24th COBEM - 2017

Associação Brasileira de Engenharia e Ciências Mecânicas

24th ABCM International Congress of Mechanical Engineering
December 3-8, 2017, Curitiba, PR, Brazil

COBEM-2017-1104
EMBEDDED CONTROL STRATEGY FOR AN ACTIVE SUSPENSION USING A LOW-COST HARDWARE IN THE LOOP PLATFORM

Alceu Farias
Reurison Rodrigues
André Murilo
Renato Lopes
Suzana Ávila
University of Brasilia, Engineering Faculty campus Gama, DF, Brazil
alceucastanheira@aluno.unb.br, reurison@gmail.com, andremurilo@unb.br, rvlopes@unb.br, avilas@unb.br

Abstract. *Through the last decades, vehicles have become more and more complex. As time passes, they are composed by even more electromechanical systems, which include many automotive control units. One essential part of a vehicle is the automotive suspension, responsible not only for supporting the weight of the vehicle itself and attenuating vibrations, but also for being an indicator of the overall performance of the car. Consequently, many studies have been performed throughout these years to improve the performance of automotive suspensions. One field of study that has gained great notoriety in this scenario is the usage of control systems to make it possible for the automotive suspension to adapt and perform in an optimized way according to the road profile characteristics, creating the so-called active suspensions. This work presents the development of an embedded LQR control strategy applied to an active suspension based on a quarter-car model, used for improving the comfort of the passenger, as well as reducing suspension travel and improving vehicle driving, while maintaining good contact between tire and road. In order to validate such system, both Model in the Loop (MIL) and Hardware in the Loop (HIL) simulations are performed, with the latter one utilizing a low-cost HIL platform developed at the University of Brasilia. Thus, this works serves as validation for both the embedded control strategy developed and the HIL platform itself.*

Keywords: *Active Suspensions, Hardware in the Loop, Control Systems, Mechatronics, Automation.*

1. INTRODUCTION

Due to technology advancements regarding computing, sensors and control areas, automotive industry has evolved fast, increasingly turning vehicles into complex electromechanical systems (Ulsoy *et al.*, 2012). Today, vehicles are composed of hierarchically distributed systems and embedded control systems, which allow automobiles to develop in a more intelligent and automated way. Because of that, control systems have been widely used on automotive industry, ranging from suspension to engine, steering and gear control systems. Among these topics, the study of suspension systems has gained importance over the years, due to its influence on the vehicle good performance (Kiencke and Nielsen, 2005).

An automotive suspension system supports the weight of the vehicle on its axles, attenuates vibrations from the excitation of the road and ensures the contact of the tires with the ground. It is typically composed of elastic elements (springs), dissipative devices (dampers) and connections that link the body of the vehicle to the wheels, allowing the chassis to be isolated from the vibrations caused by the irregular terrain profiles (Savaresi *et al.*, 2010).

The introduction of control systems in suspension design aims to improve suspension performance. The so-called passive suspensions, for example, perform satisfactorily under the conditions of the specific terrain profile considered in their design, but may cause discomfort to the occupants of the vehicle for different operating conditions. Studies show that passive suspensions in general can only achieve good results through a trade-off: an improvement in one characteristic (tire deflection, stroke or acceleration of the suspension, for example) is only possible with the detriment of another characteristic of the system (Rajamani, 2011).

From that point on, there has been an increasingly interest on suspension systems capable of having its parameters modified in real time, so that they would be able to adapt to the characteristics of the current road profile. These types of suspension systems are known as semi-active or active suspensions, depending if the control system parameters can be partially or fully modified, respectively. Studies of intelligent (semi-active and active) suspensions have been developed over the last decades and can be found in literature, e.g., (Redfield, 1991), (Cao *et al.*, 2008), (Eltantawie, 2012) and (Huang *et al.*, 2017), which allowed this technology to develop even more. However, it has not become the most popular type of suspension on the market, being restricted to low frequency bandwidth systems. This is mainly due to the considerable amount of external energy required in order to make it work properly, which is generally responsible for increasing its implementation costs (Ulsoy *et al.*, 2012).

Once automotive control systems have gained such importance, it becomes necessary to obtain effective methodologies for validating such systems and improving their overall performance. One technique that has been widely used for this purpose is known as Hardware in the Loop (HIL), which is characterized by the connection of real and computational simulated parts of a system, making it possible to analyse its behaviour under more realistic operation conditions, without the need of building the system itself or a prototype for it (Isermann *et al.*, 1999). This allows improvement of both time and costs of the application, which makes it a very popular technique used in industry.

Figure 1 represents the general architecture of a HIL simulation, that changes according to the decision of which are the real and simulated components of the system being tested.

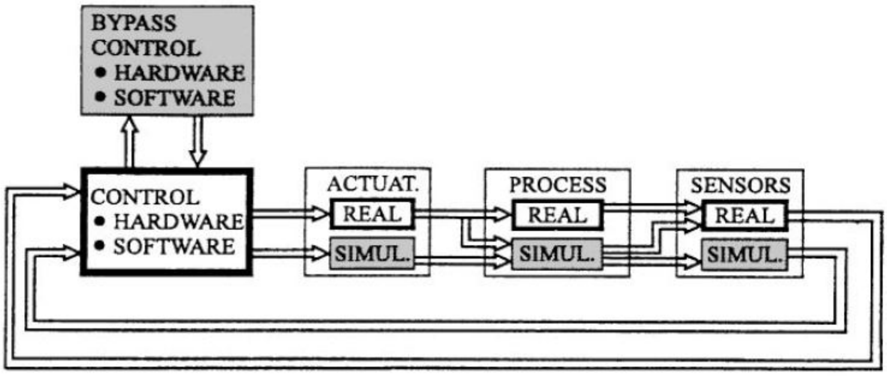


Figure 1: General architecture of a HIL simulation (Isermann *et al.*, 1999).

According to (Palladino *et al.*, 2012), HIL simulations are recognized as an effective methodology in testing control strategies and diagnostic functionalities, due to its compatibility in replacing significant portions of test procedures for different control systems as well as incorporating hardware components in a numerical simulation environment, yielding to results with better credibility than pure numerical simulations. This explains the reason why many works use HIL simulations on the automotive industry literature, ranging from the evaluation of motor characteristics for hybrid electric vehicles (Oh, 2005), testing and evaluating control strategies for actuators on a car (Gawthrop *et al.*, 2008), testing and diagnosting funcionalities on Electronic Control Units (ECUs) (Palladino *et al.*, 2012), testing multi-vehicle autonomous systems (Pannocchi *et al.*, 2017), automotive suspension systems, which is the main field of interest for this work, (Lozoya-Santos *et al.*, 2012) and (Jian *et al.*, 2015) and many others.

Regarding this scenario and the importance of HIL simulations, a low-cost HIL platform was developed at the University of Brasilia, so that it could be used for different types of application, unlike most commercial models available at the market. This work aims to use this HIL platform to test a LQR control strategy developed for an active suspension system based on a quarter-car model. As stated before, automotive applications use HIL simulations as an industrial standard approach for validating control systems and, thus, it makes a great test for the HIL platform developed.

The system is modeled on MATLAB/Simulink and simulated using Simulink Real-Time. The LQR control strategy was chosen mainly due to its capability of performing optimization operations in order to control a system while taking in consideration the restrictions of a given cost function, which is responsible for imposing the suspension system limitations (Ohri *et al.*, 2014), is embedded on a microprocessor, making it possible to test the system under more realistic operation conditions rather than performing purely computational simulations on Simulink.

This work is divided as it follows: section 2 introduces the mathematical modelling of both the active suspension system and the LQR controller. Section 3 presents an overview of the structure and architecture of the developed HIL platform and how it works. Section 4 contains the methodology used for performing HIL simulations and validating the control strategy developed. Section 5 presents and discusses the results obtained for this work. Finally, Section 6 concludes this work, with its final considerations and proposed future works.

2. SYSTEM MODELLING

2.1 ACTIVE SUSPENSION MODELLING

The automotive suspension model discussed in this paper is based on a quarter car vehicle. According to (Ulsoy *et al.*, 2012), an example of this type of system is shown in Fig. 2. The main parameters used to model the active suspension are: 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_s) and active suspension force (f).

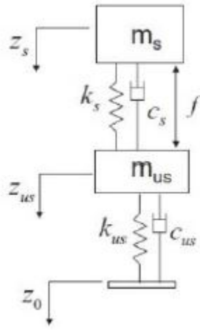


Figure 2: Quarter car suspension model with actuator (Ulsoy *et al.*, 2012).

The equations of motion that describe the dynamic behavior of the suspension are given by Eq.1 and Eq.2:

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

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

The system state variables are given by the tire deflection ($z_{us} - z_0$ or x_1), velocity of the unsprung mass (z_{us} or x_2), suspension travel ($z_s - z_{us}$ or x_3) and, finally, 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 related to the ground and the position of the sprung mass related to the unsprung mass, respectively. They are also considered to be important system outputs: tire deflection and suspension course, y_1 and y_2. In addition, there is a third variable of interest in the system, the suspension acceleration, given by $\ddot{z}_s$ or y_3. The control variable of the system will be the active suspension force normalized by the sprung mass of the system($u = f/m_s$), which is given in m/s^2, and the perturbation of the system is given by the road excitation velocity ($\dot{z}_0$).

These equations can be expressed in matrix form, allowing the representation of the active suspension model in state space in continuous form (Eq.3 and Eq.4), so that $\boldsymbol{A}$ is the state matrix, $\mathbf{x}$ is the state vector, $\boldsymbol{B}$ is the matrix of inputs , $\mathbf{u}$ is the input vector, $\boldsymbol{G}$ is the input perturbation matrix, $\mathbf{w}$ is the perturbation itself, $\boldsymbol{C}$ is the output matrix, and $\boldsymbol{D}$ is the direct transmission matrix. In this work, the perturbation $\mathbf{w}$ corresponds to the excitation velocity of the road, or z_0. Substituting Eq.1 and Eq.2 into Eq.3 and Eq.4, and normalizing the state variables according to the parameters determined by Eq.5, we obtain the matrices that characterize the system in state space (Eq. 6).

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} + \mathbf{Gw} \tag{3}$$

$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du} \tag{4}$$

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

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

The dimensionless parameters ρ, ξ_1 and ξ_2 correspond to the radio of sprung mass to unsprung mass, unsprung mass damping factor and sprung mass damping factor, respectively. Parameters ω_1 and ω_2 are the natural frequencies of unsprung and sprung mass, respectively. For this work, all 4 outputs of the system are considered to be measured (x_1, x_2, x_3 and x_4) and, therefore, $\boldsymbol{C}$ is an identity matrix of dimensions 4x4. $\boldsymbol{D}$ is considered 0, as there is no direct transmission from inputs to outputs of the system.

2.2 CONTROL SYSTEM MODELLING

One of the most important criteria for evaluating the good performance of a suspension system is the vehicle occupant's perception of the vibration within the vehicle, which depends not only on the suspension itself but also on the human body's response to vibration. Although it is desired to reduce the vibration of the vehicle, the suspension system must allow a certain level of vibration, allowing the passenger to have a certain perception of the road profile. Therefore, the main objectives of the controller are: to improve passenger comfort, reduce suspension travel and improve vehicle driving, maintaining good contact between tire and road (Ulsoy *et al.*, 2012).

The control strategy presented in this work is the Linear Quadratic Regulator (LQR). It is a widely used method in the literature, since it offers a very systematic way of calculating the feedback gain matrix of each state, while producing a stable system at the end of the process, mostly (Ohri *et al.*, 2014). The purpose of this work is to find a control variables vector that reduce the system energy, minimizing a quadratic performance index.

Considering the LQR applied to the automotive suspension, discrete control equations are used, so that the controller can be implemented in a microprocessor and validated with the HIL simulation. The weighing matrices adopted for the quadratic cost function J (Eq.7) are taken from (Ulsoy *et al.*, 2012) and shown in Eq.8. The index k indicates the current time instant and the index rms indicates the root mean square value of the state variables ($\mathbf{x_{rms}}$) and control variables ($\mathbf{u_{rms}}$). The variables r_1, r_2 and r_3 correspond to parameters that configure the road profile considered in the analysis.

$$J(u) = \mathbf{x^T_{rms}}[k]\mathbf{Q}\mathbf{x_{rms}}[k] + \mathbf{u^T_{rms}}[k]R\mathbf{u_{rms}}[k] + 2\mathbf{x^T_{rms}}[k]\mathbf{N}\mathbf{u}[k] \tag{7}$$

$$\mathbf{Q} = \begin{bmatrix} r_1 & 0 & 0 & 0 \\ 0 & (2\xi_2\omega_2)^2 & -2\xi_2\omega_2^3 & -(2\xi_2\omega_2)^2 \\ 0 & -2\xi_2\omega_2^3 & r_2+\omega_2^4 & 2\xi_2\omega_2^3 \\ 0 & -(2\xi_2\omega_2)^2 & 2\xi_2\omega_2^3 & (2\xi_2\omega_2)^2 \end{bmatrix}, \quad \mathbf{N} = \begin{bmatrix} 0 \\ -2\xi_2\omega_2 \\ 0 \\ 2\xi_2\omega_2 \end{bmatrix}, R = 1 + r_3 \tag{8}$$

The discrete LQR control law, can be understand as a simple state feedback for each time step *k*, as shown below (Eq.9). The matrix feedback gain ***K*** can be calculated according to Eq.10, which depends on the value of the Ricatti ***S*** matrix obtained by solving the discrete Ricatti equation (Eq.11). $\mathbf{A_d}$ and $\mathbf{B_d}$ represent the state matrices and discretized inputs, respectively.

$$\mathbf{u}[k] = -\mathbf{K}\mathbf{x}[k] \tag{9}$$

$$\mathbf{K} = (\mathbf{B_d}^T\mathbf{S}\mathbf{B_d} + \mathbf{R})^{-1}(\mathbf{B_d}^T\mathbf{S}\mathbf{A_d} + \mathbf{N}^T) \tag{10}$$

$$\mathbf{A_d}^T\mathbf{S}\mathbf{A_d} - \mathbf{S} - (\mathbf{A_d}^T\mathbf{S}\mathbf{B_d} + \mathbf{N})(\mathbf{B_d}^T\mathbf{S}\mathbf{B_d} + \mathbf{R})^{-1}(\mathbf{B_d}^T\mathbf{S}\mathbf{A_d} + \mathbf{N}^T) + \mathbf{Q} = 0 \tag{11}$$

3. HIL PLATFORM DESIGN AND IMPLEMENTATION

Many HIL simulation platforms found in literature are composed by three main components: a host computer responsible for simulating a part of the system, a device responsible for implementing another part of the system in real hardware (e.g., a microprocessor) and signal conditioners that allow communication between host and target computers. The HIL platform developed at the University of Brasilia has a very similar structure, shown in Fig. 3. It contains a host and a target computer (dashed blue line), responsible for creating and simulating in real time the model to be controlled, a BeagleBone Black microprocessor, that contains the embedded control strategy to be validated, and a PCI-DAC 6703 data acquisiton system, that allows communication between model and control systems.

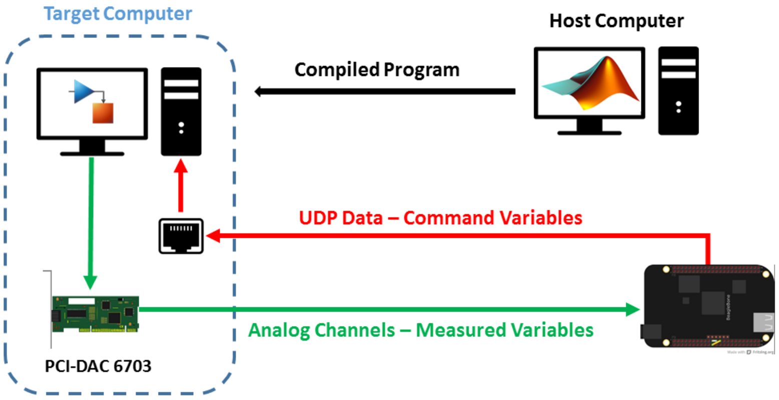

Target Computer
Host Computer
Compiled Program
UDP Data – Command Variables
Analog Channels – Measured Variables
PCI-DAC 6703

Figure 3: Architecture of the HIL platform developed at the University of Brasilia.

The host corresponds to the computer used for creating the model using the Simulink environment on MATLAB and then loading the application to the target computer, which runs the Simulink Real-Time kernel. The Simulink Real-Time toolbox allows the creation of real-time applications from Simulink models, which allows them to run on dedicated target computer hardware.

The communication interfaces present on the target computer are a PCI-DAC 6703 and an Ethernet port as shown in Fig. 3.

The data acquisition system PCI-DAC 6703 is responsible for transmitting information from the model state variables to the control algorithm present in the microprocessor (via analog channels represented by the green lines). Thus, the data acquisition works as one of the signal conditioners of the HIL platform, allowing communication between the simulated model state variables and the control system, embedded in hardware.

A Beaglebone Black Rev C is used for the embedded control system of the suspension. It is a low-power open-source hardware single-board computer with 512 MB RAM, 1 GHz processor clock and 4 GB of eMMC flash memory, having the size of a credit card. The Beaglebone is able to run an embedded GNU/Linux distribution, which allows the user to connect several external peripherals. It is also responsible for executing the control algorithm embedded in it and transmitting the control variables via an UDP communication protocol, using an Ethernet cable (red line in Fig. 3) to connect to the suspension model being simulated on the target computer.

UDP is a commonly used transport layer because of its lightweight nature. When used with Simulink Real-Time, UDP gives the application a good chance of succeeding in real-time execution. Also, the datagram nature of UDP is optimal for sending samples of data from the real-time application generated by the Simulink Coder software (Mathworks, 2017).

It is also suitable for purposes where error checking and correction are either not necessary or are performed in the application. 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 (Kurose and Ross, 2010).

It is known that soft real-time applications are delay sensitive but can tolerate isolated errors and losses. Usually, soft real-time focuses on mean times and tries to reduce deadline miss ratios (Watteyne and Auge-Blum, 2005). Thus, the HIL simulation performed by the platform allows validation of the system under more realistic operating conditions without the need to build a prototype or even the final real system, once it is possible to take into account delays on the system, which is usually not the case when performing only purely computational simulations.

Therefore, the UDP protocol via Ethernet cables is another signal conditioner present on the platform, allowing command variables to be sent from the control system embedded on the BeagleBone Black microprocessor to the model being simulated on the target computer.

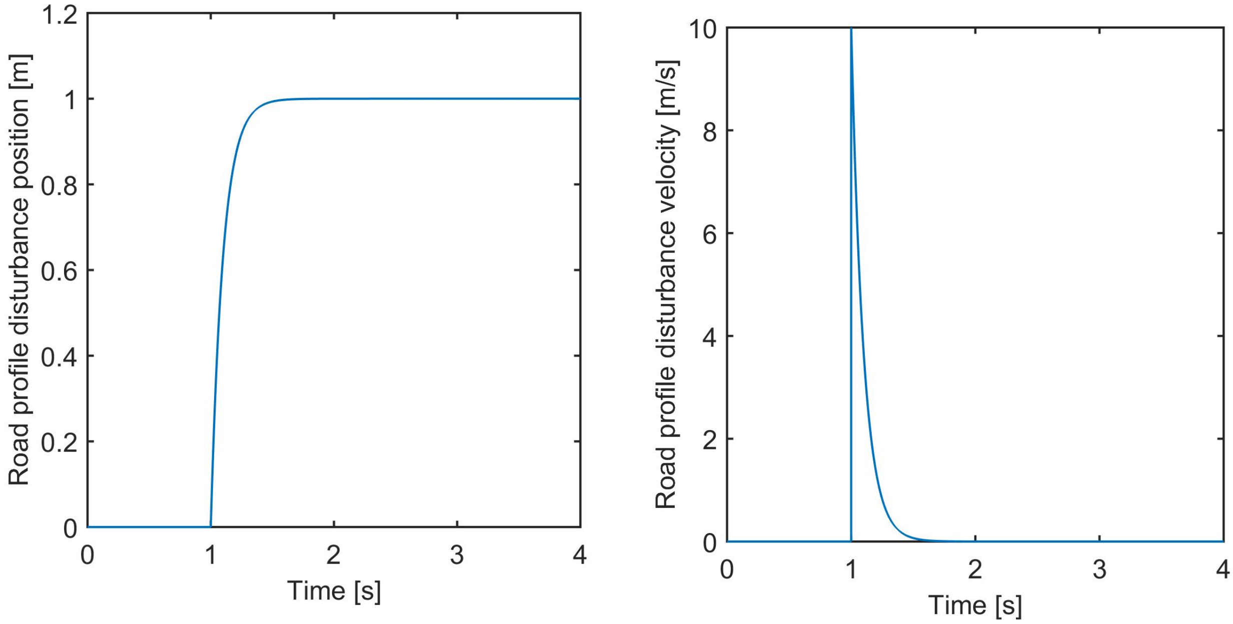
All these components are responsible for closing the loop and allowing the HIL simulations to be performed properly.

4. METHODOLOGY

The procedure for performing an appropriate HIL simulation using the platform presented on this work consists of three steps: testing and simulation of the continuous system, discretization of the system and finally, the HIL simulation properly stated.

First, the continuous active suspension model and the LQR controller are developed and simulated in MATLAB/Simulink, through purely computational simulations. This type of simulation is commonly known as Model in the Loop (MIL), which is performed without any real component of the system involved on this stage of testing (Matinnejad *et al.*, 2015).

The perturbation of the system is given by $\dot{z}_0$, i.e the speed of excitation of the road in the suspension. Two scenarios were tested on this work, representing two different types of perturbation for the active suspension system. First, a filtered pulse was used as a road profile disturbance velocity signal, which is equivalent to an unitary filtered step as a road profile disturbance position (z_0). This choice was made so as to represent an abrupt excitation, of elevated magnitude, to see if the suspension is capable of absorbing. The filtered step disturbance signal is shown in Fig.4



(a) First test scenario road profile disturbance position (filtered step).

(b) First test scenario road profile disturbance velocity.

Figure 4: First test scenario road profile disturbance position (filtered step) and velocity signals.

The step signal is filtered in order to simulate a more realistic kind of disturbance signal: even though the intention is to represent an abrupt excitation to see how the active suspension system responds, in real applications, it is not possible to introduce a theoretical step signal into the system, once it reaches its final value much faster than real disturbances step signals produced by any road profile. Thus, the filtered step signal represents more realistic operations conditions for the simulation intended.

The second scenario corresponds to the application of a Gaussian noise disturbance signal with zero mean and variance of $2\pi A u_0$, where $A = 5.0292\text{x}10^{-6}$ m is the ground-motion amplitude and $u_0 = 24.384$ m/s is the vehicle-forward velocity. According to (Ulsoy *et. al.*, 2012), this is a widely used signal for testing active suspension systems, as it is made for exciting different bandwidth values of these type of systems, making it possible to obtain a more complete frequency response for them. It is applied directly as the road profile velocity of excitation input of the system, which is shown in Fig.5.

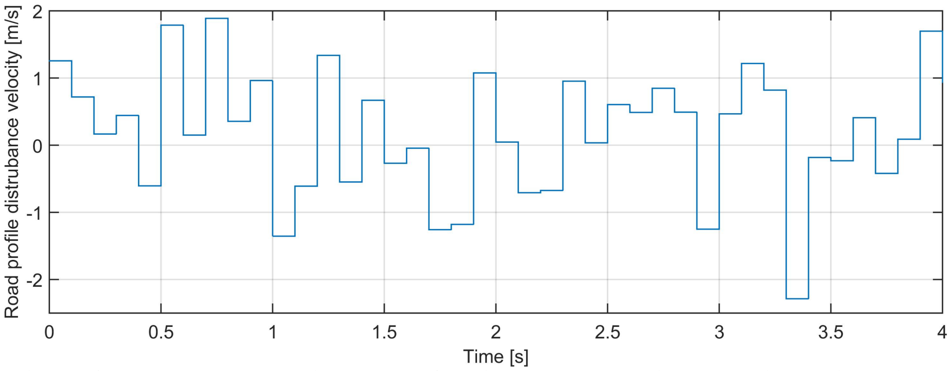


Figure 5: Second test scenario road profile disturbance velocity (Gaussian white noise).

It is also important to notice that the sampling time for the Gaussian noise disturbance signal used was 0.1s, in order to make it easier to analyse the results (once noises with higher sampling rate would result in no meaningful noisy curves during simulation) and make it closer to a real signal.

Both scenarios were tested and validated for the continuous active suspension model. The values of $\boldsymbol{K}$ were calculated using MATLAB.

The values of the constants used for the active suspension are shown in Tab. 1.

Table 1: Active suspension model constants (Ulsoy *et al.*, 2012).

Parameter	Physical meaning	Value
$\rho^{(1)}$	Ratio of sprung mass to unsprung mass	10.0
ω_1	Unsprung mass natural frequency	20π rad/s
ω_2	Sprung mass natural frequency	2π rad/s
$\xi_1^{(1)}$	Unsprung mass damping factor	0.0
$\xi_2^{(1)}$	Sprung mass damping factor	0.3

$^{(1)}$ Dimensionless parameter

The environment is simulated using typical values of weighing parameters, which corresponds to r_1 = 50000, r_2 = 5000 and r_3 = 0 (Ulsoy *et al.*, 2012).

Once this step is over, the active suspension model was discretized in MATLAB, as well the LQR controller, based on Eq.9, Eq.10 and Eq11. This step is of great importance because it is necessary for the controller to be able to be embedded in hardware later for the HIL simulation.

For the discretization process, a sampling period of 200μs was adopted, which is the shortest sampling time possible for this model. The sampling time that is possible to assign to the suspension model is limited by the Simulink Real Time kernel and by the complexity of the model itself. The kernel enforces lower and upper bounds, 8μs and 10s, respectively. A time less than 200μs causes CPU overload, which indicates that the CPU is unable to complete processing the model time step before restarting for the next time step. Possible reasons are that either the target computer is too slow or the model sampling time is too small. Another possible reason would be that the model is too complex (algorithmic complexity), which does not seem to be the case.

The values of $\boldsymbol{K}$ for the discrete case were calculated using MATLAB, similar to the continuous case.

Computational simulations in MATLAB were once again performed to validate the discrete model along with the controller.

Finally, the last step corresponds to the HIL simulation. This step allows to validate the control unit developed for the active suspension in real hardware. In the case of the platform used, it is a BeagleBone Black microprocessor, containing the same $\boldsymbol{K}$ gain values of the discrete LQR calculated for the discrete model. Sampling period used was 200μs (same as the discrete case) and the signals of disturbance adopted were a filtered step pulse as the road velocity excitation profile and Gaussian Noise, the same disturbance signals used for testing the continuous and discrete model on Simulink.

The state variables of the system (tire deflection, velocity of the unsprung mass, suspension stroke and velocity of the sprung mass) are simulated in Simulink Real-Time and transmitted to the data acquisition system PCI DAC-6703 as analog outputs. Then, these analog outputs are connected to the BeagleBone Black microprocessor analog inputs (as shown in Fig. 3 by the green lines), which can be accessed by the control algorithm embedded in it, making it possible to perform all control calculations adequately.

The control variable calculated by the embedded algorithm in BeagleBone Black is transmitted back to the model being simulated on the target computer using an UDP protocol, as it was shown on section 3 and in Fig. 3 (represented by the red lines present in it).

Once all simulations are finished, results can be compared, which is the focus of the next section.

5. RESULTS AND DISCUSSION

Figure 6 presents the results for all the simulations described on the previous section, related to the first scenario tested: a filtered step road profile position as disturbance input signal for the active suspension system. On the other hand, Fig.7 presents the results for the simulations applied to the second scenario tested: a white Gaussian noise disturbance input signal applied to the active suspension system.

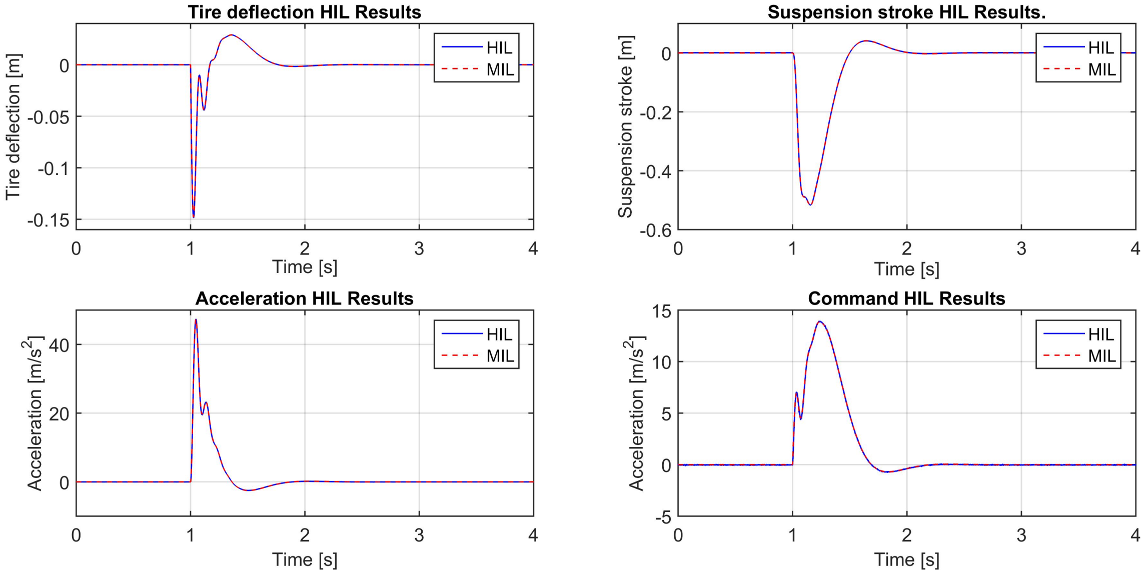


Figure 6: Active suspension results for HIL and Simulink simulations when a filtered step is applied as road profile position disturbance signal to the model.

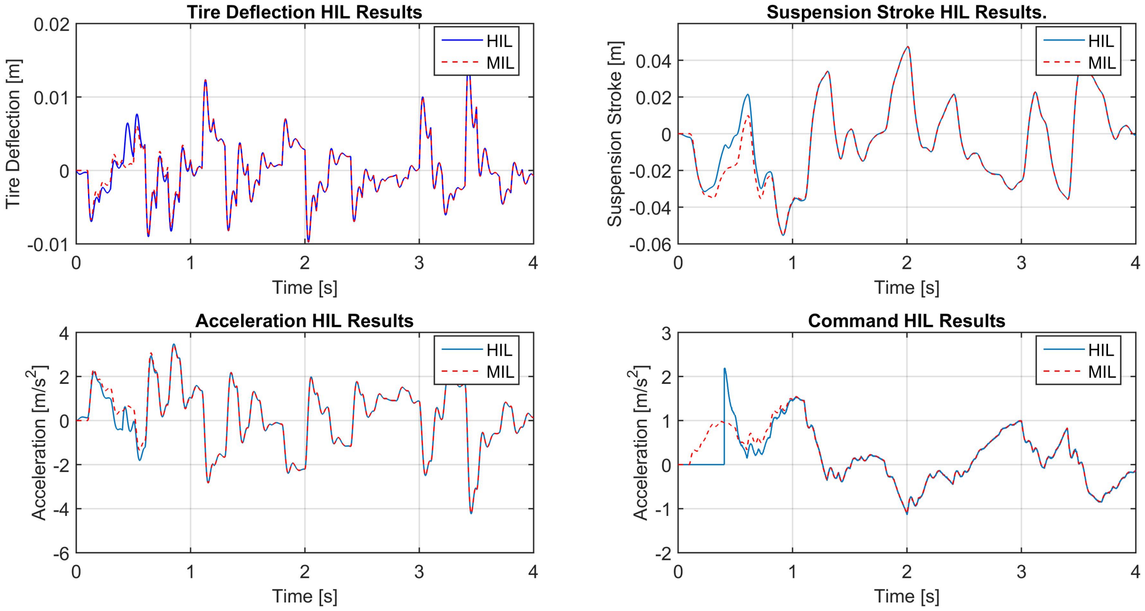


Figure 7: Active suspension results for HIL and Simulink simulations when a Gaussian white noise is applied as road profile velocity disturbance signal to the model.

Figures 6 and 7 show that, for the results obtained, Simulink and HIL simulations were very close, which validates the HIL platform and the control strategy developed for the active suspension system.

A first observation that can be made regarding these results is that the designed controller meets the previously established objectives, that is, given an excitation road profile, the system is able to absorb the disturbance, minimizing the course of the suspension and maintaining good contact between tire and road. This can be observed through the suspension stroke and tire deflection time plots shown in both Fig. 6 and Fig. 7. Once the disturbance is rejected, these parameters stabilize, ensuring these conditions.

It is also important to state that 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 acceleration of the suspension in Fig. 6 and Fig. 7, which after rejecting part of the disturbance introduced into the system, stabilizes gradually, ensuring more comfort to the passenger.

The first tested scenario (filtered step as road profile position disturbance signal) shows that HIL and MIL simulations were really close. This is expected since it a simple case for the active suspension system to handle. Although simple, it is an important validation step for the control system, once the filtered step represents an abrupt excitation, which is a common situation that occurs in practical applications.

The second scenario tested (Gaussian noise as road profile velocity disturbance signal) shows some interesting results. For most part of the curves, results are the same as the ones obtained on the previous scenario: HIL and MIL time plots are really close. There are some differences at the beginning of the results, but both simulations catch up before 1s of simulation. The bigger difference between simulations can be observed at the command variable. Since there is a small delay at the beginning of the HIL simulation, a spike occurs at first, but once again, both simulations catch up before 1s.

This is a very interesting result because it shows the effects that delays may cause on the system. Delays commonly occur in practical applications but most computational simulations do not take these delays into account. Even though it is possible to simulate delays on Simulink, most of the time it is not possible to know beforehand how much delay is introduced in the system. This is one of the most important aspects of the HIL simulation: make it possible to validate the developed control system under more realistic conditions, such as delays on the command variable calculation, than with only Simulink simulations. It is possible to obtain a more realistic response without testing the real complete system, which reduces time-to-market and cost when developing a new application.

It is also important to notice that, even though delays happened during HIL simulations, the system was able to deal with it and obtain the expected results at the end.

The control system developed in this work was validated with the HIL simulation. The hardware embedded control system was able to operate on the active suspension model satisfactorily for a sampling period of 200μs without the need to test the system with a real active suspension. Even though there will still be some differences in the performance of the control system in the case or a real active suspension, a better perception of the behaviour of the system is obtained through more realistic conditions of operation compared with purely computational simulations.

6. CONCLUSIONS

Automotive suspensions are very important systems for the performance of a vehicle. Active suspensions have gained even more importance due to the their capability of changing its parameters in real-time and allowing the system to adapt to the road profile characteristics in a more optimized way, which led to it being the focus of many studies over the years in the automotive industry.

HIL simulations are widely used for validating control systems without the need of testing a real system and provides more realistic results when compared to purely computational simulations. It is used in may different areas of industry, specially for the automotive sector, where it is considered an industrial standard for testing. Taking into consideration the importance of this type of simulation, a low-cost HIL platform was developed at the University of Brasilia and this work presents the first results with it, regarding an automotive application of an active suspension system controlled using a LQR control strategy. The system was modelled, tested and validated in MATLAB/Simulink and then validated using a HIL simulation with the platform proposed. The results show that the LQR controller developed for this application was sucessfully embedded on a BeagleBone Black microprocessor and validated for a sampling rate of 200 μs.

With the HIL simulation, it was possible to study the behaviour of the system when delays are introduced to the system, which is not always possible with purely computational simulations, but is something very recurring on practical applications. Thus, not only the control system developed was validated, but also the HIL platform presented.

For future works, it is suggested that different types of road profile are tested. This can be achieved by either changing the types of signals used as disturbance inputs of the system or by changing the weighing parameters (r_1, r_2 and r_3). Another possible future contribution is to test different kinds of application using the HIL platform. This work represents good results and shows the potential that the platform has. By changing the model being tested using Simulink Real-Time, it is possible to perform HIL simulations of different types of systems, regading aerospace, power, chemical and many other areas of application.

7. ACKNOWLEDGEMENTS

The authors acknowledge CNPq (Brazilian Council of National Scientific and Technological Development), CAPES (Coordenation of Improvement of Undergraduate and Graduate Personnel) and FAP-DF (Foundation of Research Support of Distrito Federal) for the financial support of this research.

8. REFERENCES

Cao, J., Liu, H., Li, P. and Brown, D.J., 2008. “State of the Art in Vehicle Active Suspension Adaptive Control Systems Based on Intelligent Methodologies”. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 9, No. 3, pp. 392–405.

Eltantawie, M.A., 2012. “Decentralized Neuro-Fuzzy Control for Half Car with Semi-Active Suspension System”. *Inter-*

national Journal of Automotive Technology, Vol. 13, No. 3, pp. 423–431.

Gawthrop, P.J., Virden, D.W., Neild, S.A. and Wagg, D.J., 2008. “Emulator-Based control for Actuator-Based Hardware-in-the-Loop Testing”. *Control Engineering Practice*, Vol. 16, No. 8, pp. 897–908.

Huang, D.S., Zhang, J.Q. and Liu, Y.L., 2017. “The PID Semi-Active Vibration Control on Nonlinear Suspension System with Time Delay”. *International Journal of Intelligent Transportation Systems Research*, pp. 1–13.

Isermann, R., Schaffnit, J. and Sinsel, S., 1999. “Hardware-in-the-Loop Simulation for the Design and Testing of Engine-Control Systems”. *Control Engineering Practice*, Vol. 7, No. 5, pp. 643–653.

Jian, W., Zhiyuan, L. and Houhua, J., 2015. “Semi-Active Suspension Control with Polynomial Inverse Model: Frequency Response Analysis and HIL Simulation”. In *34th Chinese Control Conference (CCC)*. IEEE, pp. 8177–8182.

Kiencke, U. and Nielsen, L., 2005. “Engine Management Systems”. *Automotive Control Systems: For Engine, Driveline and Vehicle*, pp. 43–74.

Kurose, J.F. and Ross, K.W., 2010. *Computer Networking: A Top-Down Approach*. Pearson Education, Boston, MA, 5th edition.

Lozoya-Santos, J.J., Tudon-Martinez, J.C., Morales-Menendez, R. and Ramirez-Mendoza, R., 2012. “Comparison of On-Off Control Strategies for a Semi-Active Automotive Suspension Using HIL”. *IEEE Latin America Transactions*, Vol. 10, No. 5, pp. 2045–2052.

Mathworks, 2017. *Simulink Real-Time I/O Reference Guide Version 6.6*.

Matinnejad, R., Nejati, S., Briand, L., Bruckmann, T. and Poull, C., 2015. “Search-Based Automated Testing of Continuous Controllers: Framework, Tool support, and Case Studies”. *Information and Software Technology*, Vol. 57, pp. 705–722.

Oh, S.C., 2005. “Evaluation of Motor Characteristics for Hybrid Electric Vehicles Using the Hardware-in-the-Loop Concept”. *IEEE Transactions on Vehicular Technology*, Vol. 54, No. 3, pp. 817–824.

Ohri, J. *et al.*, 2014. “GA Tuned LQR and PID Controller for Aircraft Pitch Control”. In *IEEE 6th India International Conference on Power Electronics (IICPE)*. IEEE, pp. 1–6.

Palladino, A., Fiengo, G. and Lanzo, D., 2012. “A Portable Hardware-in-the-Loop (HIL) Device for Automotive Diagnostic Control Systems”. *ISA Transactions*, Vol. 51, No. 1, pp. 229–236.

Pannocchi, L., Marinoni, M. and Buttazzo, G., 2017. “Hardware-in-the-Loop Development Framework for Multi-Vehicle Autonomous Systems”. In *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, pp. 17–22.

Rajamani, R., 2011. *Vehicle Dynamics and Control*. Springer Science & Business Media.

Redfield, R.C., 1991. “Performance of Low-Bandwidth, Semi-Active Damping Concepts for Suspension Control”. *Vehicle System Dynamics*, Vol. 20, No. 5, pp. 245–267.

Savaresi, S.M., Poussot-Vassal, C., Spelta, C., Sename, O. and Dugard, L., 2010. *Semi-Active Suspension Control Design for Vehicles*. Elsevier, 1st edition.

Ulsoy, A.G., Peng, H. and Çakmakci, M., 2012. *Automotive Control Systems*. Cambridge University Press, New York, NY, 1st edition.

Watteyne, T. and Auge-Blum, I., 2005. “Proposition of a Hard Real-Time MAC Protocol for Wireless Sensor Networks”. In *13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*. IEEE, pp. 533–536.

9. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.