

Research Article

Integrating ROS and IoT in a Virtual Laboratory for Control System Engineering

Claudio Urrea  and John Kern 

Grupo de Automática, Department of Electrical Engineering, Universidad de Santiago de Chile, Chile

Correspondence should be addressed to Claudio Urrea; claudio.urrea@usach.cl

Received 1 April 2020; Revised 27 September 2020; Accepted 10 November 2020; Published 18 November 2020

Academic Editor: Sazzad Hossien Chowdhury

Copyright © 2020 Claudio Urrea and John Kern. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article presents the implementation of a learning environment for the teaching of control systems. This environment integrates physical equipment and simulation, monitoring, and control through a network. A software platform based on ROS (Robotic Operating System) grants access to the system through intranet and Internet, facilitating the integration of new test equipment. The environment developed can be used in didactic experiences both inside and outside the classroom, enhancing the learning of four main study topics: modeling, analysis, parameter estimation, and controller design.

1. Introduction

Automatic control systems are essential in modern life. Air conditioning at workplaces as well as the provision of drinking water and energy, transportation, communications, and the most varied industrial processes all require automatic control systems to achieve efficiency, reliability, and safety. Hence, it is important to study these systems in several engineering fields.

The design and implementation of control systems is a complex process that requires advanced algebra and calculus, in addition to data analysis, mathematical modeling, and programming skills. In engineering undergraduate programs, traditional systems are studied from a mathematical perspective and students are given tasks to compare theoretical results and measurements in real or simulated systems. In these tasks, well-designed equipment is key for demonstrations and didactic activities [1–6].

The system theory identifies four main topics that define the contents of a study plan for control systems: modeling, analysis, parameter estimation, and controller design. Modeling is a critical step in the application of system theory to real processes, as it consists of finding a mathematical model that describes a physical situation accurately. During this step, the system limits and variables need to be chosen based on

previous knowledge and assumptions about the model uncertainties. After model definition, the behavior of the system is analyzed in terms of stability, sensitivity, identifiability, and observability. Parameter or status estimation takes place after obtaining an identifiable and observable model. Here, the estimation of unknown variables is addressed from an input/output data set. Finally, different control strategies are studied focusing on the calculation of a specific input in such a way that the controlled system displays a desired behavior [7].

To develop the skills necessary to carry out these activities, it is necessary to work with mathematical analysis tools based on empirical data obtained from real systems or computer simulations. Consequently, the introduction of technology to the classroom is necessary to teach control systems.

Research has demonstrated that the introduction of technology to the classroom in any discipline is not an easy matter, but a process that should take into account the students' specific needs and expectations, the methodological and curricular decisions, and the resistance to change from teachers, among other factors [8–10]. In engineering education, it should also be considered that the training received by future engineers should prepare them for the challenges of the future, that is, to face rapid technological advances and critical stakes more than ever [11]. All the above makes it necessary for students to develop learning strategies that

bring them closer to the solution for real problems and to the technologies of the real world [12].

The purpose of engineering is to manipulate the materials, energy, and energy available to solve the problems of society. Therefore, the training of engineers historically has demanded practical laboratory work to develop these skills, incorporating technology as a tool [13]. Regarding the teaching of control systems, different experiences have proved that making students integrate current simulation technologies and interaction with conventional technologies is the most effective way of learning [2, 14]. In addition, it is also essential to address motivational issues because the lack of didactic learning might influence the dropout of engineering students to a great extent [15].

Several educational institutions have contributed with valuable experiences in the design of educational equipment and software tools for the teaching of control systems. These tools can be used for both face-to-face lessons and learning experiences out of the classroom, such as virtual laboratories (remote access to simulated systems) and remote laboratories (remote access to real equipment). To teach control systems in face-to-face environments, software packages such as MatLab or LabView have been the preferred option, since they contain several libraries and specialized tools (toolboxes) that even allow users to combine simulation environments with physical laboratory equipment [16–19].

For example, in [20], a deep learning-based pipeline smart system is developed, which is able to recognize block diagrams from control systems with handwritten feedback control architectures in a whiteboard and transforms them automatically into simulation diagrams in the MatLab software. In this way, the control system is visualized, executed, and analyzed in an interactive way using MatLab. In [21], a new teaching method for robot control systems called AVPC (Algorithm, Virtual experiment, Programming, and Controller) is presented. This method shows students the stages considered during the development, verification, and application of robot algorithms, taking a typical robot with 6 DoF as a teaching case. The solutions for the algorithms are obtained by means of MatLab, and then, the dynamic model of the arm is exported to the V-REP (Virtual Robot Experimentation Platform) software (currently denominated CoppeliaSim [22]). In [23], a teaching methodology that considers system model-oriented design methods for mechatronic systems is employed to address the essential concepts of automatic control. Control algorithms are created using library modules that are installed in MatLab-Simulink. Subsequently, the executable code is generated through compilers, to then be implemented in the DSP-controllers TMS and SRM32.

Nevertheless, using these packages out of the classroom is limited by license restrictions and difficulties in working with several devices in a network. To overcome these obstacles, some experimental platforms based on the Robotic Operating System (ROS) have been developed over the last years [24–27].

ROS is particularly useful as an educational platform for the teaching of control systems due to three main characteristics: (1) It is a free open code software, very stable, and with great support from a very active community. (2) It has a large

number of libraries available, which facilitate the development of programs in different languages and support the interconnection of multiple sensors and actuators. (3) It has excellent tools for visualizing graphs and 2D and 3D simulations. Additionally, the implementation of ROS in service allows access to online resources by means of intranet or Internet. This way, ROS has gained acceptance in the robotics and automation field at both the research and educational environment levels [26, 27].

The main advantage of the approach proposed in this work is that students do not need to be physically present in an educational center laboratory, nor do they need a software license that sometimes may be difficult to obtain due to the related costs. Using free software and low-cost hardware elements, students can expand their knowledge in control systems through simulation and also through interaction with a real physical system from their computers, employing tools for real-time data acquisition and graph visualization as well.

Below, we present the implementation of a ROS-based learning environment for the study of digital control strategies in control system laboratories. The discrete equations modeling the system are presented, and the way in which ROS interacts with the system to create a simulation and control environment is explained. In addition, a didactic sequence is presented to address the four main points in control systems by means of methodological strategies that combine real equipment and simulation with ROS.

2. Description of the System under Study

A learning environment has been developed to study the well-known linear inverted pendulum system, which is shown in Figure 1. The purpose of the system is to keep the arm vertical while the base linear movement causes the angular movement of the pendulum.

An arm of l length concentrates all the mass m on its end. The τ torque on the arm causes angular acceleration $\ddot{\theta}$, which will initiate the arm motion at a $\dot{\theta}$ angular speed and change its θ angular position. Figure 2 shows the essential variables involved in the system. The base linear movement needs to be controlled along the x -axis to keep the arm in a vertical position, that is, to compensate the angular movement θ through the motion of the base, which equals $\Delta x = -l \cdot \cos \theta$.

Given $\tau[t]$ as a net torque at instant t and applied to the m mass arm, the angular acceleration $\ddot{\theta}[t]$ is described by

$$\ddot{\theta}[t] = \frac{\tau[t]}{m} \quad (1)$$

Knowing the angular acceleration $\ddot{\theta}[t]$, the angular speed $\dot{\theta}[t]$ at instant t can be estimated as $\dot{\theta}[t] = \dot{\theta}[t-1] + \ddot{\theta}[t] \cdot \delta T$, where $\dot{\theta}[t-1]$ is the angular speed at a previous moment, $\ddot{\theta}[t]$ is the angular acceleration at the current moment, and δT is the time interval passed. In addition, the angular position $\theta[t]$ at instant t can be estimated as $\theta[t] = \theta[t-1] + \dot{\theta}[t] \cdot \delta T$, where $\theta[t-1]$ is the angular position at a previous moment, $\dot{\theta}[t]$ is the angular speed at the current moment, and δT is the time interval passed.

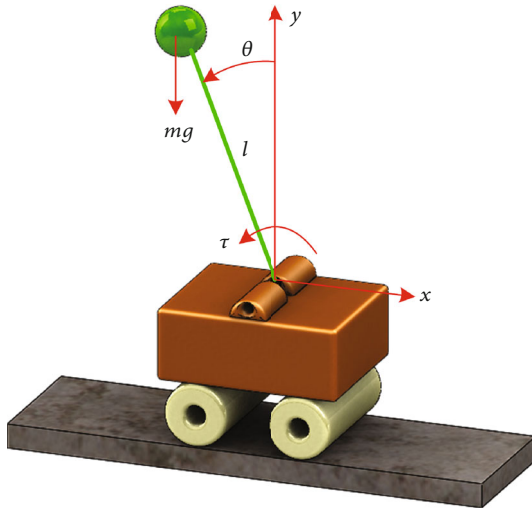


FIGURE 1: Inverted pendulum system.

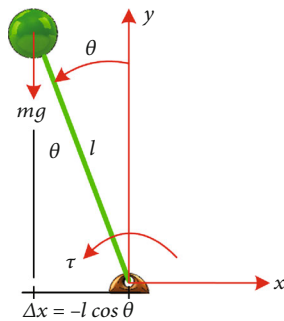


FIGURE 2: Variables involved in the inverted pendulum model.

Working with these equations, the discrete model that describes the system dynamics is

$$\theta[t] - 2\theta[t-1] + \theta[t-2] = (\delta T)^2 \frac{\tau[t]}{m}. \quad (2)$$

This second-order linear difference equation describes the evolution over discrete time t at δT intervals of the arm's angular position $\theta[t]$ as a result of torque $\tau[t]$ action over an m mass.

In the system under study, torque τ applied to the arm is the result of the gravity force F_g , which is counteracted by the dynamic friction force.

For any time instant t , the torque generated by the net force $F_t[t]$ which acts on the end of the l length and m mass pendulum can be assessed through the following equation:

$$\tau[t] = \frac{F_t[t]}{l} = \frac{1}{l} \left(mg \cdot \text{sen}\theta[t] - \mu \cdot \dot{\theta}[t] \right), \quad (3)$$

where $mg \cdot \text{sen}\theta[t]$ is the gravity force component perpendicular to the arm, μ is the friction coefficient, and $\mu \cdot \dot{\theta}[t]$ is the friction force that opposes arm movement.

Additionally, the movement of the base causes an angular movement defined by $\Delta x = -l \cdot \cos \theta$. If the linear movement of the base can be expressed in a way approximately proportional to the voltage applied to motor V_m , that is, $\Delta x = k \cdot V_m \cdot \delta T$, then the angular movement caused by voltage V_m can be written as $\Delta \theta = \cos^{-1} \left(k \cdot V_m \cdot \delta T / l \right)$.

Considering the above relations, the evolution of the system can be estimated based on the positions of the two previous instants:

$$\begin{aligned} \theta[t] = & \frac{(\delta T)^2}{ml} \left(mg \cdot \text{sen}\theta[t-1] - \mu \cdot \frac{\theta[t-1] - \theta[t-2]}{\delta T} \right) \\ & + 2\theta[t-1] - \theta[t-2] - \cos^{-1} \left(\frac{k \cdot V_m \cdot \delta T}{l} \right). \end{aligned} \quad (4)$$

Equation (4) allows us to study how the angular position of the pendulum arm, $\theta[t]$, evolves as a result of changing the voltage $V_m[t]$ applied to the motor in order to move the pendulum's base. This base represents a second-order non-linear system used to simulate the system. Since the mass and the friction coefficient have small values, the system has quick dynamics that allow us to observe instantly the effect of modifications to the controller parameters as well as the effect of the perturbations. All these characteristics make ROS suitable for the study of control systems.

The inverted pendulum system has been typically studied in control system subjects, and currently, learning experiences continue to be developed, including control techniques and modern technologies [28–34]. The learning environment ROS also facilitates the study of these new techniques, as modifications to the model and controller design are made using open source software libraries with broad support [35].

3. Description of the Learning Environment

To implement a learning environment that facilitates the study of the inverted pendulum system, physical components and the ROS software environment are integrated by means of a data acquisition and a control interface developed by Arduino, which is connected to the intranet as a node. This configuration enables easy integration with ROS, which, in turn, facilitates the monitoring of the pendulum position and the motor's electric variables, as well as the issuing of control orders to move the base. The communication between the physical system and ROS provides students with an environment that facilitates system modeling, analysis, and parameter estimation. This allows students to manipulate the physical system from their laptop and use some tools for real-time data acquisition and graph visualization.

To facilitate the controller design stage, a simulation node that uses the 3D RViz visualization tool from ROS was developed. Using the simulation node, students can test their solutions and observe the system's behavior utilizing graphs and 3D models before implementing them in the real system (see Figure 3).

In the simulation environment, students should implement their controller over the `pd_controller` node, which interacts

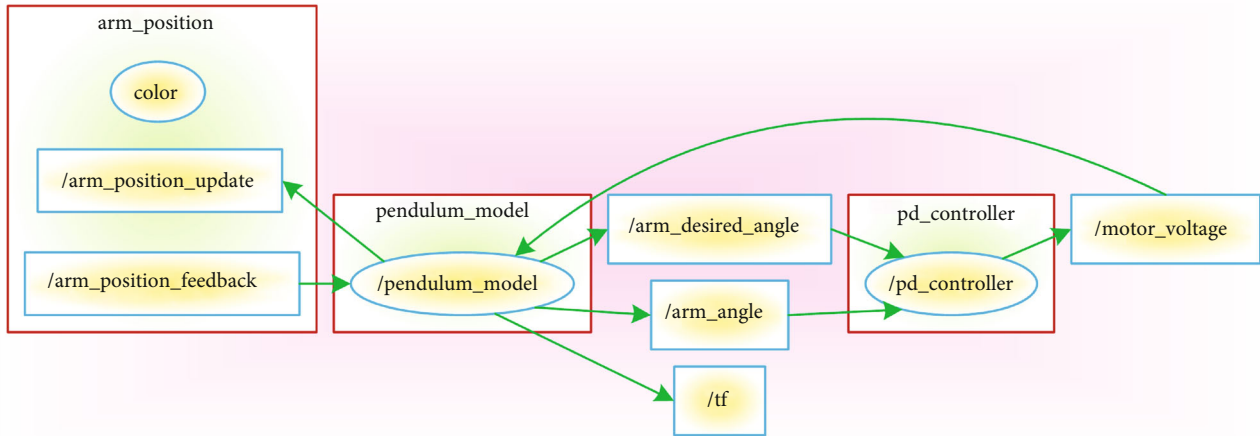


FIGURE 3: Nodes and topics associated with the simulation environment in ROS.

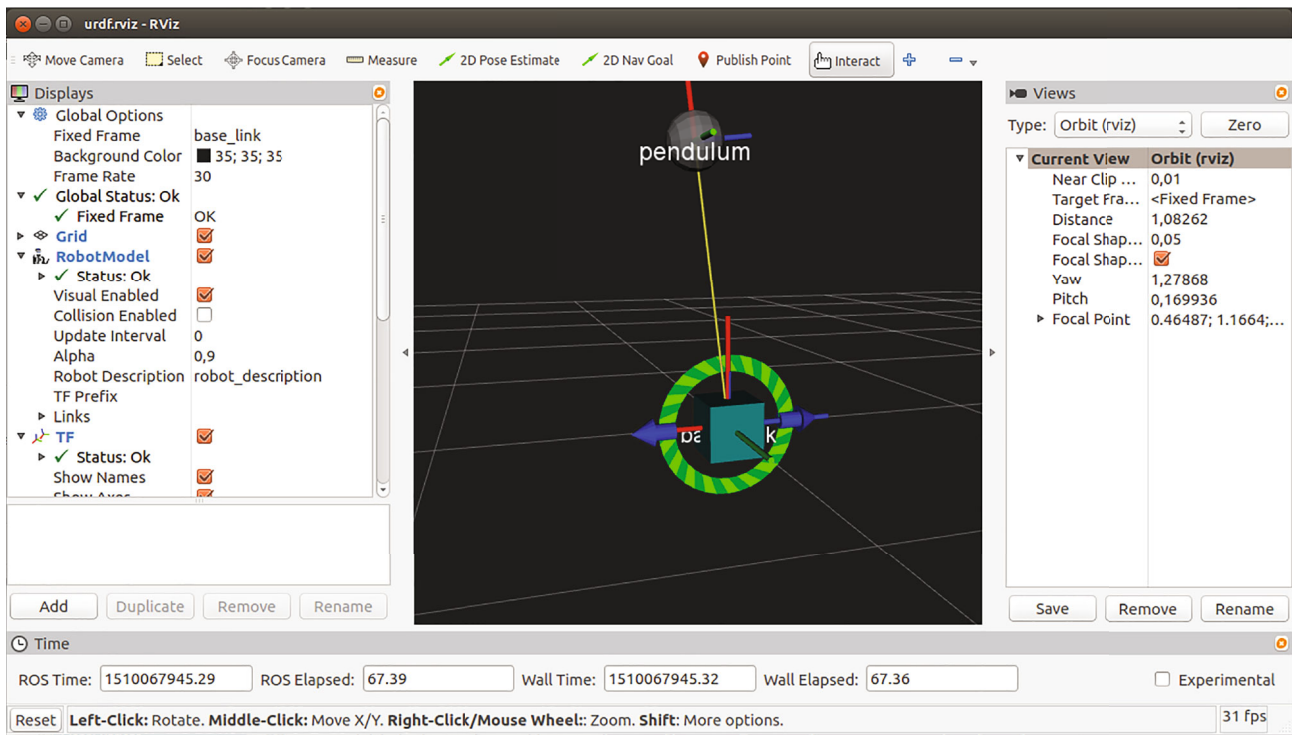


FIGURE 4: Image of the simulation system in RViz.

with RViz to modify the angular position and visualize the pendulum movement. In this way, students can observe the system's behavior with different types of controllers and different controller parameters. The interaction is intuitive and realistic and allows them to explore in a safe environment, as shown in Figure 4.

The ROS environment also includes the data acquisition tool `rqtplot`, through which graphs can be created to analyze system behavior. Using this tool, students can choose the variables of interest and build graphs or record data that can be subsequently used in other data analysis environments. Additionally, it is possible to acquire data and create graphs

from both the real system and the simulation environment (see Figure 5).

Considering that ROS is an open software environment, it also allows us to develop personalized modules for specific tasks, which can be written by the same students in C++ and Python.

4. Methodological Proposal

The use of the learning environment proposed allows students to address the four essential topics of the control theory in an integrated and sequential fashion, as described as follows:

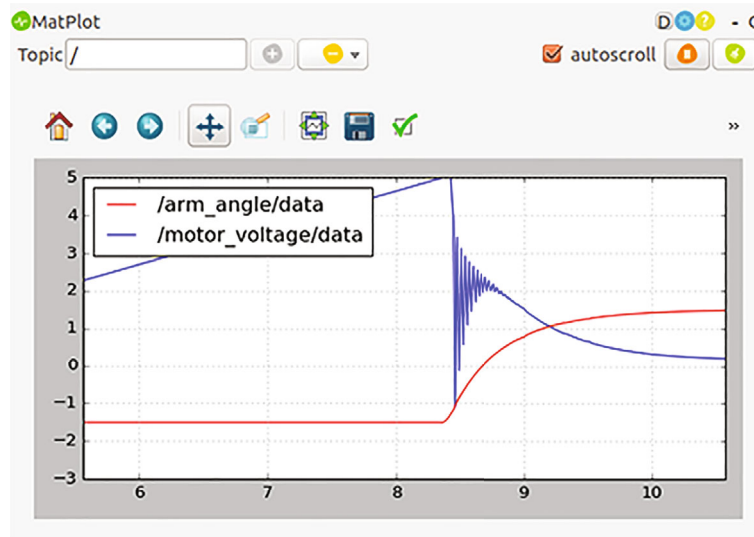


FIGURE 5: Graphs for position vs. time and motor vs. time obtained from a rqtplot tool when lifting a pendulum from resting position.

- (a) *Modeling*. Students start by identifying and applying the equations that govern the rotation of a solid body to obtain the dynamics of the system from a general equation. Through interaction with the physical system, they identify the model elements and discriminate between input variables, output variables, and constant values.
- (b) *Analysis*. From a model, students explore the effects of modifying some constant variables relevant to the system by means of simulation. Then, they analyze the system's response to different input values. The output values of a nonlinear model are also compared with those of a linear model.
- (c) *Estimation*. In this section, the requirements for determining the formulation parameters are studied, obtaining the model constant values from measurements in the real system. Then, students compare the simulated answer to the real, determining their model's validity and limitations.
- (d) *Control*. From a validated model, students can test different control techniques experimentally in a simulation and real environment. As a starting point, they distinguish between open and close control loops to then design and tune a PID controller able to keep the arm in the desired position.

5. Conclusions

The software package ROS, a middleware well known in robotics, has been used to implement a learning environment for the teaching of control systems in engineering programs. The environment proposed allows students to work in connection with the physical system from the classroom and to share learning experiences out of the classroom in a simulation environment. The characteristics of the platform developed make it possible to extend its use to virtual (remote

access to simulated systems) and remote (remote access to physical systems) laboratories.

The use of this learning environment in an automatic control subject allows us to address in an integrated way the four basic topics of control theory, which are modeling, analysis, estimation, and control. To this end, multiple data acquisition tools from ROS are employed, allowing the creation of modules utilizing programming in C++ or Python.

In the current development stage, the system has been implemented in a local server, enabling access to resources from the intranet. From this work, it is possible to extend the functions of this learning environment by adding new test systems and granting access to physical resources from locations out of the classroom through the Internet.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Vicerrectoría de Investigación, Desarrollo e Innovación of the Universidad de Santiago de Chile, Chile.

References

- [1] M. S. Ayas and I. H. Altas, "A virtual laboratory for system simulation and control with undergraduate curriculum," *Computer Applications in Engineering Education*, vol. 24, no. 1, pp. 122–130, 2016.
- [2] P. Twigg, P. Ponnappalli, and M. Fowler, "Workshop problem-solving for improved student engagement and increased

- learning in engineering control,” *International Journal of Electrical Engineering Education*, vol. 55, no. 2, pp. 120–129, 2018.
- [3] C. Urrea and D. Saa, “Design and implementation of a graphic simulator for calculating the inverse kinematics of a redundant planar manipulator robot,” *Applied Sciences*, vol. 10, no. 19, article 6770, 2020.
 - [4] M. Antonio-Cruz, V. M. Hernández-Guzmán, R. Silva-Ortigoza, and G. Silva-Ortigoza, “Implementation of a controller to eliminate the limit cycle in the inverted pendulum on a cart,” *Complexity*, vol. 2019, Article ID 8271584, 13 pages, 2019.
 - [5] C. Urrea and R. Matteoda, “Development of a virtual reality simulator for a strategy for coordinating cooperative manipulator robots using cloud computing,” *Robotics and Autonomous Systems*, vol. 126, article 103447, 2020.
 - [6] C. Urrea and J. Pascal, “Parameter identification methods for real redundant manipulators,” *Journal of Applied Research and Technology*, vol. 15, no. 4, article 103447, 2019.
 - [7] N. Spolaôr and F. B. V. Benitti, “Robotics applications grounded in learning theories on tertiary education: a systematic review,” *Computers & Education*, vol. 112, pp. 97–107, 2017.
 - [8] M. Bjelica and M. Simić-Peجویć, “Experiences with remote laboratory,” *International Journal of Electrical Engineering Education*, vol. 55, no. 1, pp. 79–87, 2018.
 - [9] R. Estriegana, J.-A. Medina-Merodio, and R. Barchino, “Student acceptance of virtual laboratory and practical work: an extension of the technology acceptance model,” *Computers & Education*, vol. 135, pp. 1–14, 2019.
 - [10] A. Dominguez, H. Alarcón, and F. J. García-Peñalvo, “Active learning experiences in engineering education,” *International Journal of Electrical Engineering Education*, vol. 35, no. 1, pp. 305–309, 2019.
 - [11] N. M. Fonseca Ferreira and E. D. C. Freitas, “Computer applications for education on industrial robotic systems,” *Computer Applications in Engineering Education*, vol. 26, no. 5, pp. 1186–1194, 2018.
 - [12] M. M. Pastor Artigues, F. Roure Fernández, M. Ferrer Ballester et al., “Learning in engineering through design, construction, analysis and experimentation,” *International Journal of Electrical Engineering Education*, vol. 35, no. 1, pp. 372–384, 2018.
 - [13] M. Yamauchi, T. Tanioka, Y. Masui et al., “Early education and its effect on students who desire to research their own topics before joining laboratories,” *IEEE Transactions on Electrical and Electronic Engineering*, vol. 13, no. 8, pp. 1092–1098, 2018.
 - [14] W. Yu and X. Li, “A magnetic levitation system for advanced control education,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 9032–9037, 2014.
 - [15] L. Salas-Moreira, A. C. Molina, J. L. Olmedilla, L. García-Hernández, and J. M. Palomo-Romero, “Factors affecting engineering students dropout: a case study,” *International Journal of Engineering Education*, vol. 35, no. 1, pp. 156–167, 2019.
 - [16] C. Urrea and D. Venegas, “Design and development of control systems for an aircraft. Comparison of performances through computational simulations,” *IEEE Latin America Transactions*, vol. 16, no. 3, pp. 735–740, 2018.
 - [17] S. Khan, M. H. Jaffery, A. Hanif, and M. R. Asif, “Teaching tool for a control systems laboratory using a quadrotor as a plant in MATLAB,” *IEEE Transactions on Education*, vol. 60, no. 4, pp. 249–256, 2017.
 - [18] T. Tamai, Y. Shoda, and K. Kawada, “Development of teaching material for the motivation of “measurements and controls” in technology education,” *IEEE Transactions on Electrical and Electronic Engineering*, vol. 13, no. 6, pp. 932–933, 2018.
 - [19] A. J. Magana and T. de Jong, “Modeling and simulation practices in engineering education,” *Computer Applications in Engineering Education*, vol. 26, no. 4, pp. 731–738, 2018.
 - [20] D. Erdem, A. Beke, and T. Kumbasar, “A deep learning-based pipeline for teaching control theory: transforming feedback control systems on whiteboard into MATLAB,” *IEEE Access*, vol. 8, pp. 84631–84641, 2020.
 - [21] D. Zhou, M. Xie, P. Xuan, and R. Jia, “A teaching method for the theory and application of robot kinematics based on MATLAB and V-REP,” *Computer Applications in Engineering Education*, vol. 28, no. 2, pp. 239–253, 2020.
 - [22] A. G. Coppelia Robotics, “Robot simulator CoppeliaSim: create, compose, simulate, any robot - Coppelia Robotics,” September 2020, <https://www.coppeliarobotics.com/>.
 - [23] R. S. Gavrylov and Y. N. Mystafaev, “Methodology for teaching model-oriented design of mechatronic systems,” *AIP Conference Proceedings*, vol. 2195, no. 1, article 020031, 2019.
 - [24] D. Galán, E. Fabregas, G. Garcia et al., “Online virtual control laboratory of mobile robots,” *IFAC - Papers On Line*, vol. 51, no. 4, pp. 316–321, 2018.
 - [25] T. Sato, N. Kawaguchi, S. Nakatani, N. Araki, and Y. Konishi, “Control education efficacy evaluation through flight control experiment,” *IFAC-Papers on Line*, vol. 49, no. 6, pp. 1–3, 2016.
 - [26] R. Awad, G. Heppner, A. Roennau, and M. Bordignon, “ROS engineering workbench based on semantically enriched app models for improved reusability,” in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, Berlin, Germany, September 2016.
 - [27] K. Takaya, T. Asai, V. Kroumov, and F. Smarandache, “Simulation environment for mobile robots testing using ROS and Gazebo,” in *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 96–101, Sinaia, Romania, October 2016.
 - [28] X. Sun and X. Fu, “Synchronization of two different dynamical systems under sinusoidal constraint,” *Journal of Applied Mathematics*, vol. 2014, Article ID 341635, 9 pages, 2014.
 - [29] J. Lee, R. Mukherjee, and H. K. Khalil, “Output feedback stabilization of inverted pendulum on a cart in the presence of uncertainties,” *Automatica*, vol. 54, pp. 146–157, 2015.
 - [30] D. Du, W. Li, B. Zhan, M. Fei, and T. Yang, “Experimental performance analysis of inverted pendulum platform,” in *Theory, Methodology, Tools and Applications for Modeling and Simulation of Complex Systems. AsiaSim 2016, SCS AutumnSim 2016. Communications in Computer and Information Science*, vol. 645, L. Zhang, X. Song, and Y. Wu, Eds., Springer, Singapore, 2016.
 - [31] T. Sato, H. Tajika, R. Vilanova, and Y. Konishi, “Adaptive PID control system with assigned robust stability,” *IEEE Transactions on Electrical and Electronic Engineering*, vol. 13, no. 8, pp. 1169–1181, 2018.
 - [32] A. Tiga, C. Ghorbel, and N. B. Braiek, “Nonlinear/linear switched control of inverted pendulum system: stability analysis and real-time implementation,” *Mathematical Problems in Engineering*, vol. 2019, Article ID 2391587, 10 pages, 2019.
 - [33] S. Hu and Y. Sun, “Mixed optimal scheduling model of flexible service system based on inverted triangle,” *Journal of Applied Mathematics*, vol. 2019, Article ID 2674697, 20 pages, 2019.

- [34] Y. Niu and Y. Liang, "An event-triggered robust control and bandwidth scheduling codesign approach for networked control systems with uncertain time delays," *Mathematical Problems in Engineering*, vol. 2019, Article ID 3596784, 11 pages, 2019.
- [35] H. Wei, Z. Shao, Z. Huang et al., "RT-ROS: a real-time ROS architecture on multi-core processors," *Future Generation Computer Systems*, vol. 56, pp. 171–178, 2016.