Received July 12, 2019, accepted August 4, 2019, date of publication August 12, 2019, date of current version August 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2934420

Low-Cost Hardware-in-the-Loop Platform for Embedded Control Strategies Simulation

ALCEU BERNARDES CASTANHEIRA DE FARIAS, REURISON SILVA RODRIGUES, ANDRÉ MURILO, RENATO VILELA LOPES, AND SUZANA AVILA
Faculty of Gama, University of Brasília, Brasília 72444-240, Brazil

Corresponding author: Alceu Bernardes Castanheira de Farias (alceu.castanheira@gmail.com)

This work was supported in part by the National Council for Scientific and Technological Development (CNPq) and in part by the Coordination of Superior Level Staff Improvement (CAPES).

ABSTRACT The increasing need for testing and prototyping designs under more realistic conditions is responsible for the advancement of new types of simulation. In this scenario, one type of simulation which has gained high notoriety and applicability is the one known as Hardware-in-the-Loop (HIL). This technique allows real and virtual components of a system to be tested together, making it possible to perform tests under realistic (and even extreme) conditions without harming the real system or a prototype built only for testing. The objective of this work was to develop a low-cost HIL simulation platform to be used for many different applications, unlike most commercial ones, that are developed for one exclusive field of application, such as automotive, aerospace, power electronics, among others. Thus, the main contribution of this work is the project of a HIL platform capable of simulating different types of systems, making it possible to validate embedded control strategies designed for them. Two different applications are tested in order to validate the HIL platform: an active suspension and a satellite attitude control air bearing table, both controlled using a discrete Linear Quadratic Regulator (LQR) designed for each of them.

INDEX TERMS Active suspension systems, control systems, hardware-in-the-loop, satellite attitude control.

I. INTRODUCTION

The increasing advancements in the field of computation and the need for testing and prototyping different types of systems faster, in order to reduce time-to-market and cost issues, has led to the creation of sophisticated and complex simulation tools for many areas of study. However, despite being widely used, in most cases, traditional software-based simulation has the disadvantage of being unable to exactly replicate real operational conditions [1]. One approach to solve this problem and attempt to reduce this gap between simulation and real conditions is a type of simulation known as Hardware-in-the-loop (HIL).

This technique is characterized by the operation of real components of a system in connection with simulated ones. Even though it is a not a rule, usually the control system hardware and software are implemented as real components and the controlled process (actuators, physical processes, and sensors) are fully or partially simulated, mainly due to the controlled process either not being available or because experiments with the real process involve high costs and require too much time [2]. Fig. 1 presents the possible structures for a HIL simulation, showing that either the controller or the process can be real, simulated or partially simulated.

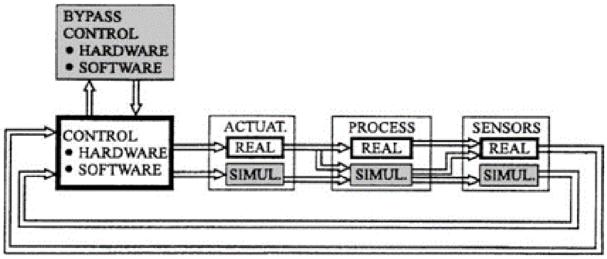

FIGURE 1. Possible hybrid structures for Hardware-in-the-loop (HIL) simulations. Reprinted from: [2].

The greatest advantage regarding HIL simulations lies in being able to perform more realistic tests to validate control systems, using real hardware and including all runtime effects without having to use the real system [3], which is specially useful when the testing of control systems under extreme conditions is needed, helping to diagnose failures/malfunctions

The associate editor coordinating the review of this article and approving it for publication was Tao Wang.

without harming or even destroying the developed system. These advantages are so important to industrial applications, that three recent and legal control handbooks (The Control Handbook, NASA Systems Engineering Handbook, and ECSS Control Standard) introduced HIL as the last step of validation of the development process of a system [4].

There are many industrial fields of application that use HIL simulations in order to validate control systems. In [2] it is stated that the first approaches to HIL simulation were probably regarding flight simulators, in which early goals consisted of simulating instruments with a fixed-cockpit, and later on moving a cockpit according to aircraft motions, e.g. for the training of pilots. Since then, aerospace systems have been using this technique as a standard for validating control strategies, from older applications such as the Cassini Mission (a joint effort between European Space Agency (ESA) and National Aeronautics and Space Administration (NASA) in order to study Saturn and its natural satellites) to newer ones, such as the testing of permanent-magnet-assisted synchronous reluctance motors (PMa-SynRM) for aerospace vehicles [6], Unmanned Aerial Vehicles (UAVs) like quadrotors [7] and UAVs used for mapping disaster-struck areas [8], and many other applications.

Other two important areas of industry in which HIL simulations are vastly used and are considered effective methodologies for testing control systems are automotive and power control applications. Regarding the first one, there are many studies available in literature using HIL simulations for validating automotive control systems under more realistic scenarios: experimental validation of the propulsion system of a GM Chevrolet Volt 2nd Generation electric car [9], lateral stability and rollover prevention via active braking [10], design and statistical validation of spark ignition engine electronic control units [11], fault injection strategy for real-time simulation in traction control systems (TCS) [12], hydraulic pressure control in automotive braking systems [13], estimation of battery in electric vehicles [14], among many others.

In the power control segment, real-time HIL simulations have been recognized as an advanced method for the analysis and testing of power system phenomena and components, being at the same time realistic and flexible when it comes to testing conditions for de-risking pieces of equipment [15]. It is such a common practice in this field of industry that it is known as Power Hardware-in-the-Loop (PHIL) [16], including many applications in literature such as power converter digital controllers [17], switched-mode power amplifiers [18], frequency-response measurements of on-board power distribution systems [19], and several others applications.

Other important fields of industry that apply HIL simulations include robotics, as it can be seen in [20] and [21], and even Internet of Things scenarios, in which HIL simulations are considered very useful, due to the complexity of creating and monitoring test setups fully consisting of real hardware [22].

Despite all the advantages presented so far and the variety of applications that use HIL simulations, there are still some disadvantages regarding this procedure. According to [23], although HIL simulations are one of the closest tests to reality, it is a stage that takes more time to perform and is more expensive than other validation methodologies for control systems. This is supposedly due to the type of hardware and software required to perform the tests.

Another important aspect to consider is that these platforms are designed for one specific area of application, as it could be seen by the examples previously discussed in this section. Many universities do not have either the required money to buy industrial hardware and equipment for the education of engineering students or space and means to maintain them, which makes HIL systems strong candidates to be used for such purpose [24]. The main problem is that commercial HIL platforms can reach prices on the range of €200.000,00 (as it is the case of automotive HIL platforms for simulating Electronic Control Units (ECUs), for instance), making it difficult for universities to acquire them.

Within this context, the main objective, and also the main contribution of this paper, is to present a low-cost HIL platform, developed at the University of Brasilia, capable of performing HIL simulations with different types of systems to validate embedded control strategies designed for them, especially academic ones.

For validating the platform, two different types of systems are simulated: an automotive application regarding an active suspension system and an aerospace application of the attitude control of a satellite using an air bearing table, controlled using a discrete Linear Quadratic Regulator (LQR) applied to both models. These two segments of industry widely use HIL simulations, as presented in this current section, and as a consequence, there are many academic applications related to these areas. Therefore, these systems mean to be two interesting study cases.

This work is divided as it follows: Section II introduces the concept of Model-Based Design, which is a methodology for developing systems that includes not only HIL but also other test possibilities; Section III presents the architecture of the HIL platform developed at this work and how its functionality; Section IV introduces the mathematical modeling of the two systems used for testing the HIL platform (active suspension system and satellite attitude control air bearing table) and the mathematical modeling of the LQR control strategy adopted for both models; Section V discusses the methodology for performing HIL simulations and how tests were made; Section VI show the results obtained and the discussions regarding them. Finally, section VII concludes this work, presenting future work propositions.

II. VALIDATION OF CONTROL SYSTEMS USING MODEL-BASED DESIGN (MBD)

The validation of control systems must be done in a series of steps, obeying design requirements. One of the strategies for

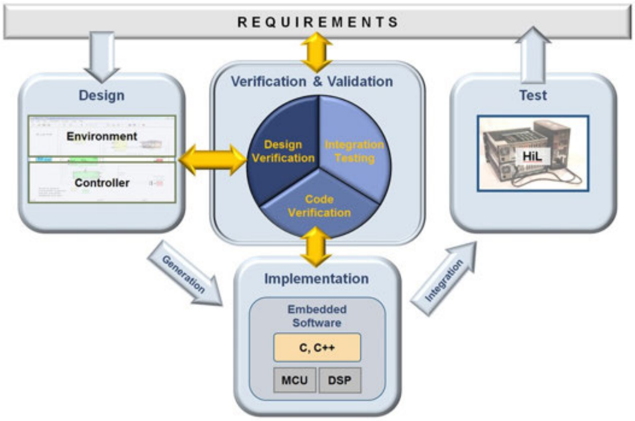

FIGURE 2. Model-Based Design visual scheme and steps: design requirements gathering, system design, implementation, verification, validation and testing. Reprinted from: [27].

system development that allows such procedures is Model-Based Design (MBD), which is being widely used today [25]. This methodology offers a systematic way to handle the development of complex control systems through a visual and mathematical approach and has been recognized for assisting systems development, becoming standard practice in systems engineering [26]. The visual scheme of MBD can be seen in Fig. 2, in which the system model is considered the center of the design [27].

The first step of MBD is the identification and documentation of the existing requirements and operating characteristics that the model should obey. The following step corresponds to the design phase, in which the requirements proposed earlier are analyzed by the developers, as they try to establish a global model and divide it into modules, each one with specific functionalities. Once these modules have been defined, they are implemented, whether through automatic code generation or manual development. To support each step of the system design, continuous tests are executed to validate the modules developed and identify errors in the earliest stages of development.

Although the steps involved in MBD were described in a specific order, it is not necessarily the order that the development project should follow. The detection of errors in a certain phase can lead to the return to previous phases of the project.

A series of tests for validation can be used in MBD, each one applied to a specific phase of the project. These tests are known as Model-in-the-loop (MIL), Software-in-the-loop (SIL) and Hardware-in-the-loop, that has already been mentioned. The MIL phase is represented in Fig. 3 and corresponds to the development of the control system and the model of the plant in the same virtual environment, in software such as MATLAB. The aim is to verify if the dynamics and the response of the system meet the requirements established.

In SIL, Fig. 4, the code of the controller is written or automatically generated and tested in the same virtual environment of the plant model. This phase focus on ensuring the correctness of the code since it will be embedded in real hardware. These results may then be compared to the results obtained in the MIL phase.

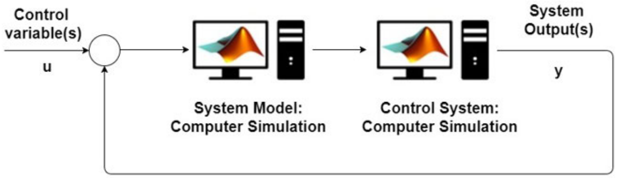
Control variable(s)
u
System Model: Computer Simulation
Control System: Computer Simulation
System Output(s)
y

FIGURE 3. Schematic view of Model-in-the-Loop (MIL). In this configuration, both the system model and the controller run in the same virtual environment, aiming to validate the proposed closed-loop.

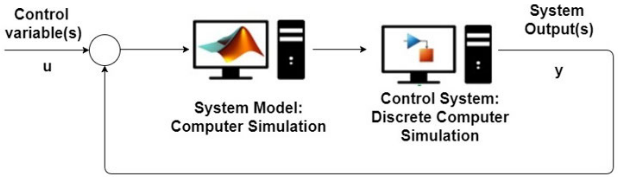
Control variable(s)
u
System Model: Computer Simulation
Control System: Discrete Computer Simulation
System Output(s)
y

FIGURE 4. Schematic view of Software-in-the-Loop (SIL). In this configuration, the controller is written or automatically generated in software, running in the same virtual environment as the system model in order to validate the control algorithm that will be embedded in hardware.

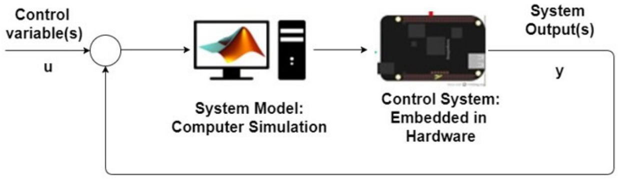
Control variable(s)
u
System Model: Computer Simulation
Control System: Embedded in Hardware
System Output(s)
y

FIGURE 5. Schematic view of Hardware-in-the-Loop (HIL). In this configuration, the control algorithm is embedded in real hardware and the model runs in real-time on a virtual environment, simulating inputs and outputs closer to the real operating conditions.

Once the results of both MIL and SIL agree with each other, the next phase corresponds to the HIL test. At the HIL phase, Fig. 5, the controller code is embedded in hardware and the system model is simulated in some device capable of operating in real-time, with simulated inputs and outputs, aiming to reproduce the nearest possible behavior of the real system.

There are other intermediate test possibilities as well, such as Processor-in-the-Loop (PIL), in which the developed controller code using SIL is embedded in an external hardware device (not necessarily the final hardware for the application), while the system model remains simulated in a virtual environment (not necessarily in real-time). However, in this work, focus is given to the test phases presented earlier: MIL, SIL, and HIL.

Regarding the test phases presented, it is possible to notice that not only HIL is the last validation step on the MBD flow, but also that there are many steps regarding system design before it is possible to run an appropriate HIL simulation. This is very important to take into consideration when developing a HIL platform, as it needs to offer all the necessary conditions for following the flow presented in Fig. 2 and validating the desired control system.

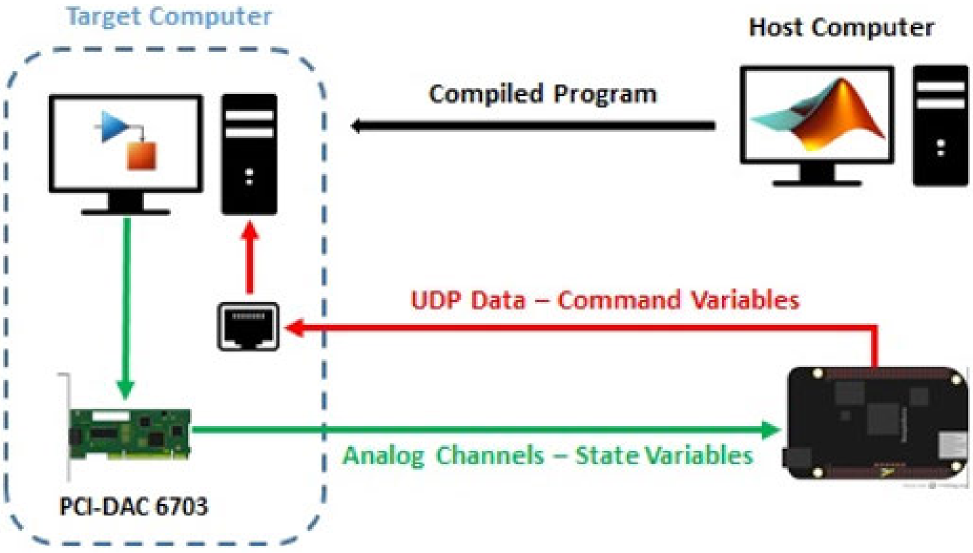
Target Computer
Host Computer
Compiled Program
UDP Data – Command Variables
Analog Channels – State Variables
PCI-DAC 6703

FIGURE 6. Proposed low-cost HIL platform architecture developed at the University of Brasilia.

III. ARCHITECTURE OF THE HARDWARE IN THE LOOP PLATFORM

As stated in Section I, the main contribution of this work is to present a HIL platform that can be used for simulating different types of systems. Also, it is desired that it is has low-cost, compared to commercial alternatives. In order to achieve such goals, it was necessary to use an architecture that not only divided the plant/controlled process model from the control system, but also made it possible to change easily from one model to another, so that other types of systems can be tested. The components used have to be analysed not only on this aspect, but also on the best benefit-cost ratio.

There are many possible hardware components for the development of the HIL platform. One type of system that has been found quite frequently in literature is the Field Programmable Gate Array or FPGA. This is a semiconductor device composed of logic elements that offer speedup alternatives for real-time simulations, without losing accuracy or introducing excessive communication latency [28]. However, such type of device is not used in the architecture of this HIL platform for a couple of reasons. First, efficient implementation of systems using FPGAs require advanced and complex hardware description languages coding and in-depth knowledge of the architecture of the FPGA being used [29]. Second, even though most applications utilize some sort of code generation tool (usually known as High Level Synthesis or HLS) for facilitating the implementation of complex systems using FPGAs, such as System Generator and DSP Builder, even the newest generation of HLS tools does not provide as good performance and resource usage as manual development coding does [30].

Therefore, for a first approach, a more user-friendly hardware device was chosen for the HIL platform, resulting in the architecture that can be seen in Fig. 6. The proposed HIL platform, developed at the University of Brasilia, contains three main components: a host-target set of computers responsible for simulating the controlled process, a microprocessor that is responsible for implementing the control system embedded in real hardware and signal conditioners that allow communication between simulated controlled process and control system embedded in real hardware.

Each component of the HIL platform will be discussed in more details in the following subsections.

A. *HOST AND TARGET COMPUTERS*

The host computer makes it possible to set up the model to be simulated. It contains the Simulink environment, in MATLAB, which is used for modeling the different types of systems that are intended to be controlled. Once the model is developed using the host computer, it can be loaded into the target computer, as an application that runs on its Simulink Real-Time kernel.

The Simulink Real-Time toolbox is the most recent version of the xPC Target toolbox, which has been widely used for simulating systems in real-time. It allows the creation of real-time applications from Simulink models that can be executed on dedicated target computer hardware.

It is important to state that the use of Simulink Real-Time is the basis that allows the testing of different types of systems in the HIL platform. It allows the inclusion of models that express varying dynamic behaviors of a system, whether they are linear or non-linear systems, high-order systems, discontinuities, look-up tables, and many others. In other words, this allows the simulation of more complex and complete real-time models, which makes it possible to simulate systems in a more realistic scenario, with characteristics that resemble even more the real system operating conditions.

The communication interfaces associated with the target computer (PCI-DAC 6703 and an Ethernet port) will be explained in further details in Subsection III-C.

B. MICROPROCESSOR: EMBEDDED CONTROL SYSTEM

The control system is implemented as the real part of the system, in a microprocessor. The chosen hardware for this task is the BeagleBone Black Rev C, since it has a great trade-off between processing capabilities and cost, as well as containing 7 analog inputs (A0 - A6). This last feature is important, as few low-cost microprocessors in the market have that many analog inputs available (usually digital-to-analog converters are needed for extra analog inputs), which makes it easier to interface the data acquisition system contained in the platform (DAC-6703) with the controller. It is a low-power open-source hardware single-board computer, containing 512 MB RAM, 1 GHz processor clock and 4 GB of eMMC flash memory. It is able to run an embedded GNU/Linux distribution, allowing the user to connect several external peripherals. It is also responsible for executing the control algorithm embedded in it and transmitting the control variables, using an UDP protocol communication, through an Ethernet cable (represented in Fig. 6 by the red line), to the model being simulated at the target computer.

BeagleBone Black brings flexibility to the control hardware as well, since it can be programmed using different types of programming languages, such as C, $C{+}{+}$, Python, JavaScript, and many others.

As it represents the control system of the desired application, BeagleBone Black must be capable of reading the outputs of the system being simulated on the target computer running the Simulink Real-Time kernel, as well as sending the control inputs to the same system, where they will be applied in order to achieve the desired dynamic behavior. This is done with the analog channels present in BeagleBlack Rev C (used for reading the outputs of the simulated model that are made available as analog outputs of the DAC-6703, as explained in Subsection III-A) and the UDP protocol, using an Ethernet cable (that makes it possible to send the command variables to the simulated model).

Since BeagleBone Black has 7 analog inputs, the platform is limited to simulation of systems that contain a maximum of 7 outputs or variables of interest. Despite being a practical limitation of this architecture, it still allows the testing of a good range of applications. Also, it is possible to use techniques such as state observers or send any extra variables needed for the HIL simulation through UDP in order to accurately simulate systems with more than 7 outputs.

C. SIGNAL CONDITIONERS

The signal conditioners are the parts of the platform responsible for allowing communication between the simulated model using the Simulink Real-Time kernel and the control algorithm embedded in BeagleBone Black. For the proposed HIL platform architecture, the signals conditioners are the DAC-6703 and an Ethernet cable used for implementing an UDP protocol.

PCI DAC-6703 is a data acquisition system that interfaces with Simulink Real-Time, making the outputs of the system become available as either digital or analog outputs. This opens the possibility of simulating both digital and analog variables, according to the type of system being simulated. At this work, both systems tested have analog outputs, so only the analog channels of the DAC-6703 component are being used.

One of the main reasons for choosing PCI-DAC 6703 as the data acquisition system for the HIL platform was that it contains 16 analog channels, which are used to interface with the BeagleBone Black analog inputs, and has compatibility with Simulink Real-Time, which is mandatory. There are other options in the market, such as devices from National Instruments, that possess these same characteristics, but with the low-cost aspect of the HIL platform in mind, PCI-DAC 6703 presented the desired features with the lowest price at the time (approximately $ 2412,35).

The other signal conditioner presented in the HIL platform is the Ethernet cable, implementing an UDP protocol. UDP is a serial communication protocol commonly used because of its lightweight nature. When used with Simulink Real-Time, UDP gives the application a good chance of succeeding in real-time execution. It is suitable for purposes where error checking and correction are either not necessary or are performed in the application, once UDP avoids the overhead of such processing at the level of the network interface. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, which may not be an option in a real-time system [31]. Also, the datagram nature of UDP is optimal for sending samples of data from the real-time application generated by the Simulink Coder software.

It is possible to find works of HIL platforms on literature using UDP protocol to communicate simulated and real components, such as in [32], in which UDP is used in a HIL platform for testing rotatory-wing unmanned aerial vehicles; [33] for longitudinal autopilot controllers testing in HIL platforms using UDP and [34], that utilizes UDP in a HIL platform for controlling and synchronizing a system containing two DC servo motors.

D. OVERVIEW OF THE HARDWARE-IN-THE-LOOP PLATFORM

With the architecture presented in this section, it is possible to model different types of systems using MATLAB and develop control algorithms using the BeagleBone Black microprocessor, which can be programmed using different programming languages.

Once again, it is important to reinforce that this is the major contribution of this work: a HIL platform that can be used for simulating different types of systems. Given the high-cost aspect of commercial HIL platforms (as stated in Section I), this becomes a great advantage, especially for academic purposes. For comparison with the example given in Section I, the final cost of the developed platform is approximately $ 4586,00, which reinforces the low-cost aspect of this work.

It is important to notice that, given the difference in cost between both solutions, commercial HIL platforms yield more complete results. However, for academic applications, this HIL platform is capable of satisfactory performances, as it will be discussed in the forthcoming sections.

IV. MATHEMATICAL MODELING OF TEST SYSTEMS

Since the main contribution of this work is to present a HIL platform capable of performing simulations for different types of systems, this section presents the modeling of the systems used in this work to validate the proposed HIL platform. First, an active suspension system model is presented in Subsection IV-A, followed by the modeling of a satellite attitude control air bearing system in Subsection IV-B. Finally, Subsection IV-C presents the modeling of the discrete LQR controller, which is applied to the active suspension model and the air-bearing table system.

A. ACTIVE SUSPENSION SYSTEM

Vehicle suspension systems are composed of a system of springs, shock absorbers and arms that connect a car body to its wheels and allows relative motion between the two [35]. According to [36], active suspension control has been a widely implemented control procedure, leading to significant developments in improving passenger comfort while maintaining good road holding capabilities for commercial vehicles. The active term is associated with a control system, which helps to improve and realize the full potential of the active suspension system as a whole [37].

The automotive suspension model discussed in this paper is based on a quarter car vehicle model. According to [38], an example of this type of system is shown in Fig. 7.

The main parameters used in this model are:

- Road profile or excitation (z_0);
- Position of the unsprung mass (z_{us});
- Position of the sprung mass (z_s);
- Sprung mass (m_s);
- Unsprung mass (m_{us});
- Suspension stiffness (k_s);
- Suspension damping (c_s);
- Tire stiffness (k_{us});
- Tire damping (c_{us});
- Active suspension force (f).

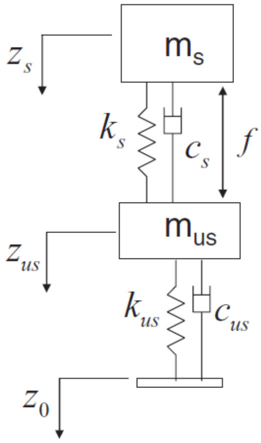

FIGURE 7. Active suspension system based on a quarter car vehicle model. Reprinted from: [38].

The active suspension system inputs are the active suspension force, which is the command variable of the system ($u(t) = f(t)$) and the disturbance or road excitation displacement input (z_0).

The system state variables are given by the tire deflection ($z_{us} - z_0$ or x_1), velocity of the unsprung mass ($\dot{z}_{us}$ or x_2), suspension travel ($z_s - z_{us}$ or x_3) and, finally, the velocity of the sprung mass ($\dot{z}_s$ or x_4). The state variables x_1 and x_3 represent the position of the unsprung mass in relation to the ground and the position of the sprung mass in relation to the unsprung mass, respectively. Also, they are considered important system outputs: tire deflection (y_1) and suspension stroke (y_2). There is also a third variable of interest: the acceleration of the suspension, given by $\ddot{z}_s$ or y_3.

The equation of motion that describes the dynamic behavior of the suspension is given by (1) and (2):

$$\begin{aligned} -f(t) &= m_s\ddot{z}_s(t) + c_s(\dot{z}_s(t) - \dot{z}_{us}(t)) \\ &\quad + k_s(z_s(t) - z_{us}(t)). \end{aligned} \tag{1}$$

$$\begin{aligned} f(t) &= m_{us}\ddot{z}_{us}(t) + c_s(\dot{z}_{us}(t) - \dot{z}_s(t)) \\ &\quad + k_s(z_{us}(t) - z_s(t)) + c_{us}(\dot{z}_{us}(t) - \dot{z}_0(t)) \\ &\quad + k_{us}(z_{us}(t) - z_0(t)). \end{aligned} \tag{2}$$

These equations can be expressed in matrix form, allowing the representation of the active suspension model in state-space continuous form (3) and (4). In these equations, **A** is the states matrix, x is the states vector, **B** is the inputs matrix, u is the inputs vector, **G** is the input disturbance matrix, w is the disturbance itself, **C** is the outputs matrix, and **D** is the direct transmission matrix.

$$\dot{x} = \mathbf{A}x + \mathbf{B}u + \mathbf{G}w. \tag{3}$$

$$y = \mathbf{C}x + \mathbf{D}u. \tag{4}$$

In this work, the perturbation w corresponds to the excitation profile in terms of velocity of the road, or $\dot{z}_0$. Substituting (1) and (2) into (3) and (4), and normalizing the state variables according to (5), we obtain the matrices that characterize the system in state space: (6).

$$\begin{aligned} &\rho = \frac{m_s}{m_{us}}, \\ &\omega_1 = \sqrt{\frac{k_{us}}{m_{us}}} \quad \omega_2 = \sqrt{\frac{k_s}{m_s}}, \\ &\xi_1 = \frac{c_{us}}{2m_{us}\omega_1} \quad \xi_2 = \frac{c_{us}}{2m_{us}\omega_2}. \end{aligned} \tag{5}$$

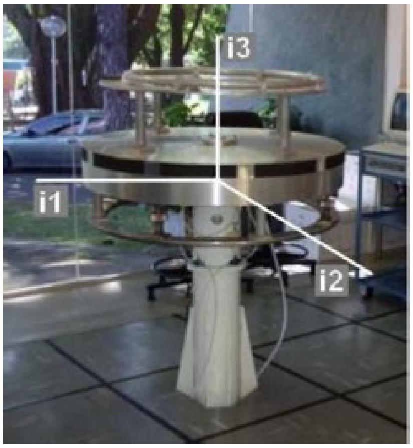

FIGURE 8. Air bearing table for satellite attitude control. Reprinted from: [39].

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_1^2 & -2(\rho\xi_2\omega_2 + \xi_1\omega_1) & \rho\omega_2^2 & 2\rho\xi_2\omega_2 \\ 0 & -1 & 0 & 1 \\ 0 & 2\xi_2\omega_2 & -\omega^2 & -2\xi_2\omega_2 \end{bmatrix},$$
$$B = \begin{bmatrix} 0 \\ \rho \\ 0 \\ -1 \end{bmatrix},$$
$$G = \begin{bmatrix} -1 \\ 2\xi_1\omega_1 \\ 0 \\ 0 \end{bmatrix}. \tag{6}$$

B. SATELLITE ATTITUDE CONTROL

In order to test and verify software and embedded electronics for satellite attitude applications, air bearing tables are useful options. The objective of using such equipment is to provide conditions similar to space, i.e., three-axis angular movement and nearly frictionless environment. The air bearing table model discussed in this paper is based on [39], which uses reaction wheels as actuators, is shown in Fig. 8.

The inertial reference system $F_i(i_1, i_2, i_3)$ is located in the center of the spherical bearing, considered the center of rotation of the simulator. The reference system of the body F_b is considered to have the same center, varying only its orientation with respect to the inertial system. The equations that describe the dynamic behavior of the air bearing table are given by (7).

$$\begin{aligned}
\dot{\theta}_1 &= \omega_2 \cdot \frac{sin(\theta_3)}{cos(\theta_2)} + \omega_3 \cdot \frac{cos(\theta_3)}{cos(\theta_2)}, \\
\dot{\theta}_2 &= \omega_2 \cdot cos(\theta_3) - \omega_3 \cdot sin(\theta_3), \\
\dot{\theta}_3 &= \omega_1 + \omega_2 \cdot \frac{sin(\theta_3) \cdot sin(\theta_2)}{cos(\theta_2)} + \omega_3 \cdot \frac{cos(\theta_3) \cdot sin(\theta_2)}{cos(\theta_2)}, \\
\dot{\omega}_1 &= \omega_2 \cdot \frac{(I_{22}\omega_3 - I_\omega\Omega_3)}{(I_{11} + I_\omega)} + \omega_3 \cdot \frac{(-I_{33}\omega_2 + I_\omega\Omega_2)}{(I_{11} + I_\omega)} \\
&\quad - \dot{\Omega}_1 \cdot \frac{I_\omega}{(I_{11} + I_\omega)},
\end{aligned}$$

$$\begin{aligned}
\dot{\omega}_2 &= \omega_1 \cdot \frac{(-I_{11}\omega_3 + I_\omega\Omega_3)}{(I_{22} + I_\omega)} + \omega_3 \cdot \frac{(I_{33}\omega_1 - I_\omega\Omega_1)}{(I_{22} + I_\omega)} \\
&\quad - \dot{\Omega}_2 \cdot \frac{I_\omega}{(I_{22} + I_\omega)}, \\
\dot{\omega}_3 &= \omega_1 \cdot \frac{(I_{11}\omega_2 - I_\omega\Omega_2)}{(I_{33} + I_\omega)} + \omega_2 \cdot \frac{(-I_{22}\omega_1 + I_\omega\Omega_1)}{(I_{33} + I_\omega)} \\
&\quad - \dot{\Omega}_3 \cdot \frac{I_\omega}{(I_{33} + I_\omega)}.
\end{aligned} \tag{7}$$

The system has six state variables: $x = (\theta_1\ \theta_2\ \theta_3\ \omega_1\ \omega_2\ \omega_3)^T$. The variables θ_1, θ_2 and θ_3 are the Euler angles that describe the attitude of the simulator as the relative orientation between the inertial frame F_i and the reference fixed in the body F_b. The variables ω_1, ω_2 and ω_3 are the angular velocities of F_b with respect to F_i. The inputs vector is composed of three terms representing the acceleration on each reaction wheel: $u = (\dot{\Omega}_1\ \dot{\Omega}_2\ \dot{\Omega}_3)^T$. The constants I_{11}, I_{22}, I_{33} are the moments of inertia around the axes i_1, i_2 and i_3, respectively, and I_w corresponds to the moment of inertia of the reaction wheels. Ω_1, Ω_2 and Ω_3 are the velocities of the three reaction wheels.

C. CONTROL SYSTEM

It is not the aim of this work to develop a new control strategy or improve existing controllers for the applications presented. Therefore, to validate the HIL platform, the chosen control strategy for both applications presented is a well established control technique in literature: Linear Quadratic Regulator or LQR, which consists of computing the state feedback gain that optimizes a quadratic cost function relating the states and inputs of the plant/system being controlled [40], while also being capable of dealing with Multiple-Input Multiple-Output (MIMO) systems.

In order to be able to perform proper HIL simulations using this control strategy, it is necessary to apply discrete control equations, making it possible to implement the control algorithm embedded in the BeagleBone Black microprocessor. The discrete LQR control law can be seen as a simple state feedback for each time instant n, as shown below (8).

$$u[n] = -\mathbf{K}x[n]. \tag{8}$$

The matrix feedback gain $\mathbf{K}$ can be calculated according to (9), which depends on the value of the Ricatti $\mathbf{S}$ matrix obtained by solving the discrete Ricatti equation (10). $\mathbf{A_d}$ and $\mathbf{B_d}$ represent the state and input matrices in discrete form, respectively.

$$\mathbf{K} = (\mathbf{B_d}^T\mathbf{S}\mathbf{B_d} + \mathbf{R})^{-1}(\mathbf{B_d}^T\mathbf{S}\mathbf{A_d} + \mathbf{N}^T). \tag{9}$$
$$\begin{aligned}
&\mathbf{A_d}^T\mathbf{S}\mathbf{A_d} - \mathbf{S} - (\mathbf{A_d}^T\mathbf{S}\mathbf{B_d} + \mathbf{N})(\mathbf{B_d}^T\mathbf{S}\mathbf{B_d} + \mathbf{R})^{-1} \\
&\quad (\mathbf{B_d}^T\mathbf{S}\mathbf{A_d} + \mathbf{N}^T) + \mathbf{Q} = 0.
\end{aligned} \tag{10}$$

The discrete LQR controller is used to control both the active suspension and the satellite attitude control air bearing systems, that were presented in the previous subsections. Each application formulates a control problem with different dimensions, that requires different parameters for each system being controlled. The general form of the quadratic cost

TABLE 1. Overview of the characteristics of the active suspension and satellite attitude control models.

Model	Number of states	Number of control variables	Sampling time
Active Suspension	4	1	0.2 ms
Satellite Attitude Control Air Bearing Table	6	3	100 ms

function adopted in this work for the LQR controller and that is given by (11).

$$J(u) = x[n]^T \mathbf{Q} x[n] + u^T[n]\mathbf{R}u + 2x^T[n]\mathbf{N}u[n]. \quad (11)$$

These weighting matrices given by matrices **Q**, **R**, and **N** used for the cost function of each application are presented in Section V.

V. EXPERIMENTAL PROCEDURES

The experimental tests aim to validate the HIL platform using both models and the control system presented in subsections IV-A, IV-B, and IV-C, respectively, through MIL, SIL and HIL testing. Table 1 presents an overview of both models, containing their main characteristics.

The first step is to validate the model and the LQR controller in the MIL phase, using MATLAB and the Simulink environment. The next step is to validate the code of LQR controller used, through the SIL test. The authors chose to write the code in C, due to familiarity with the language and its speed of execution, but there are other options available, such as code generation tools provided by MATLAB. The final step corresponds to the HIL tests: the LQR control algorithm written in C runs in BeagleBone Black and the air bearing table and active suspension models are simulated using Simulink Real-Time.

Once all tests are finished, it is possible to compare results from MIL and HIL in order validate the developed HIL platform: if HIL results are similar to MIL results, the HIL simulations were carried out properly.

A. EXPERIMENTAL PROCEDURES FOR THE ACTIVE SUSPENSION SYSTEM

For the active suspension system, two different types of scenarios were tested: a filtered step simulating disturbances as the road position profile applied to the model of the active suspension and a Gaussian noise signal simulating the road velocity profile.

The first scenario represents a drastic yet realistic disturbance to the system, that could happen on bad-conditioned roads, for instance. It is stated that a filtered step signal is a more realistic approach than using a regular step function as disturbance input of the system, mainly due to the fact that in real systems it is not possible to apply signals that increase as fast as a theoretical step function. Instead, filtering the signal forces to increase at a more realistic ratio, thus making it a more plausible situation.

TABLE 2. Road profile parameters for the active suspension system simulations.

Road profile	r_1	r_2	r_3
Harsh	1000000	100000	0
Soft	11000	100	0

On the other hand, the second scenario (Gaussian noise disturbance input) is a well-known signal used in literature, such as in [38], to excite the bandwidth of the active suspension system and validate its dynamic behavior.

The active suspension model has different parameters that can be used to determine how much the control system itself is able to modify tire deflection, suspension stroke, and the acceleration of the suspension. The weighting matrices **Q**, **N**, and **R** adopted for the quadratic cost function (11) that is used for formulating the active suspension controller are shown below (12). The variables r_1, r_2, and r_3 correspond to parameters that configure the road profile considered in the analysis. These parameters are divided in three categories: harsh, typical and soft conditions. Each category has a different set of values for r_1, r_2 and r_3. The weighting values used for this work can be found at [38] and are presented in Table II.

$$\begin{aligned}
Q &= \begin{bmatrix} r_1 & 0 & 0 & 0 \\ 0 & (2\xi_2\omega_2)^2 & -2\xi_2\omega_2^3 & -(2\xi_2\omega_2)^2 \\ 0 & -2\xi_2\omega_2^3 & r_2+\omega_2^4 & 2\xi_2\omega_2^3 \\ 0 & -(2\xi_2\omega_2)^2 & 2\xi_2\omega_2^3 & (2\xi_2\omega_2)^2 \end{bmatrix}, \\
N &= \begin{bmatrix} 0 & -2\xi_2\omega_2 & 0 & 2\xi_2\omega_2 \end{bmatrix}, \\
R &= 1 + r_3. \qquad (12)
\end{aligned}$$

Each scenario was tested for each weight category. First, a MATLAB/Simulink simulation was performed (MIL phase), resulting in two different tests: filtered step applied as the system's road disturbance position profile input for harsh and soft weighting control conditions. These conditions were chosen for the test as they reveal how the system behaves when the controller needs to act intensively (harsh conditions) or when it does not need to act so intensively (soft conditions).

After that, HIL simulations using the proposed platform were performed, using the same scenarios and weighting parameters as the ones used for the MIL simulations. The sampling time of the system used for all tests was 0.2 ms, which makes the simulated active suspension system resemble a continuous model when transferred to the target PC during the HIL simulations.

This is the shortest sampling time possible to assign to the suspension model, since this value is limited by the Simulink Real-Time kernel and by the complexity of the model being simulated. The kernel enforces lower and upper bounds, 0.08 ms and 10000 ms, respectively, and sampling times lower than 0.2 ms cause CPU overload, indicating that the CPU is unable to complete processing a model time step before restarting for the next time step, which is caused by the sampling time of the model being too slow.

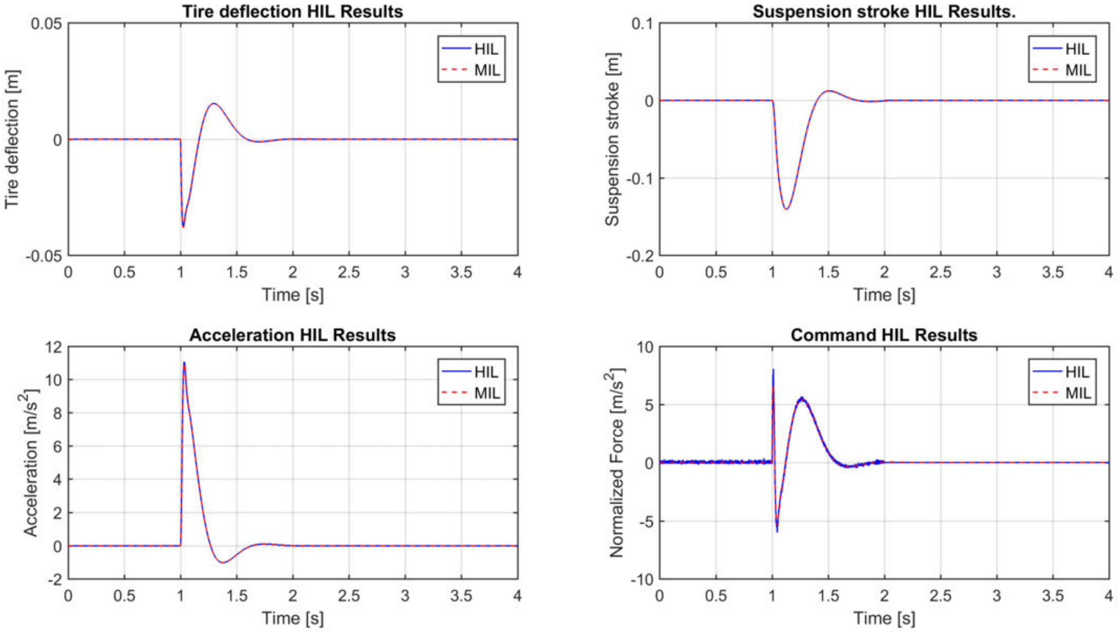

FIGURE 9. HIL and MIL simulations results for a filtered step as the harsh road profile disturbance position applied to the active suspension system.

The same procedures were applied for the second scenario: Gaussian noise applied to the active suspension's road velocity profile disturbance input. MIL simulations were performed using MATLAB/Simulink for each weighting control conditions (harsh and soft), as well as HIL simulations for each case.

B. EXPERIMENTAL PROCEDURES FOR THE ATTITUDE CONTROL OF THE SATELLITE

For the air bearing table system only one scenario is considered: when three step signals are applied to the system, representing the Euler angle (θ_1, θ_2 and θ_3) references. In this scenario, only two parameters are used in order to build the controller: the weighting matrices **Q**, which defines the weights on the states, and **R**, which defines the weights on the control input in the cost function. **N** was considered to be zero. The values used in this simulation are shown below (13):

$$\mathbf{Q} = \begin{bmatrix} 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{R} = \begin{bmatrix} 10^3 & 0 & 0 \\ 0 & 10^3 & 0 \\ 0 & 0 & 10^3 \end{bmatrix},$$

$$\mathbf{N} = 0. \tag{13}$$

MIL Simulations were carried out first in MATLAB/Simulink, while HIL simulations were carried out in the proposed HIL platform, using the same weighting matrices for both simulations and a sampling time of 100 ms. For the same reasons explained earlier, this sampling time was the shortest sampling time possible for the model of the satellite attitude control air bearing table.

VI. RESULTS AND DISCUSSION

A. ACTIVE SUSPENSION

The results for both MIL and HIL simulations can be seen in Fig. 9 - Fig. 12, where HIL simulations are plotted in blue while MIL simulations are plotted in red.

First, analyzing Fig. 9 and Fig. 10, it can be seen that the results for both MIL and HIL simulations using a filtered step function simulating the road position profile were very close. This is a very good result, even though a filtered step is a simple signal for the system to handle: it shows that the platform works accordingly to what is expected when compared to theoretical simulations and all components used for its conception are functioning properly. Thus, this first experiment is an important first step of validation for the HIL platform.

The second result shows different behavior for MIL and HIL simulations during the beginning of the experiments, which can be seen in Fig. 11- Fig. 12. This is a very interesting result because it shows the effect that even small delays may cause on the system when the controller is embedded in real hardware instead of being simulated computationally. These delays occur at the beginning of the simulation because there is a difference of time between the moment Simulink Real-Time starts running the model and the control algorithm is started. This delay is responsible for the initial different behavior between HIL and MIL simulations.

It is important to notice that these results show one of the most important aspects of a HIL simulation: the possibility of

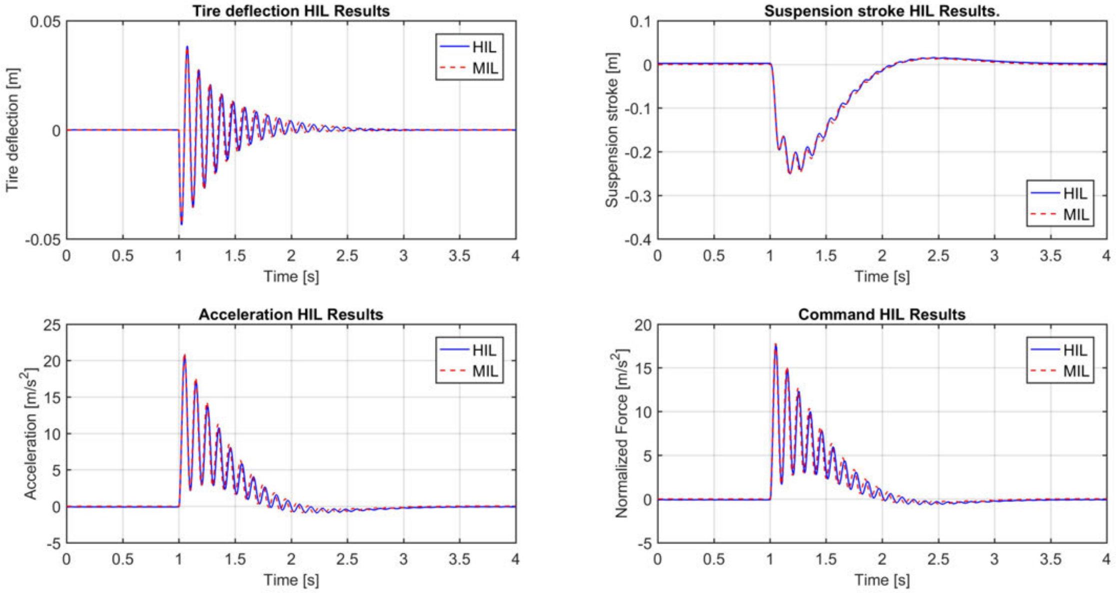

FIGURE 10. HIL and MIL simulations results for a filtered step as the soft road profile disturbance position applied to the active suspension system.

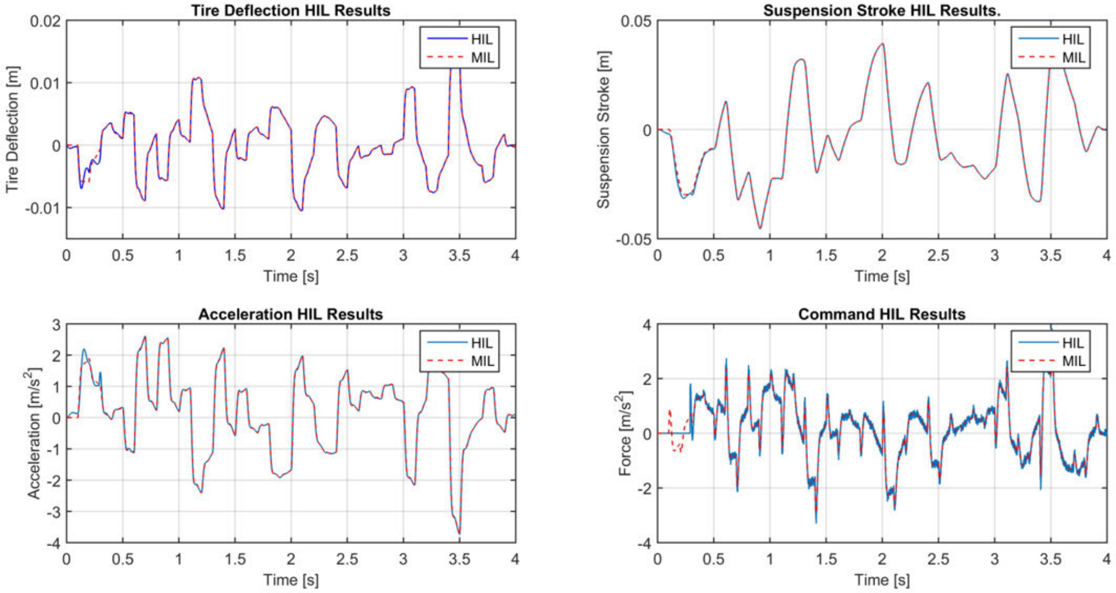

FIGURE 11. HIL and MIL simulations results for a Gausian noise as the harsh road profile disturbance velocity applied to the active suspension system.

observing the dynamic response of a system under more realistic operation conditions. Even though the delays presented on the simulation do not correspond to delays associated with the dynamics of the active suspension system, delays are present in most of the real systems and controllers must be able to deal with them, as long as they do not trespass a certain limit. This is exactly what happens: even though HIL simulation starts quite different than the MIL one, after a certain period the controller is able to deal with the system, making its dynamic behavior resembles the one obtained during MIL simulations.

The effects of delays in HIL simulations is an important topic, as shown in [41] and [42]. These works use HIL simulations to investigate delays of wide-area monitoring and control systems (WAMCS) in smart power grids and remotely connected HIL experiments, respectively, which enforces once again the importance of HIL simulations as a method for obtaining more realistic responses of a system during dynamic tests.

One other important aspect of these results is that when soft control conditions are used, that is a soft ride characteristic, the difference between HIL and MIL results increase and

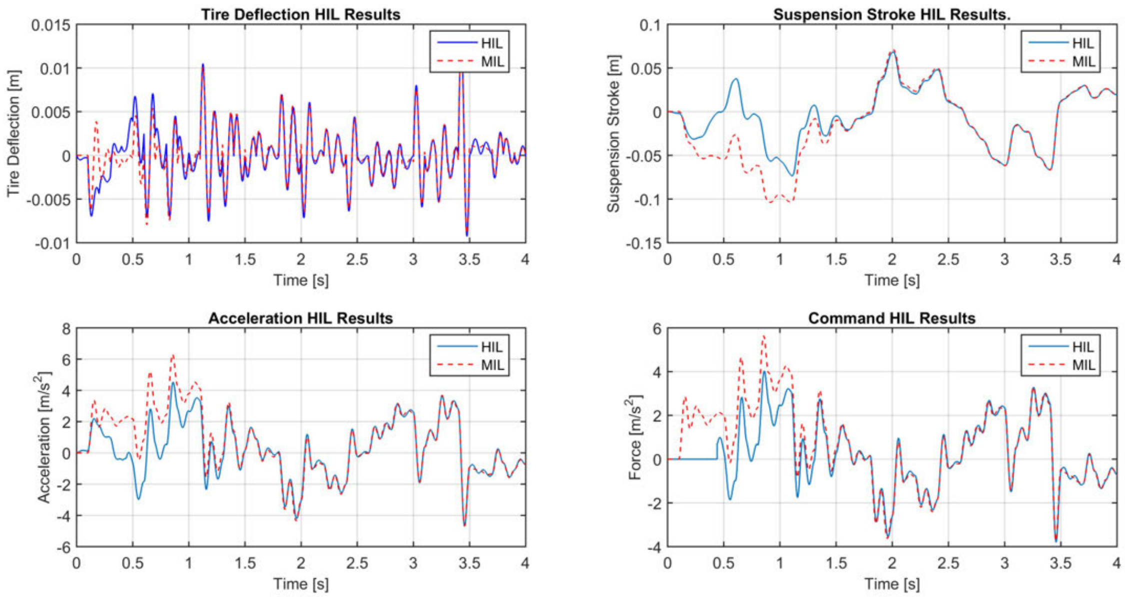

FIGURE 12. HIL and MIL simulations results for a Gaussian noise as the soft road profile disturbance velocity applied to the active suspension system.

it takes more time to obtain similar responses. It does not mean that delays increase in such situation: actually since soft conditions are applied, the control system does not act as strongly as it would under harsh conditions/firmer ride characteristics, dealing with the initial delay present at the HIL simulation slowly compared to the harsh controller.

The second scenario results are very important, since a commonly used type of signal for validating suspension systems (not only active ones) was used: the Gaussian noise signal. It allows not only the validation of the HIL platform itself, but also validates the LQR control strategy embedded in BeagleBone Black, as the embedded controller makes the active suspension system behave as expected, in comparison with MIL results.

Active control must not eliminate completely the disturbance applied to the suspension, namely, a certain level of vibration must exist so that the driver has the perception of the road profile. This can be noticed through the acceleration of the suspension, which after rejecting part of the disturbance introduced into the system, stabilizes gradually, ensuring more comfort to the passenger and thus, fulfilling the control objectives that were established earlier.

B. AIR BEARING TABLE

Fig. 13 - Fig. 15 show the results for the air bearing table after HIL and MIL simulations for the Euler angles, angular velocities and the command variables regarding the reaction wheels acceleration, respectively. As it was the case with the active suspension system results, HIL simulation data is plotted in blue, while MIL simulation data is plotted in red.

The air bearing table model has six state variables, but only references for the Euler angles are established: $\theta_1 = 5°$, $\theta_2 = 2°$ and $\theta_3 = -3°$. The results regarding the Euler angles can be seen in Fig. 13, showing that both MIL and HIL simulations achieve the values indicated by their respective reference signals. Since the sampling time of the air bearing table model is lower than the sampling time of the active suspension system (100 ms compared to 0.2 ms), due to the fact that the air bearing table model possesses more states and control outputs, delays do not impact the response of the HIL simulation and thus, both lines in Fig. 13 remain practically identical throughout the whole simulation.

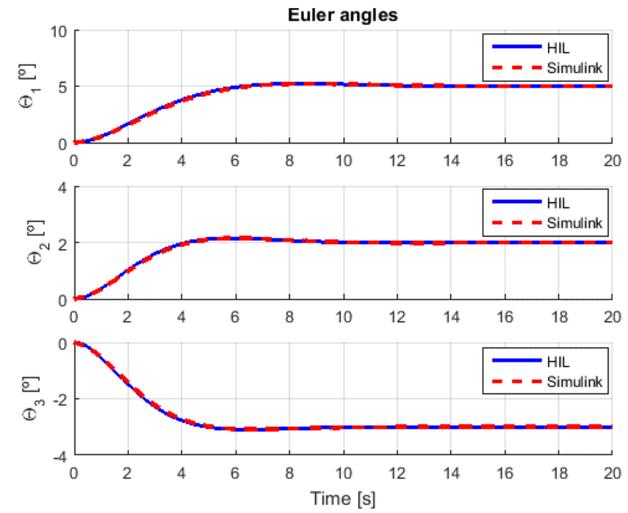

FIGURE 13. HIL and MIL Euler angles results for a filtered step as the input to the air bearing table system.

Fig. 14 shows the angular velocities of the reference system of the body F_b in rad/s. The angular velocities approach zero as the Euler angles approach their values of reference, thus making the angular movement of the satellite remain idle as long as the system remains at this desired point.

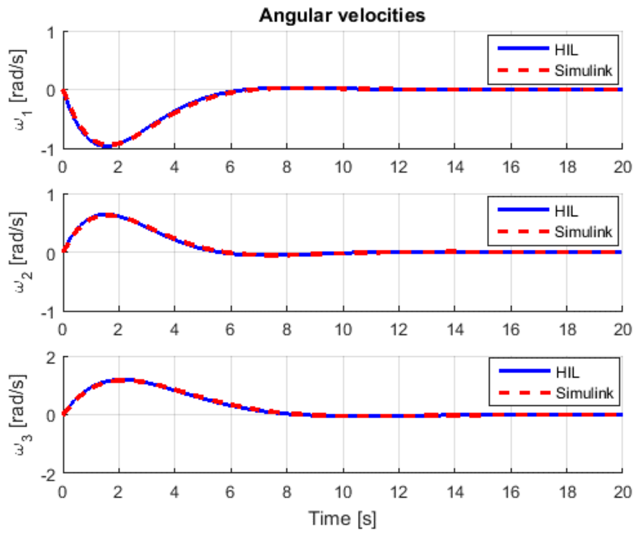

FIGURE 14. HIL and MIL angular velocities results for a filtered step as the input to the air bearing table system.

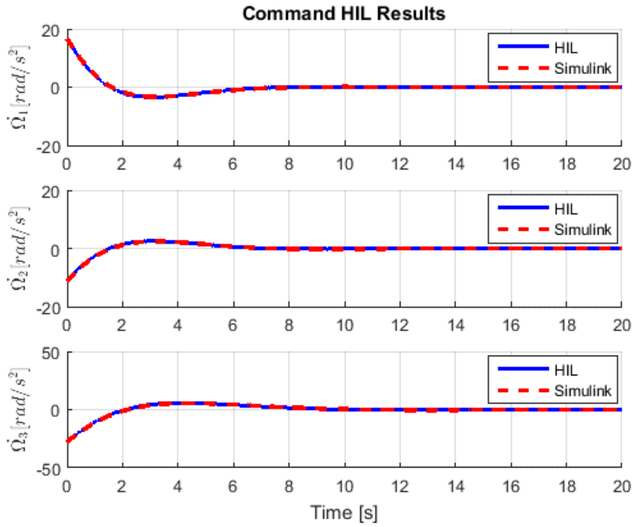

FIGURE 15. HIL and MIL reaction wheels acceleration (command) results for a filtered step as the input to the air bearing table system.

Finally, Fig. 15 includes the three control signals (reaction wheels acceleration) in rad/s^2 generated by the controller, which are sent to the model. Once again, as the Euler angles values reach their reference points, the control variables reach zero, making the satellite remain in the desired position as long as the reference values for the Euler angles remain unchanged.

As can be seen from the results obtained, both MIL and HIL simulations are very close to each other. Once again, they prove the validity of the LQR control strategy embedded in Beaglebone Black and the HIL platform itself. Therefore, the developed platform was capable of performing simulations for an automotive application and an aerospace one, fulfilling the objective of a HIL platform capable of performing simulations for different types of models.

VII. CONCLUSION

HIL simulations are widely used in industry and academia to validate control systems. The importance of this technique motivated the development and construction of the HIL platform presented in this work, that is able to perform simulations with different types of systems, not being restricted to only one type of application.

The developed HIL platform was validated using an active suspension and a satellite attitude control air bearing table system, both controlled using a discrete LQR strategy. The use of MDB also proved to be advantageous, presenting a series of tests (MIL, SIL and HIL) capable of validating each part of the system separately, making it easier to identify and correct errors during the design phase.

It is important to reinforce that the platform proved to be versatile to handle different models, with different numbers of command variables and states. These results show that both the HIL platform and the LQR control strategy embedded in BeagleBone Black are valid, being able to control systems simulated with sampling times of 0.2 ms (active suspension) and 100 ms (air bearing table). Different scenarios of disturbances and control weighting conditions for the active suspension were tested, which increases the credibility of the results presented in this work.

As future work propositions, different control strategies can be developed and validated using the presented HIL platform. This will allow not only the validation of a wider range of control strategies, but also the validation of different types of models, containing varying structures and dynamic behaviors. This will increase the range of applications that can be tested with the platform, thus contributing to the objective of developing a platform capable of performing HIL simulations for many types of systems.

Also, even more complex models of active suspension systems and satellite attitude control air bearing tables can be tested, which would help to validate the HIL platform even further.

REFERENCES

[1] B. Lu, X. Wu, H. Figueroa, and A. Monti, "A low-cost real-time hardware-in-the-loop testing approach of power electronics controls," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 919–931, Apr. 2007.

[2] R. Isermann, J. Schaffnit, and S. Sinsel, "Hardware-in-the-loop simulation for the design and testing of engine-control systems," *Control Eng. Pract.*, vol. 7, no. 5, pp. 643–653, 1999.

[3] S. Röck, "Hardware in the loop simulation of production systems dynamics," *Prod. Eng.*, vol. 5, no. 3, pp. 329–337, 2011.

[4] P. Sarhadi and S. Yousefpour, "State of the art: Hardware in the loop modeling and simulation with its applications in design, development and implementation of system and control software," *Int. J. Dyn. Control*, vol. 3, no. 4, pp. 470–479, 2015.

[5] K. S. Badaruddin, J. C. Hernandez, and J. M. Brown, "The importance of hardware-in-the-loop testing to the Cassini mission to saturn," in *Proc. IEEE Aerosp. Conf.*, Mar. 2007, pp. 1–9.

[6] M. Amin and G. A. A. Aziz, "A hardware-in-the-loop realization of a robust discrete-time current control of PMa-SynRM for aerospace vehicle applications," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 7, no. 2, pp. 936–945, Jun. 2019.

[7] Y. Wu, K. Hu, and X.-M. Sun, "Modeling and control design for quadrotors: A controlled Hamiltonian systems approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 11365–11376, Dec. 2018.

[8] M. Aljehani and M. Inoue, "Performance evaluation of multi-UAV system in post-disaster application: Validated by HITL simulator," *IEEE Access*, vol. 7, pp. 64386–64400, May 2019.

[9] A. S. Abdelrahman, K. S. Algarny, and M. Z. Youssef, "A novel platform for powertrain modeling of electric cars with experimental validation using real-time hardware in the loop (HIL): A case study of GM second generation chevrolet volt," *IEEE Trans. Power Electron.*, vol. 33, no. 11, pp. 9762–9771, Nov. 2018.

[10] L. Li, Y. Lu, R. Wang, and J. Chen, "A three-dimensional dynamics control framework of vehicle lateral stability and rollover prevention via active braking with MPC," *IEEE Trans. Ind. Electron.*, vol. 64, no. 4, pp. 3389–3401, Apr. 2017.

[11] V. V. Lopez, J. M. E. Mejia, and D. E. C. Dominguez, "Design and statistical validation of spark ignition engine electronic control unit for hardware-in-the-loop testing," *IEEE Latin Amer. Trans.*, vol. 15, no. 8, pp. 1376–1383, Aug. 2017.

[12] C. Lv, H. Wang, and D. Cao, "High-precision hydraulic pressure control based on linear pressure-drop modulation in valve critical equilibrium state," *IEEE Trans. Ind. Electron.*, vol. 64, no. 10, pp. 7984–7993, Oct. 2017.

[13] X. Yang, C. Yang, T. Peng, Z. Chen, B. Liu, and W. Gui, "Hardware-in-the-loop fault injection for traction control system," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 6, no. 2, pp. 696–706, Jun. 2018.

[14] R. Morello, R. Di Rienzo, R. Roncella, R. Saletti, and F. Baronti, "Hardware-in-the-loop platform for assessing battery state estimators in electric vehicles," *IEEE Access*, vol. 6, pp. 68210–68220, 2018.

[15] P. Kotsampopoulos, D. Lagos, N. Hatziargyriou, M. O. Faruque, G. Lauss, O. Nzimako, P. Forsyth, M. Steurer, F. Ponci, A. Monti, V. Dinavahi, and K. Strunz, "A benchmark system for hardware-in-the-loop testing of distributed energy resources," *IEEE Power Energy Technol. Syst. J.*, vol. 5, no. 3, pp. 94–103, Sep. 2018.

[16] G. Lauss and K. Strunz, "Multirate partitioning interface for enhanced stability of power hardware-in-the-loop real-time simulation," *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 595–605, Jan. 2019.

[17] A. Fernández-Álvarez, M. Portela-García, M. García-Valderas, and J. López, "HW/SW co-simulation system for enhancing hardware-in-the-loop of power converter digital controllers," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 5, no. 4, pp. 1779–1786, Dec. 2017.

[18] N. D. Marks, W. Y. Kong, and D. S. Birt, "Stability of a switched mode power amplifier interface for power hardware-in-the-loop," *IEEE Trans. Ind. Electron.*, vol. 65, no. 11, pp. 8445–8454, Nov. 2018.

[19] T. Roinila, T. Messo, R. Luhtala, R. Scharrenberg, E. C. W. de Jong, A. Fabian, and Y. Sun, "Hardware-in-the-loop methods for real-time frequency-response measurements of on-board power distribution systems," *IEEE Trans. Ind. Electron.*, vol. 66, no. 7, pp. 5769–5777, Jul. 2019.

[20] C. Qi, F. Gao, X. Zhao, Q. Wang, and Q. Sun, "Distortion compensation for a robotic hardware-in-the-loop contact simulator," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 4, pp. 1170–1179, Jul. 2018.

[21] L. Piardi, L. Eckert, J. Lima, P. Costat, A. Valente, and A. Nakano, "3D Simulator with Hardware-in-the-Loop capability for the Micromouse Competition," in *Proc. IEEE Int. Conf. Auto. Robot Syst. Competition (ICARSC)*, Apr. 2019, pp. 1–6.

[22] J. Kölsch, C. Heinz, S. Schumb, and C. Grimm, "Hardware-in-the-loop simulation for Internet of Things scenarios," in *Proc. Workshop Modeling Simulation Cyber-Phys. Energy Syst. (MSCPES)*, Apr. 2018, pp. 1–6.

[23] R. Matinnejad, S. Nejati, L. Briand, T. Bruckmann, and C. Poull, "Search-based automated testing of continuous controllers: Framework, tool support, and case studies," *Inf. Softw. Technol.*, vol. 57, pp. 705–722, Jan. 2015.

[24] O. L. Osen, "On the use of hardware-in-the-loop for teaching automation engineering," in *Proc. IEEE Global Eng. Educ. Conf. (EDUCON)*, Apr. 2019, pp. 1308–1315.

[25] Y. Yonezawa, H. Nakao, and Y. Nakashima, "Novel hardware-in-the-loop simulation (HILS) technology for virtual testing of a power supply," in *Proc. IEEE Appl. Power Electron. Conf. Expo. (APEC)*, Mar. 2018, pp. 2947–2951.

[26] D. Zhu, E. G. D. Pritchard, and L. M. Silverberg, "A new system development framework driven by a model-based testing approach bridged by information flow," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2917–2924, Sep. 2018.

[27] T. Kelemenová, M. Kelemen, L. Miková, V. Maxim, E. Prada, T. Lipták, and F. Menda, "Model based design and HIL simulations," *Amer. J. Mech. Eng.*, vol. 1, no. 7, pp. 276–281, 2013.

[28] C. Liu, R. Ma, B. Hao, H. Luo, F. Gao, and F. Gechter, "FPGA based hardware in the loop test of railway traction system," in *Proc. IEEE Int. Conf. Ind. Electron. Sustain. Energy Syst. (IESES)*, Jan./Feb. 2018, pp. 206–211.

[29] S. Mojlish, N. Erdogan, D. Levine, and A. Davoudi, "Review of hardware platforms for real-time simulation of electric machines," *IEEE Trans. Transp. Electrific.*, vol. 3, no. 1, pp. 130–146, Mar. 2017.

[30] S. Lahti, P. Sjövall, J. Vanne, T. D. Hämäläinen, "Are we there yet? A study on the state of high-level synthesis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 5, pp. 898–911, May 2019.

[31] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 5th ed. Boston, MA, USA: Pearson, 2010.

[32] I. H. B. Pizetta, A. S. Brandao, and M. Sarcinelli-Filho, "A hardware-in-the-loop platform for rotary-wing unmanned aerial vehicles," *J. Intell. Robot. Syst.*, vol. 84, pp. 725–743, Dec. 2016.

[33] S. R. B. dos Santos and N. M. F. de Oliveira, "Longitudinal autopilot controllers test platform hardware in the loop," in *Proc. Int. Syst. Conf. (SysCon)*, Apr. 2011, pp. 379–386.

[34] M. Mătăsaru, "Synchronisation of two DC servo SRV-02ET motors using xPC Target," in *Proc. 17th Int. Conf. Syst. Theory, Control Comput. (ICSTCC)*, Oct. 2013, pp. 337–342.

[35] R. N. Jazar, *Vehicle Dynamics: Theory and Application*. New York, NY, USA: Springer, 2013.

[36] J. J. Rath, M. Defoort, H. R. Karimi, and K. C. Veluvolu, "Output feedback active suspension control with higher order terminal sliding mode," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1392–1403, Feb. 2017.

[37] M. Yu, C. Arana, S. A. Evangelou, and D. Dini, "Quarter-car experimental study for series active variable geometry suspension," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 2, pp. 743–759, Mar. 2019.

[38] A. G. Ulsoy, H. Peng, and M. Çakmakci, *Automotive Control Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[39] R. G. Gonzales, "Utilização dos métodos SDRE e Filtro de Kalman para o controle de atitude de simuladores de satélites," M.S. thesis, Dept. Space Mech. Control, Inst. Nacional de Pesquisas Espaciais, São José dos Campos, Brazil, 2009.

[40] G. R. G. da Silva, A. S. Bazanella, C. Lorenzini, and L. Campestrini, "Data-driven LQR control design," *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 180–185, Jan. 2019.

[41] A. S. Musleh, S. M. Muyeen, A. Al-Durra, I. Kamwa, M. A. S. Masoum, and S. Islam, "Time-delay analysis of wide-area voltage control considering smart grid contingences in a real-time environment," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1242–1252, Mar. 2018.

[42] J. L. Cale, B. B. Johnson, E. Dall'Anese, P. M. Young, G. Duggan, P. A. Bedge, D. Zimmerle, and L. Holton, "Mitigating communication delays in remotely connected hardware-in-the-loop experiments," *IEEE Trans. Ind. Electron.*, vol. 65, no. 12, pp. 9739–9748, Dec. 2018.

ALCEU BERNARDES CASTANHEIRA DE FARIAS received the degree in electronic engineering and the master's degree in mechatronic systems from the University of Brasília (UnB), in 2016 and 2019, respectively. His research interests include control systems, embedded systems, and reconfigurable systems using FPGA.

REURISON SILVA RODRIGUES received the B.S. degree in mechanical engineering from the Federal University of Pernambuco (UFPE), in 2014, and the M.S. degree in mechatronic systems from the University of Brasília (UnB), in 2018. His research interests include embedded systems and automatic control.

ANDRÉ MURILO received the degree in mechatronics engineering from the Pontifical Catholic University of Minas Gerais, in 2001, and the master's and Ph.D. degrees in automatic control from the Institut National Polytechique de Grenoble (INPG), France, in 2006 and 2009, respectively. He is currently an Associate Professor with the Faculty of Gama, University of Brasília, Brazil. He has experience in predictive control, nonlinear control, automotive and aerospace control systems, robotics, mechatronics, and industrial automation.

RENATO VILELA LOPES received the B.Sc. degree in electrical engineering from São Paulo State University (UNESP), Ilha Solteira, Brazil, in 2004, the master's degree in automated systems and control from the Instituto Tecnológico de Aeronáutica, São José dos Campos, Brazil, in 2006, and the Ph.D. degree in hybrid systems identification from the University of Brasília (UnB), Brazil, in 2014. In 2011, he joined the UnB Gama College, UnB, where he is currently an Associate Professor. His research interests include control theory and applications, systems identification, hybrid systems, and estimation and nonlinear filtering.

SUZANA AVILA received the degree in civil engineering and the master's degree in structural engineering from the University of Brasília, in 1993 and 1997, respectively, and the Ph.D. degree in civil engineering from the Pontifical Catholic University of Rio de Janeiro, in 2002. She has experience in civil engineering, focusing on structural mechanics, acting on the following subjects: structural dynamics, vibration control, tuned mass damper, hybrid control, and nonlinear dynamics.