



Hardware-in-the-loop simulation of DC motor as an instructional media for control system design and testing

Muhammad Zakiyullah Romdlony^a, Fakhri Irsyadi^{b,*}

^a School of Electrical Engineering, Telkom University
Jalan Telekomunikasi No. 1, Bandung, 40257, Indonesia

^b Department of Electrical Engineering and Informatics, Vocational College, Universitas Gadjah Mada
Jalan Yacarana, Sekip Unit III, Yogyakarta, 55281, Indonesia

Received 1 July 2021; Accepted 14 September 2021; Published online 31 December 2021

Abstract

Instructional media in control systems typically requires a real plant as an element to be controlled. However, this real plant, which is costly to be implemented, can be replaced by a virtual plant implemented in a computer and modelled in such a way that it resembles the behavior of a real plant. This kind of set-up is widely termed as hardware-in-the-loop (HIL) simulation. HIL simulation is an alternative way to reduce the development cost. A virtual plant is easy to adjust to represent various plants or processes that are widely used in industry. This paper proposes a simple HIL simulation set-up designed as instructional media for design and testing a simple control system. The experimental result on DC motor control shows that HIL simulation dynamical response is similar to the real hardware response with a small average error on measured transient response, represented in 0.5 seconds difference in settling time and 7.43 % difference in overshoot. This result shows the efficacy of our HIL simulation set-up.

©2021 Research Centre for Electrical Power and Mechatronics - Indonesian Institute of Sciences. This is an open access article under the CC BY-NC-SA license (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).

Keywords: control systems; hardware-in-the-loop (HIL); instructional media.

I. Introduction

The Industrial revolution gives big challenges for engineering education institutions. The institutions have to provide students not only basic knowledge but also competency that is matched with the industrial needs. That approach is motivated by acknowledging the career readiness of students as an important learning outcome. The institutions have to provide various tools and facilities that are currently used in the industry. It can give a great experience for the student to learn in the same fashion as they will face when they work in the real workplace. However, it is not easy for most institutions to set up that ideal environment even if they have the budget for expensive equipment. They must provide enough space to place the equipment and cost for maintenance of the equipment. For some kinds of hardware, it might not be feasible to give access to students due to the risk of damaging expensive equipment. Also, in special circumstances

such as Covid-19 pandemic, most education institutions restrict access to the laboratory due to health and safety regulations [1], thus students have lack of hands-on experience, even though the hardware is available.

Hardware-in-the-loop (HIL) simulation can be used to be an alternative solution for these problems as it proposes the "Lab-at-home" paradigm [2]. HIL simulation uses virtual plants or processes that can be used to test the performance of developed control algorithms and also animated work principles of real machines or processes. The virtual plant/process is adjustable. It can be used to represent various plants or processes that are rapidly evolving. It can reduce time and cost for infrastructural development. It also makes the development process more realistic and safer.

HIL simulation in education is very common. There are several papers that propose the example of HIL simulation application in some fields of education, such as in energy [3], automation, and robotics [4][5]. HIL is also implemented in automotive control [6][7][8] and agricultural tractor [9]. There are various combination devices that are

* Corresponding Author. Tel: +62-274-6491302
E-mail address: fakhri.irsyadi@ugm.ac.id

used in the HIL simulation set up. Some HIL simulations use advanced and sophisticated devices to simulate complex systems. For less complex systems, a low-cost hardware combination can be opted. Some set-ups use a low-cost, portable and popular microcontroller Arduino as a controller [10][11][12] and a single-board computer such as Raspberry Pi [13]. Most of that set-up uses a data communication channel to integrate the controller hardware into the simulation device. This scenario implies that the code of the controller cannot be directly used to control the real system. It needs some adjustment, especially in the input-output part.

This paper proposes an alternative design of a simple and low-cost HIL simulation set-up based on Arduino. The HIL simulation set-up consists of a computer as a simulation device, an Arduino as the main controller, National Instrumentation data acquisition systems (DAQ), and a simple resistor-capacitor (RC) circuit as signal conditioning for system integration. This set-up can be used as instructional media in engineering education or training institutions. In this paper, we present the application of such an HIL simulation set-up for DC motor control. Experiments are conducted to measure the performance of the HIL simulation set-up and to test a simple proportional integrator (PI) control on rotational speed control of a DC motor.

II. Materials and Methods

This section discusses real-time simulation, the topology of HIL simulation and details steps for the proposed HIL simulation implementation.

A. Real-time simulation

Real-time simulation refers to a simulation that can execute the computation at the same rate as the actual physical system. Based on the form of the components of real-time simulation, there are three types of real-time simulation, as shown in Table 1.

The first scheme consists of simulated control theory (software-based) and real plant/process. It is commonly called prototyping. The advantage of this scheme is that the engineer can directly implement continuous-time domain control theory without digitizing any process; most control theory is developed in a continuous-time domain. This scheme is also commonly used in online tuning because it uses real plants/processes to provide feedback signals for control algorithms.

The second scheme is software-in-the-loop (SIL) simulation. In this scheme, both components (controller and plant/process) are simulated on the simulation software. One of the most challenging phases to set up this scheme is modelling the plant/process. In several cases, it is not easy to get a

mathematical model that represents the dynamical system, especially for the complex plant/process. This scheme is commonly used for education purposes, such as in control engineering courses. It also can be used to develop advanced control that is developed based on a model of plant/process.

The last scheme is HIL simulation. In this scheme, a real electronic control unit (ECU) is used to control a simulated plant/process modelled on the simulation software. One of the advantages of this scheme is that control development can be directly implemented on the real system with just small modifications.

B. Hardware-in-the-loop simulation

HIL simulation is one kind of real-time simulation. It was used more than 15 years ago. It was commonly used for complex systems in the industry. Several works implementing HIL can be found in [15][16][17]. Two main reasons HIL is suitable for various industries, are that it can decrease "time to market" and reduce the complexity of the real-time simulation. The purpose of the HIL simulation is to develop and test the control algorithm implemented in real ECU to the simulated plant. Some advantages of using HIL simulation are as follows:

- It can reduce development costs and time to market because it is not necessary to build a real plant or process.
- Some of complex systems need to do the development and testing of plants/processes in separate areas. The usage of HIL simulation is very suitable because it is easy to modify or adapt the plant/process.
- When the plant model is valid and accurate, the tuning of control variables can be done on simulation and the result can be implemented to the real plant with just a small modification.
- The use of HIL simulation can reduce the risk of damage to the overall system and environment during the development and testing process.
- HIL simulation can be used as instructional media to explain the working principles of a plant/process.

The control structure of the HIL simulation is presented in Figure 1. The simulation consists of a controller implemented in the real hardware and plant/process simulated in simulation software.

C. Implementation process

1) Plant modeling

The first step for HIL simulation realization is plant modelling. The aim of this step is to find mathematical equations that can represent the dynamic behavior of a real plant/process. Those result equations would be implemented on simulation software to be a virtual plant on HIL simulation. There are two methods that can be used to do plant/process modelling. In case of a lack of information of plant parameters or specifications, one typically uses a system identification method. This method uses a pair of data input and output to produce the model or mathematical equations of a

Table 1.
Various real-time simulation scheme [14]

Scheme	Set-up	Controller	Plants/processes
1	Prototyping	Real	Real
2	SIL	Simulated	Simulated
3	HIL	Real	Simulated

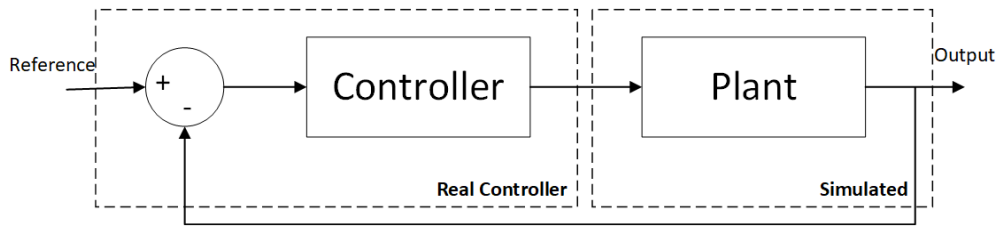


Figure 1. Structure of hardware-in-the-loop simulation

real plant/process. In this paper, we demonstrate a simple experiment to derive a DC motor model from experimental data. The modelling process consists of several steps as follows.

a) Sensor calibration

In this step, the measured data of the sensor is compared with the measured data of standard measurement instruments to validate the sensor. An optical encoder is used to measure the rotational speed of a DC motor and validated with an optical tachometer. Table 2 shows the comparison between those two measurements for each given input voltage. The average error is 4.625 rpm which is relatively small.

b) Data gathering

The system identification methods need a pair of input-output data to provide the modelling equations. In this case, the input signal is the input voltage of the DC motor and the output signal is the rotational speed of the DC motor.

c) Algorithm execution

The recorded input and output data are submitted to the system identification software in order to get the model [18]. MATLAB System Identification Toolbox is used to find the transfer function, the correlation between the input voltage and rotary speed of the DC motor.

Based on the derivation of the modelling DC motor using system identification as reported in [16], the relation between rotational speed $\omega(t)$ and input voltage $v(t)$ of the DC motor, with assumes that the value of armature inductance is small compared to the armature resistance and it can be neglected, is approached into a first order system without

zero. In our case, the resulting transfer function is

$$G(s) = \frac{\Omega(s)}{V(s)} = \frac{117.3}{s+1.14} \quad (1)$$

d) Model validation

After the mathematical model of a real plant is found, the next step is validating the model. In this step, both the real plant and the model are injected with the same input signal value. In this case, the input value is the input voltage of the DC motor in the range of 2 V until 11 V, which is in the linear operation range of the DC motor.

Figure 2 shows the result of a model validation experiment. The graph shows that the dynamic response of the model is close to the dynamic response of a real plant. The accuracy of the model is 87,95% compared with real plant output for the same combination of input.

This result implies that the mathematical modelling of a real plant is valid and it would be implemented as a virtual plant/process in the HIL set-up. Figure 2 shows that the highest error of the model is on the lower input voltage area. The reason is that the model generated by the system identification algorithm is a linear model that cannot capture the nonlinearity characteristic, i.e., a dead zone of the DC motor.

2) Hardware-in-the-loop setup

a) Hardware design

In this section, the proposed HIL simulation set-up is discussed. The interconnection of each component in the set-up is shown in Figure 3.

The ECU of this system is an Arduino microcontroller. This component is used for reading reference signal (that can be adjusted externally by a potentiometer) through

Table 2.
Optical encoder calibration

Voltage (volt)	Rotational speed - tachometer (rpm)	Rotational speed - optical encoder (rpm)	Error (rpm)
4	355	346	9
6	549	550	1
8	752	754	2
10	960	958	2
12	1182	1183	1
14	1460	1466	5
16	1662	1672	10
18	1849	1856	7
Average error (rpm)			4.625

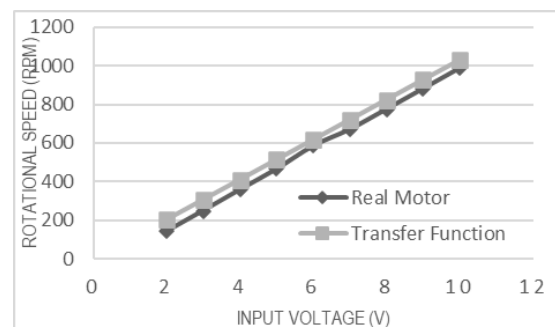


Figure 2. Graph of error modeling

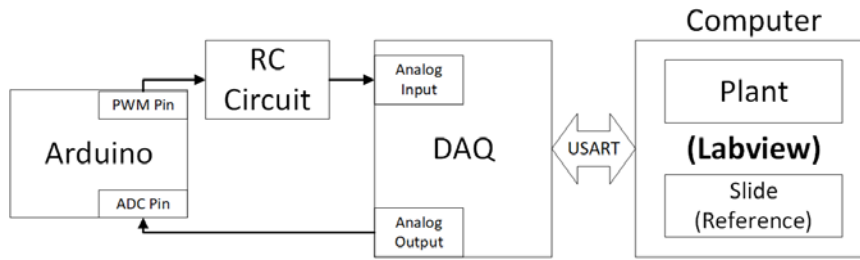


Figure 3. The proposed hardware-in-the-loop simulation interconnection

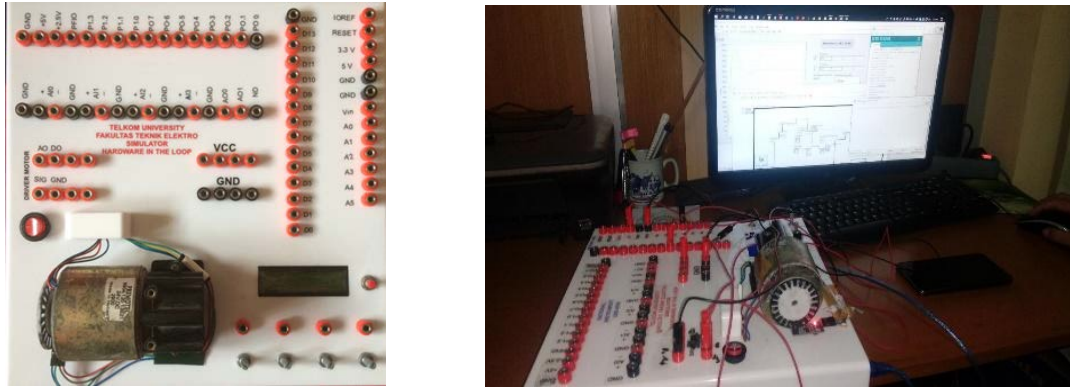


Figure 4. The proposed hardware-in-the-loop simulation set-up

analog-to-digital converter (ADC) pin, read the feedback from the simulated plant, execute the control algorithm, and provide the control signal to the simulated plant through digital-to-analog converter (DAC) and DAQ components. DAC, in this set-up, is a simple RC circuit used to convert the control signal in the form of pulse-width modulation (PWM) to an analog value. In this set-up, DAQ is used to convert the analog signal from DAC to data (serial communication protocol) to communicate with the simulated plant/process in simulation software. This component also converts the signal from the simulated plant/process to an analog signal as a feedback control signal.

b) System realization

This subsection discusses the implementation of hardware and software systems that are used in the experiment. The overall specifications of the system are as follows (Table 3).

The system realization is shown in Figure 4 and Figure 5. Figure 4 shows the realization board of controller and signal processing discussed in Figure 3. The interconnection of each component is wired by banana connectors. The hardware realization includes

a power button, potentiometers for setting up the reference value and tuning up the control parameters, and a liquid-crystal display (LCD) as a simple display. The DC motor, as a real plant, is also mounted on the board. Figure 5 shows the graphical user interface (GUI) for interactively changing the parameters and displaying the time plot of the output signal.

III. Results and Discussions

This section discusses the result of HIL implementation for controlling a DC motor. An experiment is conducted to compare the performance of HIL simulation and hardware implementation (prototyping, scheme 1 as in Table 1). In this experiment, a simple PI control was implemented on the system to control the rotational speed of the DC motor. The PI control parameter was tuned on simulation software to provide various performances of the system.

The experiment results are shown in Table 4. The trend of dynamic response in HIL simulation is similar to real hardware simulation. The average difference of the transient response between HIL simulation and prototyping is around 0.5 seconds. The average difference of maximum overshoot between HIL simulation and real hardware simulation is around 7.43 %.

Table 3.
Specification of the system

Hardware	Software
Personal computer	Arduino IDE 1.8.8
Arduino board Uno R3 (Atmega328P)	NI LabVIEW 2017 64bit student version
NI USB-6008 (DAQ), 8 inputs, 12 bit, 10k/s multifunction I/O	MATLAB student version for system identification
DC motor 24 V	
Optical encoder (24 holes)	

Table 4.
Performance comparison

Control parameters		Settling time (second)		Maximum overshoot (% rpm)	
Proportional gain (K_p)	Integral gain (K_i)	HIL	Prototyping	HIL	Prototyping
0.0519089126	0.5326817035	2.3	2.8	24.70	30.50
0.0399423885	0.3308636903	2.6	3.8	24.20	31.80
0.0276177501	0.1744241952	4.4	4.2	21.60	30.50
Average		3.1	3.6	23.50	30.93
Difference between HIL and prototyping		0.5		7.43	

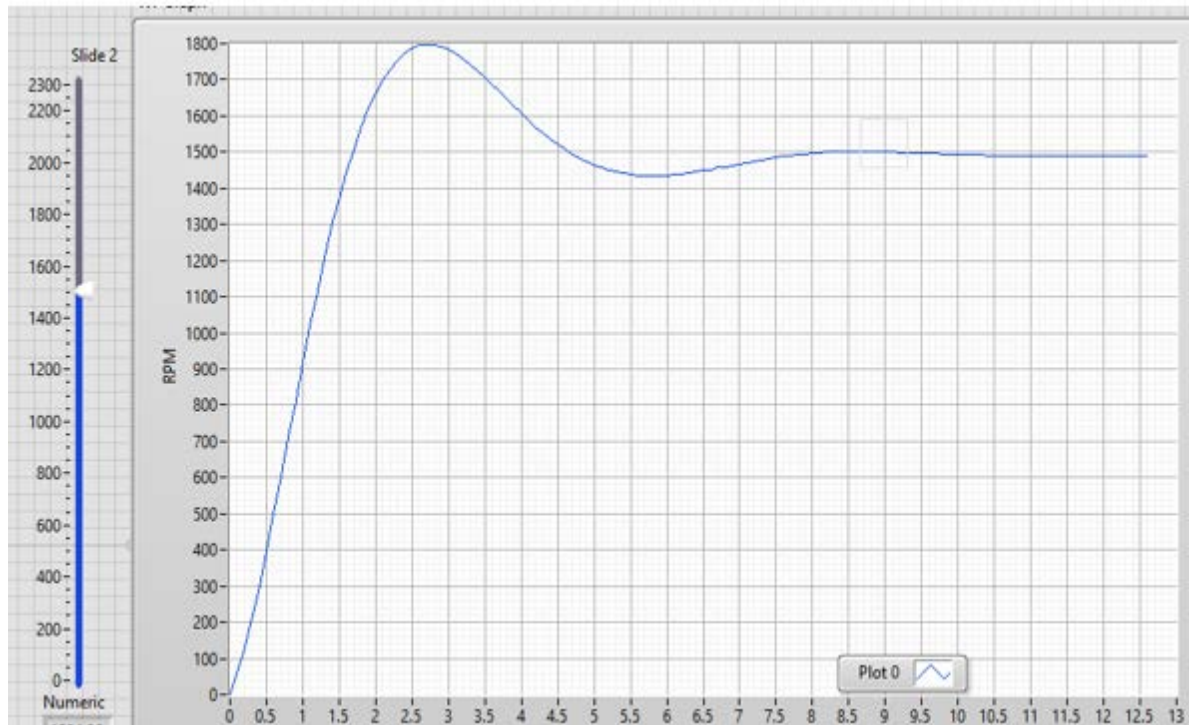


Figure 5. Graphical user interface developed in LabVIEW

IV. Conclusion

In this paper, a simple HIL simulation set-up is successfully designed. It can be used to implement closed-loop control algorithms for development, testing, and educational purposes. Experiment results show that the trend of dynamic response in HIL simulation is similar to the one in real hardware implementation, with an average error of 0.5 seconds in measured transient response and 7.43% in maximum overshoot. These findings show that the HIL simulation set-up is able to represent the dynamic response of a real plant or process. In order to enhance the accessibility of the proposed HIL simulation set-up for future development, we suggest the use of open-source simulation software, for example, Sci-Lab, for simulating the plant/process in the HIL setup.

Acknowledgment

This research was supported by The Directorate of Research and Community Service (PPM) of Telkom University.

Declarations

Author contribution

All authors contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no conflict of interest.

Additional information

Reprints and permission information is available at <https://mev.lipi.go.id/>.

Publisher's Note: Research Centre for Electrical Power and Mechatronics - Indonesian Institute of Sciences remains neutral with regard to jurisdictional claims and institutional affiliations.

References

- [1] K. Gamage *et.al.*, "Online delivery of teaching and laboratory practices: continuity of university programmes during Covid-19 pandemic," *Education Science*, 2020.

- [2] S. Usenmez *et al.*, "A new hardware-in-the-loop simulator for control engineering education," in *IEEE Global Engineering Education Conference (EDUCON)*, 2014.
- [3] A. Gambier, "Real-time control and hardware-in-the-loop simulation for educational purposes of wind energy systems," *ScienceDirect*, vol. 53, issue 2, pp. 17344–17349, 2020.
- [4] M. Goubey and M. Langmajer, "Experience with use of HIL simulators in control engineering course," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 17308–17313, 2020.
- [5] O. L. Osen, "On the use of hardware-in-the-loop for teaching automation engineering," in *IEEE Glob. Eng. Educ. Conf. EDUCON*, vol. April-2019, pp. 1308–1315, 2019.
- [6] A. Soltani *et al.*, "A hardware-in-the-loop facility for integrated vehicle dynamics control system design and validation," *7th IFAC Symposium on Mechatronic Systems MECHATRONICS*, Leicestershire, UK, 2016.
- [7] A. Taksale *et al.*, "Low cost hardware-in-loop for automotive application," in *International Conference on Industrial Instrumentation and Control (ICIC)*, India, 2015.
- [8] L. Heidrich *et al.*, "Hardware-in-the-loop test rig for integrated vehicle control systems," *7th IFAC Symposium on Advances in Automotive Control*, Tokyo, Japan, 2013.
- [9] S. Raikwar *et al.*, "Hardware-in-the-Loop test automation of embedded systems for agricultural tractors," *Measurement*, 133, pp. 271–280, 2019.
- [10] G. Gambino *et al.*, "A low-cost HIL platform for testing professional refrigerators controllers," *IFAC*, vol. 19, no. 3, 2014.
- [11] I. Tejado, J. Serrano, E. Pérez, D. Torres, and B. M. Vinagre, "low-cost hardware-in-the-loop testbed of a mobile robot to support learning in automatic control and robotics," *IFAC-Papers OnLine*, vol. 49, no. 6, pp. 242–247, 2016.
- [12] S. Biswas, "Real-time arduino based simulator enabled hardware-in-the-loop electric DC machine drive system," in *IEEE Region 10 Humanitarian Technology Conference*, Dhaka, Bangladesh, 2017.
- [13] J. Sobota, "Raspberry Pi-based HIL simulators for control education," *IFAC Papers OnLine*, pp.52-9 68–73, 2019.
- [14] R. Isermann, J. Schaffnit, and S. Sinsel, "Hardware-in-the-Loop simulation for the design and testing of engine-control systems," in *IFAC Proc.*, vol. 31, no. 4, pp. 1–10, 1998.
- [15] X. Wang, Yin T, Chen J, et al., "Rapid design of motor DC speed control system based on Matlab," *Applied Mechanics & Materials*, 743:168-171, 2015.
- [16] Tang, Wei-Jie, Zhen-Tao Liu, and Qian Wang. "DC motor speed control based on system identification and PID auto tuning," in *Chinese Control Conference (CCC)*, 36th Chinese. IEEE, 2017.
- [17] Ismeal, Godem A., Karol Kyslan, and Viliam Fedák. "DC motor identification based on recurrent Neural Networks," in *Mechatronics-Mechatronika (ME)*, 2014 16th International Conference on. IEEE, 2014.
- [18] L. Ljung, "Prediction error estimation methods," *Circ. Syst. Signal Process*, 21, 11–21, 2002.