MODEL-BASED DESIGN OF A BALLBOT SYSTEM CONTROLLED BY MINDSTORMS EV3

Ngoc Huy Duong, Minh Quan Dinh, Pham Anh-Duc*

The University of Danang - University of Science and Technology, Vietnam

*Corresponding author: ducpham@dut.udn.vn

(Received: November 07, 2025; Revised: February 12, 2026; Accepted: February 14, 2026)

DOI: 10.31130/ud-jst.2026.24(1).617E

Abstract - Translating theoretical control principles into practical implementation remains a central challenge in control system engineering. Model-Based Design (MBD) has emerged as a crucial methodology to address this problem, especially when utilizing accessible platforms like LEGO Mindstorms EV3. This paper presents the MBD workflow for modeling and control of dynamic systems on the LEGO EV3 hardware, starting with DC motor position, speed control using a PID controller, and then advancing to a real-time balancing of a complex underactuated Ballbot system using an LQR controller. The MBD workflow includes system identification, model formulation, controller simulation, and tuning in MATLAB/Simulink, automatic code deployment to the EV3 brick and data acquisition for further analysis. Results show that this approach effectively reduces development effort, clarifies the link between theory and implementation, and highlights its broad applicability for systems of varying complexity, from simple motor control to advanced robotics.

Key words – Model-based design; Ballbot; Mindstorms EV3.

1. Introduction

Nowadays, simulations and experiments play a crucial role in the development and validation of control systems [1-5]. Popular simulation software, such as MATLAB/Simulink, LabVIEW, and Gazebo, allows for the accurate modeling of objects and control algorithms prior to experimental implementation, thereby saving time and costs [1, 6]. However, there is a gap between simulations and experiments due to model discrepancies, real-world noise, and other factors, making tuning and trial-and-error during experiments an unavoidable necessity [7]. Furthermore, the investment costs associated with the experimental evaluation of systems may pose a barrier to users.

Consequently, the current trend is to closely integrate simulation and experimentation through Model-Based Design (MBD) techniques, which enable simulation, refinement, code generation, and direct deployment of controllers onto hardware [8]. Some popular platforms, such as Arduino, STM32, and Raspberry Pi, are utilized because of their flexibility and diverse applications; however, they require users to engage deeply in hardware setup and programming, thereby increasing the complexity for the user [9]. On the other hand, specialized platforms such as dSPACE or Speedgoat support MBD at an industrial level through Hardware-in-the-Loop (HIL) [4] and Rapid Control Prototyping (RCP) [10], making them suitable for complex systems such as automotive and aerospace applications [11].

In this context, LEGO Mindstorms EV3 presents an ideal choice for education and basic research in the field of automation control [12]. Its synchronized hardware ecosystem, intuitive programming interface, and integration capabilities with Simulink via the official MathWorks support package allow learners to fully engage in the MBD process-from modeling and simulation to experimental implementation-without the need for extensive embedded programming experience. Furthermore, LEGO Mindstorms EV3 excels in automated control with its versatile programming capabilities, robust sensors, and user-friendly interface, enabling users to create complex and dynamic robotic systems [13]-[15].

This paper focuses on the practical implementation of MBD workflow, demonstrated through the modeling and control of both a single motor system and a complex Ballbot system using the LEGO Mindstorms EV3 platform. Firstly, the characteristics of LEGO Mindstorms EV3 are described. Subsequently, the automated control system, particularly the control system of the ball-balancing robot (Ballbot) using the Mindstorms EV3, is introduced. This automated control system is oriented towards a model-based design approach to illustrate the connectivity between simulation and experimentation. Consequently, the simulation and experimental evaluation results of the Ballbot system, controlled by the Mindstorms EV3, were compared.

2. Overview of control system with Mindstorms EV3

The closed-loop control system, as shown in Figure 1, requires central control systems to manage the output response based on input signals and feedback from sensors. The control system using Mindstorms EV3 is constructed in a similar manner. In this system, the Intelligent Brick serves as the central component. This block utilizes the TI Sitara AM1808 ARM9 microprocessor (MPU) along with the EV3 firmware operating system, which is built on a Linux kernel, functioning at a maximum clock speed of 375 MHz and integrated with various peripherals such as USB, Wi-Fi, speakers, and LCD screens. In addition to the input that directly affects the operating model through the available functions in the EV3 Intelligent Brick, the input signal of the system can be received from sensor signals via sensor ports 1÷4. These ports support various common communication standards, such as Analog, Digital, UART, and I2C, enabling communication with a diverse range of sensors (for example: touch sensor, color sensor, ultrasonic sensor, etc). Some smart sensors, as shown in

Figure 2, are integrated with 8-bit MCUs for preliminary signal processing, which are then transmitted to a computer and MATLAB/Simulink for customized processing, resulting in a complete output signal. To enhance obstacle detection capabilities, the Camera-Pixy2 component can be integrated into the system. The output signal from the Mindstorms EV3 is directly connected to the actuators (DC and servo motors) via RJ12 cables to supply power and perform tasks as required by the controller. Additionally, the output signal parameters can be displayed on the LCD screen. Furthermore, the processing blocks receive signals through bus protocols such as UART, I2C, TCP/IP, and UDP, as shown in Figure 3, allowing this system to utilize sensors from various manufacturers or connect with commonly available control boards, such as Arduino and Raspberry Pi. The EV3 system also supports real-time control owing to the built-in timer block, which facilitates the generation of logic signals, transmission interrupts, and real-time feedback. Moreover, to support higher-level research and development, a module for Mindstorms EV3 in MATLAB has been developed to assist in programming language conversion and ensure compatibility with the ARM9 AM1808 processor platform.

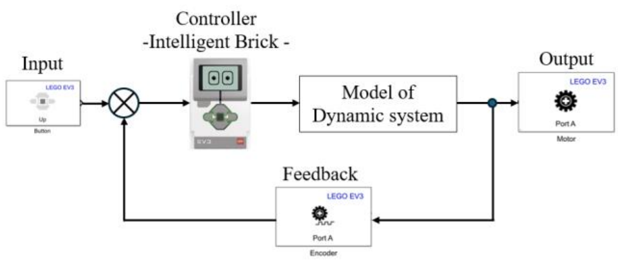


Figure 1. *Block diagram of control system with Mindstorms EV3*

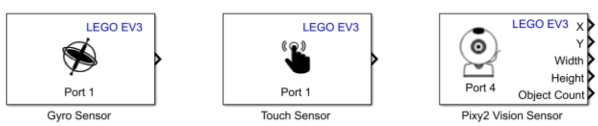


Figure 2. *Smart sensor blocks of LEGO Mindstorms EV3*

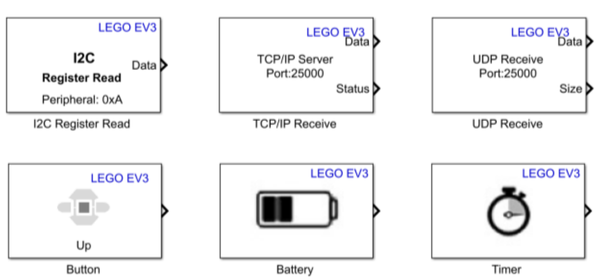


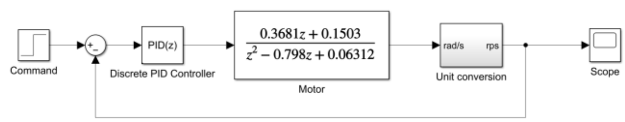
Figure 3. *Additional input blocks of Mindstorms EV3*

3. Motor control with *Mindstorms* EV3

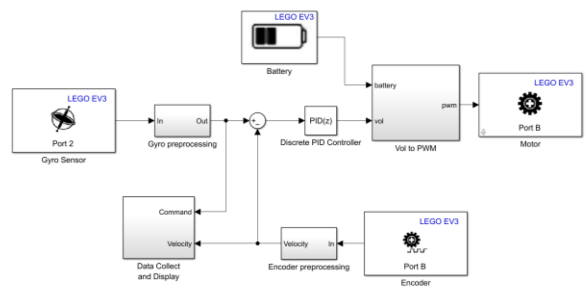
To process the control design, a preliminary system identification is conducted to estimate motor's electromechanical parameters through various experiments. However, high accuracy is not mandatory at this stage, as the MBD workflow refines these initial estimation errors through HIL testing and iterative experimental tuning.

$$G(s) = \frac{0.305}{3.10^{-6}s^2 + 0.002072s + 0.156} \quad (1)$$

The motor continuous transfer function, as obtained in Eq. (1), is discretized at system's sampling rate $T_s = 0.004$s yielding the motor discrete-time transfer function as shown in Figure 4(a). Consequently, the PID control system for the motor, as shown in Figure 4, is described in accordance with the motor's transfer function in the 'Motor' block. The PID coefficients (K_P = 0.2, K_I = 0.12, K_D= 0.05) were validated experimentally and compared with the simulation results, as illustrated in Figure 5. These PID coefficients were selected to align with practical experimental conditions, such as the effects of inertia, friction, system noise, and noise filtering.

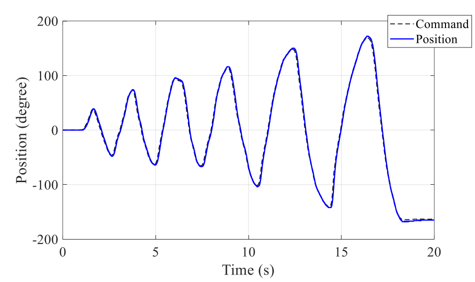


(a) Closed-loop control model of motor with transfer function

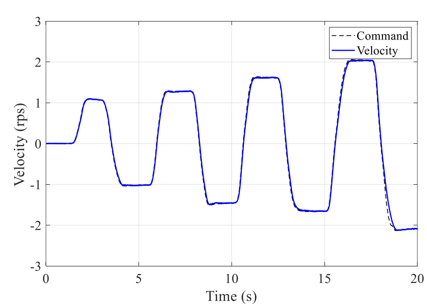


(b) HIL integration model of motor with Mindstorms EV3

Figure 4. *Block diagram for motor modeling and HIL integration*



(a) Position



(b) Velocity

Figure 5. *Experimental response of Motor control system with Mindstorms EV3*

Moreover, Figure 5 shows that the system response is relatively rapid and accurate with minimal errors, and the response of the system can be gradually adjusted according to the amplitude of the control command. Therefore, whenever there is an abrupt adjustment, the control system demonstates stability, with limited overshoot that could negatively impact the system. From the motor control system, it can be suggested that Mindstorms EV3 has powerful capabilities for simulation and experimentation with control systems that have high practical applications. As a result, complex models can be optimally calibrated in a simulation environment before experimental deployment, making the embedding process and achieving the desired response more efficient and accurate. To validate this assertion, simulations and experiments were conducted on the Ballbot system, which has a more complex structure.

4. Control system of the Ballbot with Mindstorms EV3

The Ballbot illustrated in Figure 6 is a system capable of achieving a stable upright balance by controlling the motion of the ball in contact with the ground. This is a classical modern control problem with significantly higher complexity than traditional systems, such as inverted pendulums and two-wheeled balancing robots. Research and development of the Ballbot [16]-[18] provide an opportunity to engage practically with the core issues of modern control, ranging from nonlinear dynamics and system stability to the design and implementation of advanced controllers.

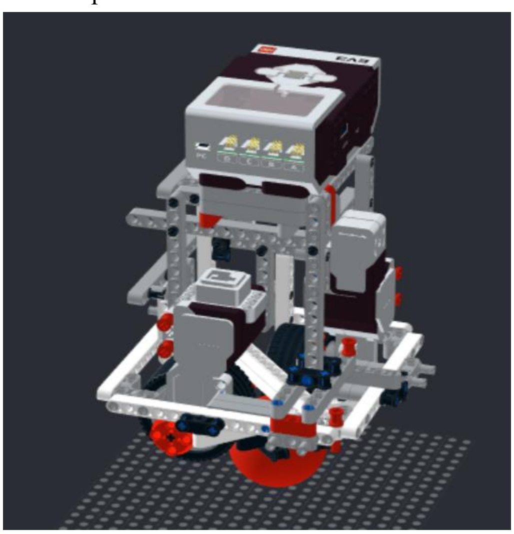


Figure 6. *The Ballbot with Mindstorms EV3*

The non-linear, underactuated Ballbot is modeled as a multi-body system with rigid bodies (the ball and the body) and kinematic constraints. The Ballbot's motion is decoupled into two identical planes, each for one motor and gyro sensor, therefore, θ and ψ are selected as generalized coordinates. Assuming no-slip conditions between the ball and the body, the planar dynamics are derived using the Euler-Lagrange method [1], as shown in Equation (2):

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} &= F_\theta \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\psi}}\right) - \frac{\partial L}{\partial \psi} &= F_\psi \end{aligned} \tag{2}$$

where: θ and ψ rotating angle of the ball and tilting angle of the Ballbot;

F_θ and F_ψ: external forces act on the system;

and

The Lagrangian L is determined by kinetic (T) and potential (V) energies of the system (L=T-V). Thus, the state-space model can be illustrated in Figure 7.

LQR was selected as the controller for the Ballbot due to its computational efficiency and its inherent suitability for multivariable stabilization systems. All parameters used in the LQR method for the Ballbot with Mindstorms EV3 can be described in Eq. (3):

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 6000 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; R = 750 \tag{3}$$

$$K_{LQR} = \begin{bmatrix} -0.0158 & -1.786 & -0.0275 & -0.2882 \end{bmatrix}$$

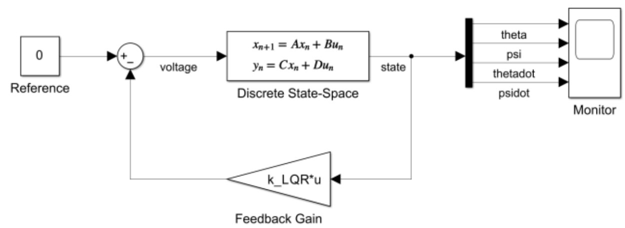


Figure 7. *State-space model for Ballbot simulation*

The effectiveness of the LQR controller used for the Ballbot model with Mindstorms EV3 was initially evaluated through simulation results, as shown in Figure 8. The Ballbot system demonstrates a quick and stable response. After an initial tilting angle ψ of 4^o (x_0=[0 4 0 0]), the robot rapidly regained an upright and balanced state in just 0.7 seconds. Although the movements of the ball exhibited a high overshoot amplitude and a low stabilization speed (≈ 3 seconds), this is normal as the Ballbot system prioritizes the balancing process, necessitating significant displacement of the ball. At t = 3s, an external push disturbance, which lasts for 0.1s, is applied, causing a sudden change in the tilt angle. Despite this, the system maintains its stability even before all states fully return to their equilibrium. This demonstrates the system's fast recovery speed, which is essential for handling the fast dynamics in the Ballbot architecture.

The feedback coefficient K_{LQR} obtained from the simulation system is applied to the experimental Ballbot system for verification experiments, with the block diagram summarized in Figure 9. Here, the controller block processes sensor signals, estimates and reconstructs the corresponding state variables, and then uses the feedback system to obtain the voltage command signal. The control signal is then converted to a PWM duty cycle for the Motor block, which includes a built-in saturation filter (ranging from -100 to 100) to ensure hardware safety. For further reliability, the addition of an anti-windup mechanism is optional in systems with integral terms.

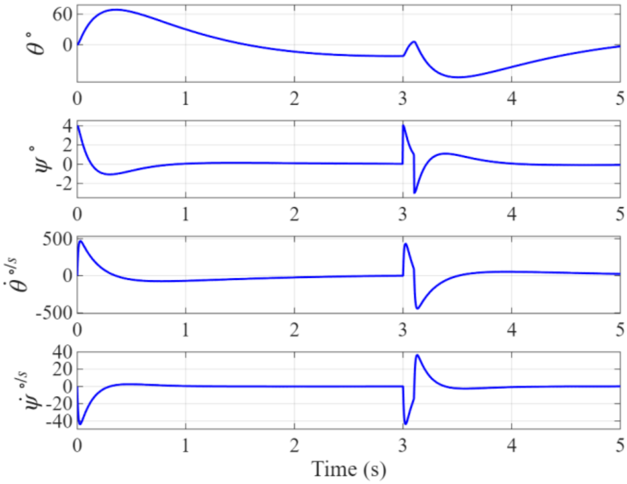


Figure 8. *Simulation results of LQR controller for the Ballbot with Mindstorms EV3*

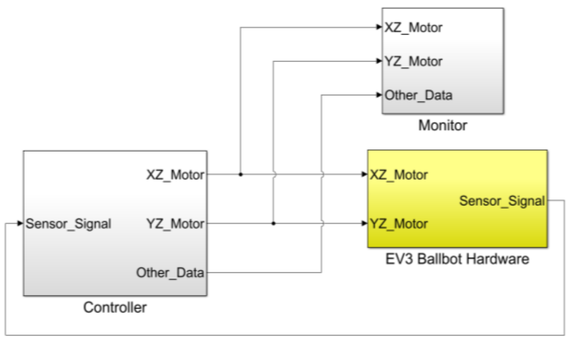


Figure 9. *Block diagram of LQR control for Ballbot HIL integration with Mindstorms EV3*

Before applying to the Ballbot hardware, the system's dynamic parameters, as well as the motor's electromechanical parameters, were thoroughly and precisely identified. Afteward, the Ballbot was constructed based on the mechanical structure shown in Figure 6, featuring optimized mass distribution and Center of Mass (COM). Given the gyroscope's maximum frequency of 300Hz, a 250Hz sampling frequency was selected, integrated with a low-pass filter for accurate tilt angle estimation. Experimental data showing 6,251 samples collected over 25s confirms a precise and efficient computation process with no latency or task overrun.

The experimental results for balancing control of the Ballbot with Mindstorms EV3 are presented in Figure 10. It can be observed that the experimental system exhibits significantly stronger oscillations compared to the simulations. The main reason is that in reality, the Ballbot tends to experience micro-imbalances when at rest, causing the system to continuously generate new unstable states after each oscillation cycle. However, when examining individual oscillations, it is clear that the maximum overshoot amplitude and stabilization speed of the experimental system still correspond with the simulations, whether of the roll or pitch angles. Additionally, the most important parameter reflecting the stability of the system is the tilting angle ψ with respect to the vertical, which oscillates within a certain range [-2°, 2°]. Although there is a moving away from zero over time, the amplitude of the oscillation remains consistent. This moving range arises due to the angle deviation ψ, which is indirectly calculated from the integral of the angular velocity. ψ obtained from the tilting sensor, making the accumulation of error that leads to deviation from the origin unavoidable. Notably, the experimental oscillation amplitudes closely match the simulation results, demonstrating that a valid MBD workflow can achieve significant correlation between the model and the real hardware. Consequently, the experimental signals clearly demonstrate the stability characteristics of the Ballbot body, affirming the high reliability achieved through accurate design of the simulation system.

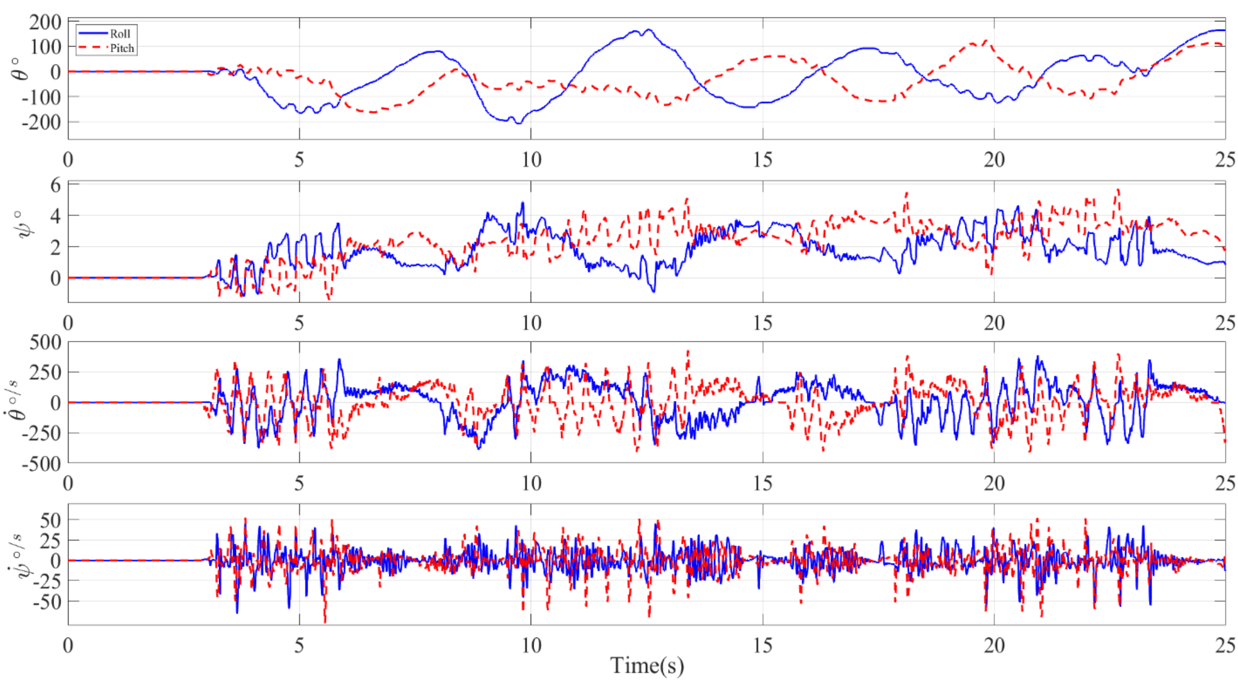


Figure 10. *Experimental results of LQR controller for Ballbot HIL integration with Mindstorms EV3*

Through the evaluation and analysis of differences between the simulation and experimental results, the system can be improved by adding control blocks, such as an integrator to eliminate static offset, a feed-forward algorithm to predict imbalance trends, or a signal filter to stabilize the angle value ψ. After refinement, the model is updated and embedded back into the experimental hardware, resulting in a Ballbot system with more stable and accurate responses. This iterative process clearly illustrates the effectiveness of the Model-Based Design approach in optimizing and shortening the development cycle of real control systems.

5. Conclusion

In summary, this paper demonstrated the successful validation of the MBD workflow on the LEGO EV3, showing a consistent relationship between simulation and hardware performance. This is supported by the observed performance of the PID controller, characterized by relatively fast settling times and low steady-state errors in both position and speed tracking. Furthermore, the Ballbot system demonstrated robust stability via the LQR controller, achieving both upright balancing and position holding with minimal tilt angles and no apparent divergence, as simulated and predicted. These results suggest that the MBD methodology accelerates the development cycle and validates its reliability. This makes the MBD workflow an effective tool for education and research, makes modeling advanced control theory easier and provides a clear, practical way for students and researchers to advance from simple motor control to successfully using complex state-space techniques on accessible hardware, such as the LEGO EV3.

REFERENCES

[1] A. D. Pham, B. H. Thai, P. V. Dang, and N. T. Vo, "Analysis of the parametric configuration impact on BallBot control performance", *International Journal of Mechanical System Dynamics*, vol. 4, no. 4, pp. 446-460, 2024. 10.1002/msd2.12133.

[2] H. N. Le, P. V. Dang, A. D. Pham, and N. T. Vo, "System identifications of a 2DOF pendulum controlled by QUBE-servo and its unwanted oscillation factors", *Archive of Mechanical Engineering,* vol. 67, no. 4, pp. 435-450, 2020.

[3] N. T. Vo, D. D. Tran, H. N. Le, K. A. Nguyen, and A. D. Pham, "Model Reference Adaptive Controller (MRAC) for a Double Line-Sensors Mobile Robot", *2024 International Conference on Connected Innovation and Technology (ICCITX),* Rabat, Morocco, 2024, pp. 1-6. 10.1109/ICCITX61791.2024.11071069.

[4] N. T. Vo, A. D. Pham, and D. M. T. Nguyen, "Hierarchical sliding mode control with oscillation compensation for low-cost inverted pendulum systems using hardware-in-loop", *Archive of Mechanical Engineering*, vol. 72, no. 1, pp. 31-55. 2025, 10.24425/ame.2025.153737.

[5] N. T. Vo, T. H. T. Tran, H. N. Le, D. S. Nguyen, and A. D. Pham, "Design of Fuzzy PID Controller for Delta Robot Talking into Account the Mass of the End Effector", *2024 International Conference on Connected Innovation and Technology (ICCITX)*, Rabat, Morocco, 2024, pp. 1-5. 10.1109/ICCITX61791.2024.11070380.

[6] A. D. Pham and H. J. Ahn, "Evaluation of Input Shaping Methods for the Nonlinear Vibration System Using a Furuta Pendulum", *Journal of the Korean Society for Precision Engineering*, vol. 37, no. 11, pp. 827-833, 2020. 10.7736/JKSPE.020.056.

[7] R. L. A. Shauri, N. M. Salleh, and A. K. A. Hadi, "PID position control of 7-DOF three-fingered robotic hand for grasping task", *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*, Penang, Malaysia, 2014, pp. 70-74. 10.1109/ICCSCE.2014.7072691.

[8] T. Kelemenová *et al*., "Model based design and HIL simulations," *American Journal of Mechanical Engineering*, vol. 1, no. 7, pp. 276-281, 2013. 10.12691/ajme-1-7-25.

[9] L. Belhamel, A. Buscarino, A. Cucuccio, L. Fortuna, and G. Rascona, "Model-Based Design Streamlines for STM32 Motor Control Embedded Software System", *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, Prague, Czech Republic, 2020, pp. 223-228. 10.1109/CoDIT49905.2020.9263910.

[10] J. Hoyos-Gutiérrez, J. Cardona-Aristizabal, P. Muñoz-Gutiérrez and D. Ramirez-Jimenez, "A Systematic Literature Review on Rapid Control Prototyping Applications", *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 18, no. 1, pp. 76-85, Feb. 2023. 10.1109/RITA.2023.3250559.

[11] V. Voth, and O. Bertram, "Aircraft System Design: A Model-Based and Collaborative Approach", in *Proc. 34th Congress of the International Council of the Aeronautical Sciences* (ICAS), Florence, Italy, Sep. 9–13, 2024. ISSN: 2958-4647.

[12] F. Klassner, "A case study of LEGO Mindstorms'™ suitability for artificial intelligence and robotics courses at the college level", *SIGCSE Bulletin*, vol. 34, no. 1, pp. 8-12, Feb. 2002. 10.1145/563517.563345.

[13] P. Udvardy and B. Beszédes, "Educational application of LEGO Mindstorms EV3 system by using different simulation environments", *Proc. 15th International Symposium on Applied Informatics and Related Areas (AIS 2020)*, Székesfehérvár, Hungary, Jan. 2020, pp. 133-136. ISBN 978-963-449-209-2.

[14] T. Chetty, "Lego© Mindstorms: Merely a toy or a powerful pedagogical tool for learning computer programming?", *Proc. 38th Australasian Computer Science Conference (ACSC 2015)*, vol. 27, Jan. 2015, pp. 30.

[15] C. Dong, O. Povorozniuk, A. Topalov, K. Wang, and Z. Chen, "Development of the control system for LEGO Mindstorms EV3 mobile robot based on MATLAB/Simulink elements", *Technology Audit & Production Reserves – Systems & Control Processes*, vol. 1, no. 2, pp. 30-35, Feb. 2023. 10.15587/2706-5448.2023.274846.

[16] R. A. García-García and M. Arias-Montiel, "Linear Controllers for the NXT Ballbot with Parameter Variations Using Linear Matrix Inequalities [Lecture Notes]", *IEEE Control Systems Magazine*, vol. 36, no. 3, pp. 121-136, June 2016. 10.1109/MCS.2016.2535915.

[17] R. A. García-García, and M. Arias-Montiel. "A Robust Control Scheme Against Some Parametric Uncertainties for the NXT Ballbot", *Multibody Mechatronic Systems, Mechanisms and Machine Science*, vol. 25, Cham: Springer, 2015, pp. 249-260. 10.1007/978-3-319-09858-6_24.

[18] R. Enemegio, F. Jurado, and J. Villanueva-Tavira, "Experimental evaluation of a Takagi–Sugeno fuzzy controller for an EV3 Ballbot system", *Applied Sciences*, vol. 14, no. 10, article 4103, May 2024. 10.3390/app14104103.