology.

A. What is HiL Testing?

Hardware-in-the-Loop (HIL) simulation is a technique used to test a functional system without the need for physical hardware. When applying HIL simulation to test a control system, it is referred to as HIL testing. Typically, the control system interfaces with the physical part of the system, known as the "plant," through actuators and sensors. However, HIL testing replaces the actual plant with a simulation, which we call the "HIL simulator." A well-designed HIL simulator accurately mimics the behaviour of the plant, enabling thorough testing of the control system. In some cases, advanced HIL simulators are equipped with 3D visualization, rendering the plant representation so realistic that they can also be utilized for training purposes, earning the name "training simulators." The advantage of using a simulated plant is that it allows running tests that would otherwise damage a real plant or pose risks to individuals in a real-world scenario. As a result, HIL simulation significantly enhances the safety of operating machines and systems.

Hardware-in-the-loop (HIL) simulation, also known as HWIL, is a powerful technique used in the development and testing of complex real-time embedded systems. It offers an effective platform by incorporating the intricacies of the controlled plant into the testing environment. This involves integrating a mathematical representation of all related dynamic systems, collectively referred to as the "plant simulation," to account for the complexity of the plant under control during testing and development. The embedded system being tested interacts with this plant simulation. For a HIL simulation to be comprehensive, it must include the electrical emulation of sensors and actuators. These electrical emulations serve as the intermediary between the plant simulation and the embedded system being tested. The values of each electrically emulated sensor are controlled by the plant simulation and then read by the embedded system under test (feedback). Similarly, the embedded system under test utilizes its control algorithms to generate actuator control signals. Changes in these control signals lead to corresponding alterations in variable values within the plant simulation. This interactive process ensures a thorough and accurate evaluation of the embedded system's performance.

As an illustration, consider a HiL simulation platform designed for developing automotive anti-lock braking systems. In this scenario, the plant simulation would include mathematical representations for various subsystems, such as the following:

- Vehicle dynamics, such as suspension, wheels, tires, roll, pitch, and yaw.
- Dynamics of the brake system's hydraulic components.
- Road characteristics.

Recently, software systems have become heterogeneous especially in automotive industry, where dealing with safety and involved complexity have raised an important issue. Usually there are several complex subsystem components attached physically to the core system and this makes the use of isolated yet connected use of many ECUs which perform a specific task relevant to the component attached. And the use of these component usage is growing exponentially day by day with the advent of electric vehicles which again involves a wide range of sensors and actuators to facilitate a more granular monitoring and control. This in turn have given rise to the number of functions to be performed by each component individually and hence increased the likelihood of errors that occur primarily due to the involvement of parallel tasks that a single control unit must handle or in some other cases these errors are prone to some unexpected environmental conditions which in no way could be handled without prior knowledge.

An inappropriate behaviour identified during runtime could be hazardous and it is certainly difficult to control the aftereffects of the hazards caused to the system under process. Therefore, it is a major issue to attend the continuously changing non-repetitive errors in the system while having inexperienced professionals in the given fields with no skills in analytical techniques. Though these techniques are incorporated at a later stage for identification of these dynamic errors they always come with some limitations thereby leaving the human and environmental hazards still open. Failure mode and effect analysis (FMEA) and Fault Tree Analysis (FTA) are in use since a long time in the form of analytical techniques for evaluating the documents required for a given specification. Therefore, proof-based experiments are much needed to overcome the traditional technique limitations especially with development of complex systems where the faults are not identified in advance.

Fault injection is another method which plays an important role in identifying the flaws in the system prior to its actual launch. There are numerous ways of identifying possible errors that may occur at the time of actual launch but with the features of simulation tools such as MATLAB, LTSpice, etc it is possible to not only observe the behavioural aspects of a process or system but also ingest error at run time which in turn provides insights when the system behaves abnormally. However, it is important to note that these errors are neither predicted nor does belong to a standard practice of error finding but can achieve substantial results to understand the effects of a variety of errors in each system. These days the error finding process is majorly used in the automotive sector where an entire process is simulated using software tools. Another similar method is emulating the part of the entire test bench [11] which is currently employed by almost every automotive manufacturer. This method of emulating the part of the entire process by simply replacing the real part with a similar electronic device that behaves and emulates the same signals as that of the part considered for replacement. This process is sometimes referred as XiL which stands for Anything-in-the-loop. As recommended by the ISO 26262 standard fault injection plays an important role in evaluating the effects of faults in the automotive industry for identifying the system level, component level and software unit errors. This process basically involves the ingestion of faults, failures or errors and the response of these sample ingestions are analysed to observe the behavioural changes at the desired test points of the system in a test bench.

The main contribution of the study is summarized below:

- The proposed frameworks enable real-time analysis of fault impacts on complex systems.
- Additionally, they facilitate the validation of real/soft ECU performance in both nominal and faulty conditions using intricate test scenarios.
- This approach considers a wide range of common automotive software signal faults, including gain, offset/bias, noise, hard-over, spike, stuck-at, packet loss, delay, and drift faults.
- Moreover, the framework allows for simulation and injection of multiple faults to analyse their combined impact on the system.
- Finally, fault modes are programmatically modelled and injected in real time via the HiL platform, avoiding the need to modify the original system model or add extra blocks.

B. Fault Injection Framework

The Fault Injection Graphical User Interface (FI GUI) is designed to assist testers in Hardware-in-the-Loop (HiL) frameworks by enabling real-time Fault Injection [12]. Testers can use the GUI to select fault types and values

from a library. For successful Fault Injection (FI), defining three key dimensions - fault type, fault location, and injection time - is crucial. Various potential faults like gain, offset, noise, and more are observed in time series data [13-15], caused by factors including sensor issues, faulty components, or external conditions like uneven roads.

III. KEY CHALLENGES

A significant set of challenges is centered around preemptively detecting faults before the integration of these systems into the overarching project or application. The inherent complexity lies in the integration process itself, entailing the seamless coordination of system elements, plant components, and their constituent subcomponents, encompassing interoperability concerns and web enablement, particularly relevant to Internet of Things (IoT) applications. While attempts have been made to address these challenges through certain descriptions and standardizations, a comprehensive resolution is yet to be achieved. Achieving success in this paradigm necessitates the conduction of meticulous experimental investigations coupled with the establishment of substantiated evidence.

A. What is sensor ML?

Sensor ML is a standard language for describing the characteristics and capabilities of sensors and sensor systems. It is an XML-based language that provides a common format for describing sensor metadata, such as sensor type, location, and measurement capabilities. This information can be used to discover and access sensors, as well as to integrate sensor data into other systems and applications. SensorML is a standard of the Open Geospatial Consortium (OGC) [16-17].

B. What is Sen ML?

Sen ML (Sensor Markup Language) is a lightweight, JSON-based format for describing sensor and device data. It is designed to be simple and easy to use, while also providing a powerful and flexible way to represent sensor data. Sen ML can be used to represent a wide range of sensor data, including measurements of physical quantities such as temperature, humidity, and light, as well as more abstract data such as sensor status and configuration information. Sen ML also includes a set of standard units for describing sensor data, which helps to ensure interoperability between different systems and devices.

C. What is a Sensor Schema?

A sensor schema is a structure or format that defines the data types, fields, and other information that a sensor can provide. It is used to describe the capabilities and characteristics of a sensor, such as the type of measurements it can take, the range of values it can measure, and the units of measurement it uses. A sensor schema can also include information about the sensor's location, manufacturer, and other metadata. Sensor schemas are commonly used in sensor networks and the Internet of Things (IoT) to allow devices and systems to discover, access, and integrate sensor data. Sensor schemas can be defined using a variety of languages and formats such as SensorML, SenML, and others. A sensor schema can be used to validate the data coming from the sensor, to ensure that the data is in the correct format and contains the expected fields, and also to help to understand the context of the data and what it represents. This can be particularly important in situations where sensor data is being used for critical applications, such as in industrial or medical settings.

An example sensor schema for temperature is shown below as a reference, based on which any sensor with suitable protocol can be defined.

TABLE I: REPRESENTATION OF SENSOR SCHEMA FOR A TEMPERATURE SENSOR

<pre> "sensorName": "TemperatureSensor", "sensorType": "Analog", "measurementUnit": "Celsius", "sensorModel": "LM35", "manufacturer": "Texas Instruments", "accuracy": "±0.5°C", "outputRange": { "minValue": 0, "maxValue": 1023 </pre>
--

```

},
"interface": "Analog",
"communicationProtocol": "Analog Voltage",
"description": "A temperature sensor that provides analog voltage output
proportional to temperature in Celsius."
}

```

D. What is Observations & Measurements (O&M)?

Observations & Measurements (O&M) is a standard for describing and encoding sensor observations and measurements. It is part of the Open Geospatial Consortium (OGC) Sensor Web Enablement (SWE) framework. O&M provides a model for describing sensor observations, including the feature of interest, the procedure used to see, and the result of the observation, along with other relevant metadata [18].

The O&M standard provides a common format for encoding sensor observations, which helps to ensure that sensor data can be easily shared and integrated into other systems. It also includes a standard vocabulary for describing sensor observations, which helps to ensure that sensor data can be understood and used across different domains and applications. O&M standard can be encoded in xml or json, and it's widely used in the fields of environmental monitoring and earth observation.

E. What is Sensor Web Enablement (SWE)?

Sensor Web Enablement (SWE) is a framework and set of standards developed by the Open Geospatial Consortium (OGC) for enabling the discovery, access, and integration of sensor data. SWE provides a set of web services and protocols for connecting sensors and sensor networks to the Internet, allowing sensor data to be shared and used by other systems and applications.

SWE includes several different standards, including the Sensor Observation Service (SOS) for accessing sensor data, the Sensor Planning Service (SPS) for managing sensor tasks, and the Transducer Markup Language (SensorML) for describing sensor capabilities [19]. SWE also includes the Observations & Measurements (O&M) standard for encoding sensor observations and measurements.

SWE is designed to make sensor data more accessible and usable, and it is widely used in a variety of domains including environmental monitoring, weather forecasting, and transportation. SWE services and standards are based on web protocols such as HTTP and SOAP, which allows for easy integration with other systems, and enables the sensor data to be accessible from any location via the internet.

F. What is standards development organization (SDO)?

A Standards Development Organization (SDO) is a type of organization that is responsible for developing and maintaining standards in a specific industry or field. These standards provide a common set of guidelines and specifications that help ensure that products, services, and systems are compatible and can work together effectively. SDOs are typically composed of experts and representatives from various organizations and industries. SDOs develop standards through a consensus-based process, where members come together to discuss, review, and agree upon the specifications of a standard. The process typically includes public reviews and comments, as well as testing and validation to ensure that the standards are technically sound and meet the needs of the industry. There are many different types of SDOs, such as international, national, and regional organizations, and they cover a wide range of industries and fields, including technology, manufacturing, construction, healthcare, and more. Some examples of well-known SDOs include the International Organization for Standardization (ISO), the American National Standards Institute (ANSI), and the Institute of Electrical and Electronics Engineers (IEEE). SDOs play a key role in helping to ensure that products and services are interoperable, safe, and reliable, and they help to promote innovation and competitiveness in the industries they serve.

REFERENCES

- [1] J. V. Barreras et al., "An Advanced HIL Simulation Battery Model for Battery Management System Testing," in IEEE Transactions on Industry Applications, vol. 52, no. 6, pp. 5086-5099, Nov.-Dec. 2016, doi: 10.1109/TIA.2016.2585539.
- [2] Lauss and K. Strunz, "Accurate and Stable Hardware-in-the-Loop (HIL) Real-Time Simulation of Integrated Power Electronics and Power Systems," in IEEE Transactions on Power Electronics, vol. 36, no. 9, pp. 10920-10932, Sept. 2021, doi: 10.1109/TPEL.2020.3040071.

- [3] S. Karimi, P. Poure and S. Saadate, "An HIL-Based Reconfigurable Platform for Design, Implementation, and Verification of Electrical System Digital Controllers," in *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1226-1236, April 2010, doi: 10.1109/TIE.2009.2036644.
- [4] S. R. Lee, E. Y. Ko, J. Lee and M. Dinh, "Development and HIL Testing of a Protection System for the Application of 154-kV SFCL in South Korea," in *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, pp. 1-4, Aug. 2019, Art no. 5602104, doi: 10.1109/TASC.2019.2897838.
- [5] Sarikhani and O. A. Mohammed, "HIL-Based Finite-Element Design Optimization Process for the Computational Prototyping of Electric Motor Drives," in *IEEE Transactions on Energy Conversion*, vol. 27, no. 3, pp. 737-746, Sept. 2012, doi: 10.1109/TEC.2012.2200897.
- [6] S. Karimi, P. Poure and S. Saadate, "An HIL-Based Reconfigurable Platform for Design, Implementation, and Verification of Electrical System Digital Controllers," in *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1226-1236, April 2010, doi: 10.1109/TIE.2009.2036644.
- [7] M. Rezkallah, A. Hamadi, A. Chandra and B. Singh, "Real-Time HIL Implementation of Sliding Mode Control for Standalone System Based on PV Array Without Using Dumpload," in *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1389-1398, Oct. 2015, doi: 10.1109/TSTE.2015.2436333.
- [8] <https://opensensorhub.org/home/welcome/explore/>
- [9] V. V. Nair and P. Pathiyil, "High dynamic HIL model for complete software testing solution of Hev/Ev," 2015 IEEE International Transportation Electrification Conference (ITEC), 2015, pp. 1-7, doi: 10.1109/ITEC-India.2015.7386869.
- [10] Y. Liang and G. -P. Liu, "Design of Large Scale Virtual Equipment for Interactive HIL Control System Labs," in *IEEE Transactions on Learning Technologies*, vol. 11, no. 3, pp. 376-388, 1 July-Sept. 2018, doi: 10.1109/TLT.2017.2731772.
- [11] W. Li and J. Bélanger, "An Equivalent Circuit Method for Modelling and Simulation of Modular Multilevel Converters in Real-Time HIL Test Bench," in *IEEE Transactions on Power Delivery*, vol. 31, no. 5, pp. 2401-2409, Oct. 2016, doi: 10.1109/TPWRD.2016.2541461.
- [12] Jandaghi and V. Dinavahi, "Real-Time HIL Emulation of Faulted Electric Machines Based on Nonlinear MEC Model," in *IEEE Transactions on Energy Conversion*, vol. 34, no. 3, pp. 1190-1199, Sept. 2019, doi: 10.1109/TEC.2019.2891560.
- [13] Moradi, M.; Van Acker, B.; Vanherpen, K.; Denil, J. Model-implemented hybrid fault injection for Simulink (tool demonstrations), *Cyber Physical Systems. Model-Based Design*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 71–90.
- [14] Yang, J.L.; Chen, Y.s.; Zhang, L.L.; Sun, Z. Fault detection, isolation, and diagnosis of self-validating multifunctional sensors. *Rev. Sci. Instruments* 2016, 87, 065004.
- [15] Saeed, U.; Jan, S.U.; Lee, Y.D.; Koo, I. Fault diagnosis based on extremely randomized trees in wireless sensor networks. *Reliab. Eng. Syst. Saf.* 2021, 205, 107284.
- [16] <https://opensensorhub.org/2020/11/30/augmented-sensor-observations-and-tasking-through-artificial-intelligence-ai-machine-learning-ml-and-computer-vision-cv-processes-in-opensensorhub/>
- [17] Jin M, Bai Y, Devys E, Di L. Toward a Standardized Encoding of Remote Sensing Geo-Positioning Sensor Models. *Remote Sensing*. 2020; 12(9):1530. <https://doi.org/10.3390/rs12091530>
- [18] A Sensor Model Language: Moving Sensor Data onto the Internet | Fierce Electronics by Lance McKee, <https://www.fierceelectronics.com/iot-wireless/a-sensor-model-language-moving-sensor-data-onto-internet>
- [19] Tagliolato P, Fugazza C, Oggioni A, Carrara P. Semantic Profiles for Easing SensorML Description: Review and Proposal. *ISPRS International Journal of Geo-Information*. 2019; 8(8):340. <https://doi.org/10.3390/ijgi8080340>