Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Experience with use of HIL simulators in control engineering course

Martin Goubej, **Martin Langmajer**

University of West Bohemia, Department of Cybernetics, Pilsen, Czech Republic (e-mail: mgoubej@ntis.zcu.cz, gosh@kky.zcu.cz).

Abstract: The goal of the paper is to share experience with the use of hardware-in-the-loop (HIL) simulators in a control-engineering course being taught at the University of West Bohemia. The hardware simulators were introduced recently in the course curriculum aiming to get more realistic application scenarios for the students. They allow simple explanation of the concepts of model-based systems engineering in a form close to the workflow used in industrial practice. The achieved results show some significant benefits when compared to former course content, which relied on numerical simulations only. The paper presents one of the application use-cases dealing with the problem of active car suspension control. Individual phases of the control system development as done by students are explained step by step, revealing the main benefits of the hands-on experience with the physical setup.

Keywords: control education, automatic control, control systems engineering, hardware-in-the-loop simulation, model-based design, quarter-car system

1. INTRODUCTION

We are currently witnessing rapid progress in technology development affecting many application domains. The universities focused on the science, technology, engineering and mathematics (STEM) disciplines are the institutions responsible for training new generations of engineers expected to master the technology (Hallinen (2015)). They are facing new challenges while trying to keep up with the constantly changing environment, follow recent trends and breed graduates capable of finding employment in technical practice.

The need of change in the role of academia with respect to society is sometimes designated as '4th generation universities' concept (Lukovics and Zuti (2015)). One of the main issues identified so far is the increasing gap between the needs of industrial practice and content of the higher education delivered by those institutions (Sobota et al. (2019), Čech et al. (2019)). In the scope of the control engineering field, it is often observed that more focus is given on the theoretical part of the subject (models, equations, algorithms...) while disregarding the technical aspects necessary to employ controls in practice. Many control theory courses focus on the algorithmic part only, using numerical simulations as the main means of validation of the results (Despeisse (2018), Venkatalakshmi et al. (2016), Smith and Pollard (1986)). This is quite understandable as it allows to demonstrate the theoretical concepts rapidly without bothering with implementation details, which should be covered in other subjects. However, our experience shows that this often causes severe confusion and misunderstanding of the basic principles behind control engineering. Students that never had a chance to close any real control loop cannot deeply understand the difference between the model and real plant, reveal fundamental limitations given by the used instrumentation and control software and hardware or be aware of danger of instability caused by improper feedback design potentially causing damage or destruction of the equipment.

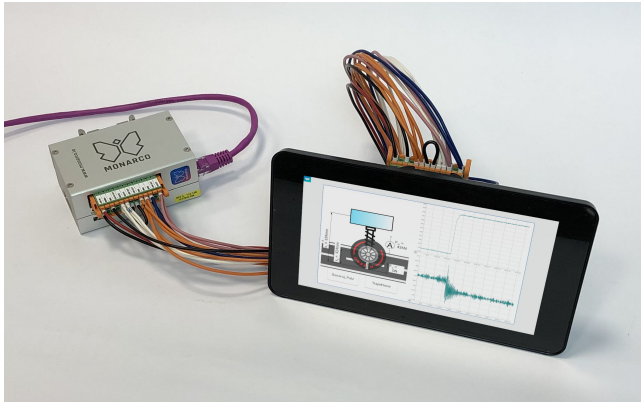

Fig. 1. HIL simulator & controller setup based on Raspberry Pi and Monarco HAT HW platform

One of the possibilities to bring the teaching of the control subjects closer to reality is to introduce various physical models emulating real-life control problems (Perumal and Ganesan (2018)). They bring up various implementation issues close to practical applications while keeping the complexity at a reasonable level and they proved to be an invaluable tool for control education. On the other hand, the purchase of such hardware is often expensive, the models need some maintenance and it is usually not affordable to equip each student with its own setup. A common way to alleviate this is to introduce remote or virtual laboratories allowing either sharing of the physical hardware with the users via remote connection or replacing it by a software simulation, see e.g. Uribe et al. (2016),

Ma and Nickerson (2006), Gomes and Bogosyan (2010) or Heradio et al. (2016).

Both approaches come with specific drawbacks which can be mitigated by introducing real-time hardware simulators representing the controlled plants (Sobota et al. (2019), Parodi et al. (2009), Smolinski et al. (2017), Rahmani and Hashemi (2015)). They are much cheaper than the physical models while retaining some of the key features, namely the control through a set of physical inputs and outputs and emulation of real-time response via proper visualization tools. This gives the students at least a feeling of controlling a real plant and allows to demonstrate most of the practical issues encountered in control engineering. They can serve as a perfect complement to virtual software models and physical setups used for education.

The paper is organized as follows. Section II deals with hardware details regarding the HIL simulators used at our department. Section III presents one particular use case employed in a semester project in terms of a second undergraduate control engineering course. The students are guided in the process of control system design trough several intermediate steps of model-, software-, processor- and hardware-in-the-loop simulation scenarios following the workflow typical for technical practice. Each of the individual phases focuses on different aspects of control engineering allowing to develop more thorough understanding of the subject compared to purely numerical simulations, as discussed in Section IV.

2. RASPBERRY PI-BASED HIL SIMULATOR

A set of a Raspberry Pi minicomputer together with a Monarco HAT add-on board forms the basis of the HW simulator platform. The Raspberry Pi 3 contains 1 GB of RAM and 1.4 GHz quad-core CPU providing sufficient computational power for our purposes (The Raspberry Pi Foundation (2018)). Monarco HAT offers 4 digital and 2 analog inputs and 4 digital and 2 analog outputs (REX Controls, s.r.o. (2016)). Complementing it by REXYGEN software tools (REX Controls s.r.o. (2019b)) and 7" touch screen display allowed to create a perfect low cost HIL simulator for the purpose of education. The simulator provides industry standard analog signals in 0-10 V range and digital signals in 24 V logic allowing to connect any PLC or compact controller. More details regarding the HW part can be found in Sobota et al. (2019).

3. QUARTER-CAR USE CASE

This section deals with a particular application case used as a semester project in a second course of control systems engineering. The students are already familiar with the basic concepts of linear systems theory from the preceding course including state space and transfer function models, frequency domain characteristics, stability theory, root locus and simple feedback structures including lead-lag or PID controllers. They are gradually introduced to more involved subjects such as digital control, observers, state feedback, time-delay systems, modal control via pole-placement techniques or frequency domain loop-shaping methods.

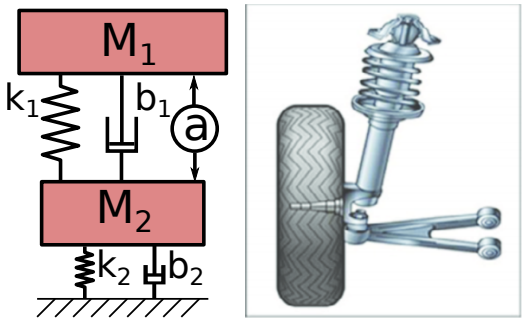

Fig. 2. Quarter-car suspension model represented by a two-mass system (Alvarez Sanchez (2013))

A quarter-car suspension model representing the plant to be controlled was chosen for implementation in the HIL simulator for several reasons:

- The system is easy to visualize, all the students are familiar with cars from their everyday life
- Oscillatory dynamics can be introduced which brings some inherent difficulties from control perspective, allowing to explain the importance of several design choices, the difference between open- and closed-loop behaviour is clearly visible at a first glance
- The system is simple enough for the purpose of analysis and control design, on the other hand, complex enough to demonstrate all the relevant theory in practice
- The time constants of the system can be short enough to allow fast execution of experiments without time-consuming waiting, online parameter or input changes manifest immediately in the observed plant response
- Nonlinear behaviour can be easily incorporated in the model to explain differences between real plant (represented by the HIL setup) and the idealized model obtained from the process of local linearization
- Inherent trade-offs emerging in control design can easily be demonstrated, e.g. bandwidth vs noise amplification, robustness to unmodelled dynamics, actuator/sensor imperfections etc.

3.1 Dynamic model

The quarter car model (Fig. 2) is a commonly used simplified model of the car chassis suspension (Alvarez Sanchez, 2013). It consists of two masses connected by spring and damper elements. The higher mass usually represents the car body while the lower one stands for the wheel. The car and the wheel are connected through a damper represented by a spring element. Another damper-spring pair connects the lower mass with the ground and represents the flexibility of the tire.

The model used in the simulator introduces nonlinear springs and dampers or actuator and sensor saturation/rate limits, that simulate the real behaviour. The purpose of this is to show the issues connected with real plants to students and teach them how to cope with them. The students do not have the access to the exact form of the model and approach the HIL simulator as a black-box representing the real physical plant.

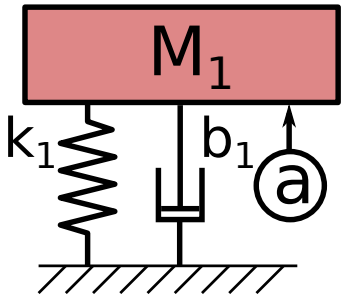

Fig. 3. Simplified single-mass system

3.2 Phase 1: Basic understanding and analytical modelling

The first goal is to get students familiar with the model and simulator. The model has two inputs. First of them simulates an unmeasured input disturbance in the form of external force exerted by road bumps. The input is excited by a signal generator implemented in the simulator which can create both random and deterministic signals. The second one is the manipulating variable representing the actuator force between the car and the the wheel. The students should design effective feedback control system that keeps the car height constant, delivering an active suspension functionality.

In the first phase, students derive an idealized mathematical model of the system with linear elements (Fig. 2) representing a local behaviour around the equilibrium point

$$m_1\ddot{x}_1 = F_m - m_1 g - b_1(\dot{x}_1 - \dot{x}_2) - k_1(x_1 - x_2 - x_{01} + x_{02}) \quad (1)$$

$$m_2\ddot{x}_2 = -F_m + F_d - m_2 g + b_1(\dot{x}_1 - \dot{x}_2) + \\ + k_1(x_1 - x_2 - x_{01} + x_{02}) - b_2\dot{x}_2 - k_2(x_2 - x_{02}), \quad (2)$$

where m_i is the mass, k_i the stiffness and b_i the damping of the i-th element, x_i, $\dot{x}_i$ and $\ddot{x}_i$ denote position, speed and acceleration, x_{0i} is the equilibrium position of the i-th mass, g is the gravity constant, F_m is the force generated by the actuator and F_d is the disturbance caused by a variable road profile.

They create the model in the Matlab-Simulink environment and measure several time- and frequency-domain responses using a set of approximate plant parameters. They realize that the frequency response of the system contains two flexible modes corresponding to the dynamics of the car and the wheel. The dominant dynamics is caused by the car-suspension flexibility. The second oscillatory mode caused by the wheel is manifested only at high frequencies. This is the reason why the mathematical model can be simplified for the purpose of control algorithm design, leading to the single-mass system in Fig. 3 with the equation of motion given as:

$$m_1\ddot{x}_1 = F_m - b_1\dot{x}_1 - k_1(x_1 - x_{01}). \quad (3)$$

Later, the students measure the responses of the simulator and find out that the system is not linear for higher actuator forces. They have to choose a suitable linear working area and identify model parameters. For the identification purposes, they assume the single mass model structure trying to model the first flexible mode only. The lessons learned from this phase include the use of state space models, local linearization around system equilibrium points and model reduction issues.

3.3 Phase 2: Data-driven identification

Students have determined a proper structure of the model but do not know the exact values of the model parameters. That is why they perform a gray-box identification. First they form a linear regression model and derive its optimal parameters using the least squares method based on the experimental data

$$Y = \Phi\Theta + \varepsilon,\ \Theta^* = argmin\{J(\Theta) = \varepsilon^T\varepsilon\} = (\Phi^T\Phi)^{-1}\Phi^T Y \quad (4)$$

where Y is a vector of measurements, Φ is a matrix of regressors, Θ is a vector of searched parameters, ε is a residuals vector and Θ^* is an optimal parameters estimate obtained from the ordinary least squares (OLS) method.

Students realize that the OLS method does not provide good results due to the bias in the estimated parameters caused by noise-corrupted measurements. The next step is to learn how to use more complex Prediction error and Instrumental variable methods implemented in the System identification toolbox of Matlab. They repeat the whole process for the data measured from the simulator, calculate the estimated mass, damping and stiffness and compare the model outputs with the HIL plant response using a validation trajectory.

This phase learns them how to cope with complex systems with unknown parameters. It gives a brief introduction to system identification, which is a part of the following System identification course. Experiences from this course then leads to understanding of how important is to have a valid model for the purpose of simulations and control design.

3.4 Phase 3: Model-in-the-loop

Based on the identified model of controlled system, several different approaches to controller designing are introduced. The results are validated on the virtual models of controller and controlled system in Matlab-Simulink environment in the Model-in-the-Loop (MIL) setting.

Students have to meet several design constraints defined both in time and frequency domain, e.g. maximum settling time, overshoot and actuator effort, bandwidth or robustness margins. They start with a design of PID controllers in the standard ISA form

$$u(t) = K\{e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{\mathrm{d}}{\mathrm{d}t}e(t)\}, \quad (5)$$

where $u(t)$ is the output of the controller, $\{K, T_i, T_d\}$ are the gains of proportional, integrative and derivative components and $e(t)$ is the tracking error. The difficulties of implementing the ideal derivative action are explained, followed by a modification to the filtered variant with the controller transfer function given as

$$\frac{U(s)}{E(s)} = K\left(1 + \frac{1}{T_i s} + \frac{T_d s}{T_f s + 1}\right). \quad (6)$$

Integrator windup problems are studied and several versions of anti-windup mechanisms are introduced. The students are led to use various design methods including root locus, pole-placement or loop-shaping techniques. The controller has to be robust and fast enough at the same

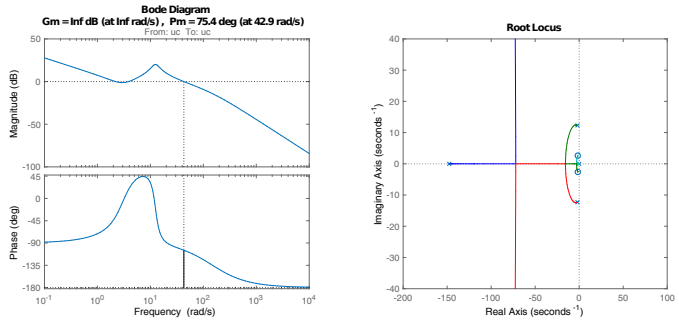

Fig. 4. Frequency and algebraic-domain controller design in SISOtool GUI

time which forces them to fulfill several contradictory requirements. Then they evaluate their designs, at first in the ideal environment during off-line numeric simulations.

Students learn to use the SISOtool GUI of Matlab which allows them to combine time, frequency and also algebraic-domain information about the closed-loop performance (Fig. 4). Therefore, they get a deeper insight to relevant connections between the three domains allowing them to understand the results of the design choices they make. Several bandwidth limitations appear due to the unmodelled dynamics (mainly the neglected second mode), noise and actuator imperfections introduced in the HIL setup. Therefore, a robust controller design is required.

The next goal is to deal with a state space control. A linear state feedback is designed based on the model, assuming the states are measured in the first step. The students recognize that the controller steers the system to the stable equilibrium, but does not allow precise setpoint tracking and disturbance rejection without further adjustments. Therefore, two modifications of the controller are introduced. The first one which uses a feedforward compensation of closed-loop static gain

$$u(t) = -Kx(t) + u_k(t) = -Kx(t) + K_k w(t), \qquad (7)$$

where $u(t)$ is the manipulating variable, $x(t)$ is the state vector, K is the controller gains vector and $u_k(t)$ is a compensation control composed as a properly scaled reference variable. The second more advanced variant uses an integrator driven by the output feedback

$$\begin{aligned} \dot{x}_i(t) &= w(t) - Cx(t), \\ u(t) &= -Kx(t) + k_i x_i(t), \end{aligned} \qquad (8)$$

$w(t)$ is a reference signal, C is the output matrix of the controlled system, x_i is a state added by the integrator and k_i is an integral gain. The Internal model principle is explained, advocating the necessity of the integrator in the loop for achieving zero steady-state error under assumption of constant disturbances. It is shown how to extend the controller structure to an arbitrary internal model of the exogenous disturbance.

When the controllers are designed well, the students face the problem of immeasurable state quantities. They learn about the asymptotic observers in the form of

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + \kappa[y(t) - C\hat{x}(t)] \\ \hat{y}(t) &= C\hat{x}(t), \end{aligned} \qquad (9)$$

where $\hat{x}(t)$ is the estimated state, A, B and C are a dynamic, input and output matrices of a controlled system, $\hat{y}(t)$ is an estimated output and κ is a matrix of the innovation feedback gain parameters.

Pole placement method is used for both controller and estimator design. A link to optimal control is also explained by introducing the ITAE performance criterion

$$J(p) = \int_0^\infty t|e(\tau, p)|d\tau, \qquad (10)$$

where $J(p)$ is criterion function, $e(\tau, p)$ is error, t and τ is time and p is set of controller's parameters. The set of parameters must be chosen to minimize the cost function.

This phase is essential from the theoretical and algorithmic point of view. The students develop understanding of various control design methods and means for their evaluation using closed-loop models.

3.5 Phase 4: Software-in-the-loop

The goal of this phase is to implement the designed controllers in the real-time software environment of the target control platform. Suitable discretization of control algorithms has to be done to make them compatible with the sampled-data control system. During the course, the REXYGEN control framework is being used (REX Controls s.r.o. (2019b)). Students can download a free version of REXYGEN development environment into their own computers and implement the model of the system and the control strategy. This part is called software-in-the-loop (SIL) and the main goal is the validation of the *correct implementation* of the algorithms, which were designed in the previous phase. Students encounter practical aspects of PID controllers design and study various forms of their discrete-time implementation.

REXYGEN control system

REXYGEN is a real-time control system developed by REX Controls company (REX Controls s.r.o. (2019a)). It was chosen to be the platform for the controller implementation as it represents a commercial industrial grade software which uses concepts that students may encounter in practice after graduation. On the other hand, it allows implementation of complex control applications by graphical programming without much hand-coding in the way similar to the Simulink environment, which students already know from previous courses.

The control algorithm can be composed from a library of existing functional blocks implementing various functionalities. The user is also allowed to write a custom functional block using a C-like or Python scripting languages. The control application can be divided to several tasks that are periodically repeated with a chosen update rate. A real-time scheduler takes care about proper timing of the tasks and assigns the CPU time based on the defined priorities. The REXYGEN system supports most of the state of the art communication protocols like CAN/CANopen, Ethernet, EtherCAT, Modbus, Profinet, or Ethernet Powerlink allowing seamless connection to various kind of peripherals and I/O devices. Remote access via TCP/IP protocol allows simple connection to the control platform from a personal computer over Ethernet. The actual state of the control algorithm can be observed in

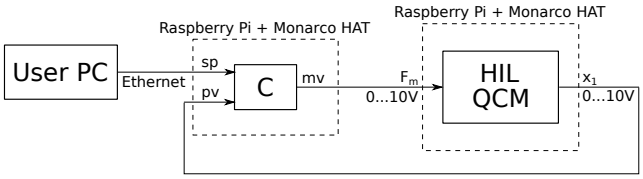

Fig. 5. Control loop - physical interconnection of the controller and HIL simulator

real-time, allowing trending of important variables and online tuning of parameters.

3.6 Phase 5: Processor-in-the-loop

The next stage is Processor-in-the-loop (PIL). Up to now, students had the whole control loop running in their computers. In this part, they have to move the whole SW application to the target hardware, implementing both the controller and the plant model. For this purpose, students get a second piece of the Raspberry Pi computer with Monarco HAT add-on board. This serves as a real-time controller device. This phase is important for understanding the practical limitations imposed by the target hardware. Stability, memory requirements and calculation time of the whole control application have to be evaluated before moving to next step.

3.7 Phase 6: Hardware-in-the-loop

This phase is closest to reality as it introduces the HIL simulator representing the real plant. The control software has been prepared on the target controller device in the previous step but the model of the controlled system has to be replaced by the HW simulator that is the best copy of the real system. The simulator is connected to the controller via physical inputs and outputs forming the whole control loop (Fig. 5, Fig. 1). The controller uses the position of the second mass from the quarter car model as the feedback variable via 0-10V analog signal. On the other side, the manipulated variable representing the actuator setpoint is transmitted to the HIL setup using a second analog channel. Wiring of the whole loop can be a part of the students' assignment as it forces them to fully understand the interconnections between the plant and controller. The analog IOs come with some inherent errors such as high-frequency noise and bias. Proper scaling of the signal is necessary to put the closed loop into operation. All these practical issues can be considered as the biggest advantage of this HIL process in comparison with purely numerical off-line simulations as it develops deeper understanding of implementation aspects encountered in real-life control engineering problems.

After HIL tests, experiments on real system usually follow. A physical quarter-car setup is not available at our university at the moment but we plan to construct it and introduce it to the course as the last step of the whole X-in-the-loop process. In an ideal case, the HIL setup will be only replaced by the physical device using the same inputs and outputs and the controller should work as expected, provided that all the previous design steps were performed correctly.

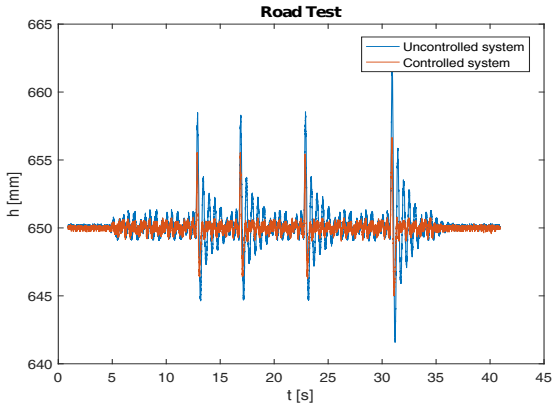

Fig. 6. Road profile test - comparison of open- and closed-loop performance

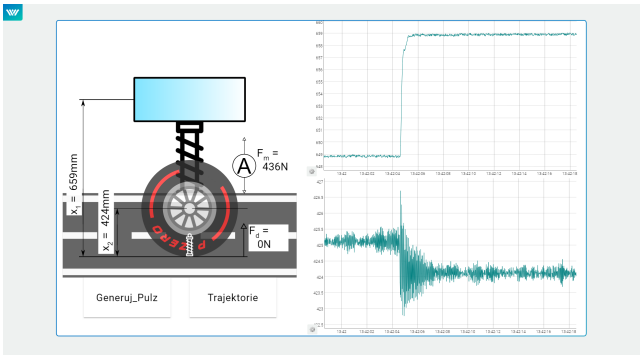

Fig. 7. Example of closed-loop simulator response observed at the HIL setup display

Students go through the whole procedure to learn how the control system development works in real applications and how to acquire good habits to succeed in the individual phases. They often find out that the controllers that worked well with the model in numeric simulations perform poorly with the physical plant represented by the HIL setup and there is still a lot of work. They need to go back to the previous phases and reconsider all the design choices made on the way.

The designed controllers are finally validated on a testing trajectory simulating a real road profile (Fig. 6). Students learn how to compare various controller designs using different time- and frequency-domain criteria. An example of the HIL simulator output provided via the touch-screen is shown in Fig. 7

4. CONCLUSION

The goal of the paper was to share our experience of introduction of HIL simulators in undergraduate control engineering course. It turns out that the extension of purely numerical simulations by a hands-on experience with physical hardware brings some significant advantages. First of all, the students are much more enthusiastic when allowed to play with physical gadgets. They are guided through the whole model-based control engineering process in the way close to the workflow typical for industrial practice. They get much deeper insight into what the control engineering is actually about and some important practical skills are acquired.

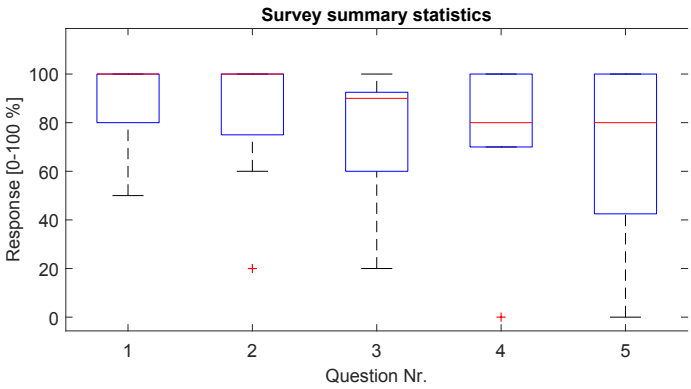

Fig. 8. Results of the survey conducted among students, blue box denotes the range between 25th and 75th percentiles, red line shows the median (middle quantile), the black dashed whiskers extend to minimum and maximum values, red crosses designate outliers

A survey was conducted among first 23 course graduates regarding the use of HIL setups. The students rated the validity of the following five statements on the scale of $0 - 100\%$:

(1) I like the idea of HIL setups in the undergraduate control engineering course
(2) I find the experiments with the setup more useful when compared to purely virtual simulations with offline numerical models
(3) The semester project helped me to understand the concept of X-IL cycles applied in the model-based control engineering
(4) I find the active suspension control problem attractive
(5) I would like to have more semester projects using the HIL setups in the forthcoming control courses

The summary statistics of the survey is given in Fig. 8 showing a clearly positive response. For the future work, we plan to extend the course further by introducing the physical quarter-car model which will close the whole X-IL cycle with real plant experiments. We are also thinking about implementing other types of systems and develop more advanced control problems suitable for subsequent graduate level control courses.

ACKNOWLEDGEMENTS

This work was supported from the H2020 ECSEL JU project I-MECH (Čech et al. (2019)) under grant agreement Nr. 737453 and from ERDF under project "Research and Development of Intelligent Components of Advanced Technologies for the Pilsen Metropolitan Area (InteCom)" No. CZ.02.1.01/0.0/0.0/17_048/0007267.

REFERENCES

Alvarez Sanchez, E. (2013). A quarter-car suspension system: Car body mass estimator and sliding mode control. *Procedia Technology*, 7, 208–214.

Despeisse, M. (2018). Games and simulations in industrial engineering education: A review of the cognitive and affective learning outcomes. *2018 Winter Simulation Conference*.

Gomes, L. and Bogosyan, S. (2010). Current trends in remote laboratories. *IEEE Transactions on Industrial electronics, 56*.

Hallinen, J. (2015). STEM education curriculum. *Encyclopaedia Britanica*.

Heradio, R., de la Torre Cubillo, L., and Dormido, S. (2016). Virtual and remote labs in control education: A survey. *Annual Reviews in Control, 42, 1-10*.

Lukovics, M. and Zuti, B. (2015). New functions of universities in the XXI. century. *University of Szeged*.

Ma, J. and Nickerson, J. (2006). Hands-on, simulated, and remote laboratories: A comparative literature review. *ACM Computing Surveys 38(3), pp. 1*.

Parodi, O., Bibuli, M., Lapierre, L., and Caccia, M. (2009). Missions preparation and design of new algorithms for Charlie and Taipan within Thetis, a HIL simulator. *IFAC Proceedings Volumes*, 42(18), 173 – 178. 8th IFAC Conference on Manoeuvring and Control of Marine Craft.

Perumal, K. and Ganesan, R. (2018). Integration of numerical and physical experiments to enhance student learning experience. *Computer Applications in Engineering Education*.

Rahmani, B. and Hashemi, S.R. (2015). Internet-based control of fcu hardware-in-the-loop simulators. *Simulation Modelling Practice and Theory*, 56, 69 – 81.

REX Controls, s.r.o. (2016). Monarco HAT add-on board for the Raspberry Pi. URL `http://www.monarco.io`.

REX Controls s.r.o. (2019a). URL `https://www.rexcontrols.cz`.

REX Controls s.r.o. (2019b). Rexygen - programming automation devices without hand coding. URL `http://www.rexygen.com`.

Smith, P. and Pollard, D. (1986). The role of computer simulations in engineering education. *Computers and education*.

Smolinski, E., Benkmann, A., Westerhoff, P., Nguyen, M.V., Drewelow, W., and Jeinsch, T. (2017). A hardware-in-the-loop simulator for the development of medical therapy devices. *IFAC-PapersOnLine*, 50(1), 15050 – 15055. 20th IFAC World Congress.

Sobota, J., Goubej, M., Königsmarková, J., and Čech, M. (2019). Raspberry Pi-based control education. *IFAC Advances in control education*.

The Raspberry Pi Foundation (2018). Raspberry Pi 3 model b+. URL `http://www.raspberrypi.org`.

Uribe, M., Magana, A., Bahk, J.H., and Shakouri, A. (2016). Computational simulations as virtual laboratories for online engineering education: A case study in the field of thermoelectricity. *Computer Applications in Engineering Education 24(3), pp. 428-442*.

Čech, M., Königsmarková, J., Goubej, M., Oomen, T., and Visioli, A. (2019). Essential challenges in motion control education. *IFAC Advances in control education*.

Venkatalakshmi, B., Balakrishnan, R., Saravanan, V., and Renold, A. (2016). Impact of simulation softwares as teaching tools in engineering learning - an instructional design choice. *IEEE Global Engineering Education Conference*.

Čech, M., Beltman, A., and Ozols, K. (2019). I-MECH – smart system integration for mechatronic applications. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*.