Design and Development of a Low-Cost Robotic Platform for STEM Education of an Automatic Control System: Rotary-Type Double Inverted Pendulum Case

Wiput TUVAYANOND[a] & Piyanun RUANGURAI [b*]

[a]*Industry Agricultural Engineering, Faculty of Engineering, Rajamangala University of Technology Thanyaburi, Thailand*

[b]*Department of Mechanical Engineering Technology, College of Industrial Engineering Technology, King Mongkut's University of Technology North Bangkok, Thailand*

*piyanun.r@cit.kmutnb.ac.th

Abstract: In Thailand STEM education, teaching of robotics or automatic control in undergraduate is limited to the theoretical textbooks or computer simulation due to inadequate control plants for classroom demonstration and laboratory experimentation. This crucially prevents students to understand the gaps or limitations between the theory and the real-world problem. Thus, this research is aimed to design and develop the robotic platform, which is the rotary-type double inverted pendulum, by considering low fabrication cost, easy part maintenance and safety. Also, the platform permits the students to design the Linear Quadratic Regulator (LQR) controller and implement it for balancing the robot.

Keywords: STEM, teaching, education, robot, control

1. Introduction

In present, STEM education of robotic control is key learning for undergraduate engineering students. Ideally, students are effectively trained in systematic thinking and problem solving through building the robot (Chen & Chang, 2018). However, there exists several difficulties for those students who are highly expected from the industrial side to have the ability to apply the theory to practical applications. The important one of problems is the lack or insufficiency of controlled plants for experimentation and hand-on practices due to highly expensive cost, as a result, most, teaching courses are only limited to the studying of conventional textbooks or conducting experiments through software simulation. These can cause students to mislead some control issues in real-world situations.

The well-known robotic platform, so called Rotary-type Double Inverted Pendulum(RDIP), has been broadly illustrated for teaching in control theory courses (Gustafsson, 2016; YeeChin, & Chiam, n.d.). The study objective of the robot is to instruct students to design the controller for automatically balancing the two vertical linkages in up-right position. In addition, there has been the commercial robot offered by Quanser company (Quanser consulting inc., 2021) with the complete set of laboratory instruction, electronic control devices, mechanical robot structure and control software. Although this perfectly facilitates both instructors and students' study, the cost is extremely high. Moreover, part maintenance is not simple because of non-domestic products. To solve the problem, this research therefore aims to design and develop the RDIP and control software for teaching students in control theory or robotic courses by focusing on low fabrication cost, easy maintenance, and safety.

2. Literature Review

In pioneer works of RDIP, the robots were in-house constructed for either balancing or swing-up both pendulums (Komine, Iwase, Suzuki, & Furuta, 2004; Yamakita, Iwashiro, Sugahara, & Furuta, 1995).

Yamakita et. al (1995) presented the state transfer controllers that are switched to convey the state of the RDIP based on pre-defined control areas. Komine et al. (2004) presented the multi-step swing-up controller. Firstly, the Linear Quadratic Regulator (LQR) was designed to stabilize the second link of the robot in up-right position, then the linear state feedback controller was switched to swing the third link of the robot from the hanging position to the upright position.

Later, the RDIPs have been received significant attention in control due to its dynamic nonlinearity, underactuated and unstable system that challenges various controller designs and dynamic studies. There existed the commercial DRIP product used in some works (Casanova, Salt, Piza, & Cuenca, 2012; Patil & Kurode, 2018). Casanova et. al (2012) designed the balancing controller that compensates for delay effect resulted from the distant sensor and actuator in RDIP and does experiment with the commercial RDIP. Also, Patil and Kurode (2018) experimented with the robustness of the high order sliding mode controller to regulate the RDIP at the up-right position. This commercial RDIP supports not only the robot hardware but also the software packages that users can design the controllers and collect data based on MATLAB/Simulink or LabVIEW, effectively.

3. Methodology

3.1 Robot design

The Solidworks CAD software is used to create the linkages of RDIP, mechanical transmission (pulleys, timing belts and shafts) and DC motor. In the 1st axis, the DC motor indirectly drives the robot through timing belt transmission with a tensioner. This allows the electrical cables of the 2nd and 3rd encoders to be rotated around the 1st axis freely by passing them through the slip ring and hollow shaft of the 1st axis as shown in Figure 1. To increase the link stiffness under bending load, the 5083-grade aluminum is specially selected to fabricate the 1st link with 5-millimeter thickness while the other two linkages (2nd and 3rd axis), 1060-grade aluminum with 3-millimeter thickness is instead used to minimize the load driven by the motor. All linkage length is adjustable and summarized in Table 1.

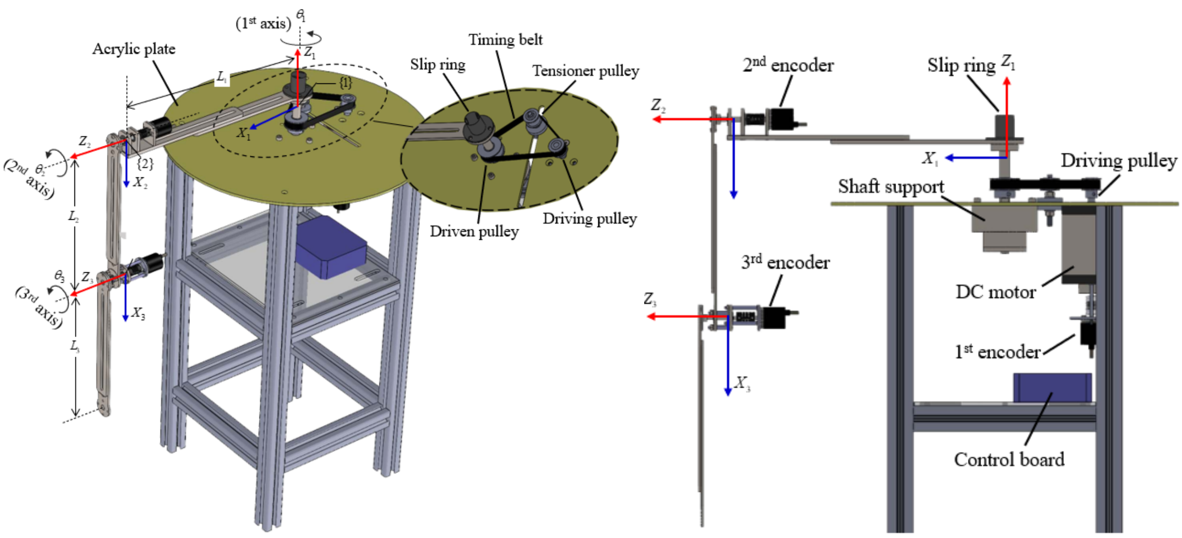

Figure 1. RDIP structure design

Table 1. *Link Maximum and Minimum Length of RDIP*

	Symbol	Max. length(mm.)	Min. length(mm.)
Link 1	L_1	375.0	275.0
Link 2	L_2	337.0	230.0
Link 3	L_3	337.0	230.0

3.2 *Electronic Control Design*

To control and interface of the RDIP, the low-cost but high-performance Neucleo STM32F411RET6 development board is chosen as core computation and communicates user via the full-speed USB to serial communication. The interfacing board plugged over the Nucleo board is custom designed to collect the 1st, 2nd and 3rd position data of RDIP from three incremental quadrature encoders and send the command to DC motor driver; Junus JPS-090-20 model, which controls the motor current at 20 kHz sampling rate. For user safe operation, the emergency and footswitches are integrated to switch on/off the electrical power and start or stop the robot at a distant location, respectively. The electrical control schematic of the RDIP shows in Figure2. Note that all components and its specifications are summarized in Table 2.

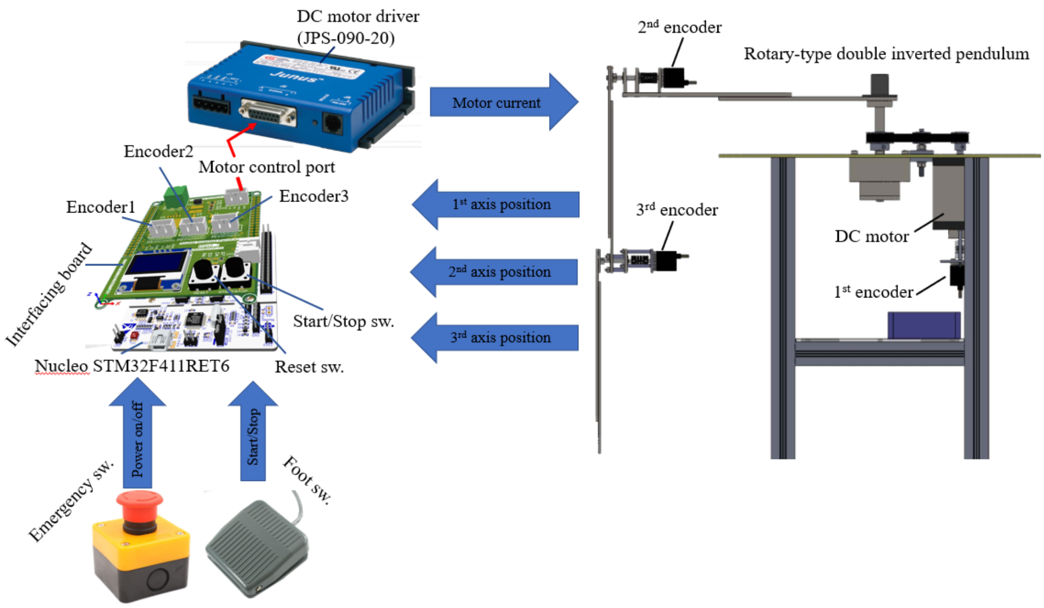

Figure 2. Electrical control schematic of the RDIP

Table 2. *Electrical Components of RDIP Control*

	Component	Specification
Control board	Nucleo STM32F411RET6	32-bit ARM-Cortex M4 RISC 128 KB Ram 512K flash Microcontroller
Motor driver	Junus JSP-090-20	Velocity/Torque control mode, 10A Continuous current/20A peek current, 20-90V input
DC motor	MY6812	150W/24V DC motor
Quadrature encoder	E6A2-CWZ3C	Incremental quadrature encoders 1000 pulse/revolution, 5-12V input

3.3 *Control Software*

3.3.1 *Controller Design*

To balance the RDIP at the up-right position, the Linear Quadratic Regulator(LQR) algorithm, which is linear optimal control referenced in many control textbooks (Nise, 2020; Ogata, 2015) is implemented in an embedded board. The objective of the controller is aimed to compromise the effects of state variables and control input in the linear state feedback problem shown in Figure 3. By designing the weighted diagonal matrix $\mathbf{Q} = diag(q_1, q_2, \ldots, q_6) \in \sim^{6\times 6}$ for state variables and scalar value $R \in \sim$ for control input in Equation 1, the optimal feedback gain matrix $\mathbf{K} = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 & k_5 & k_6 \end{bmatrix} \in \sim^{1\times 6}$ that minimizes the cost function J subjected to the closed-loop system constraint in Equation 2 is efficiently obtained by the *lqr* MATLAB command (Mathworks, 2021).

$$J = \int_0^{\infty} \left(\mathbf{x}^{\mathrm{T}} \mathbf{Q} \mathbf{x} + u^{\mathrm{T}} R u \right) dt \tag{1}$$

Where, $\mathbf{x}^{\mathrm{T}} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 \end{bmatrix}^{\mathrm{T}} \in \sim^{6\times 1}$ and $u \in \sim$ are state variable vector and control input, respectively. The matrix $\mathbf{Q} = diag(q_1, q_2, \ldots, q_6) \in \sim^{6\times 6}$ and scalar value $R \in \sim$ are weighted diagonal matric for state variables and weighted coefficient of control input, respectively.

$$\dot{\mathbf{x}} = \left(\mathbf{A} - \mathbf{BK} \right) \mathbf{x} \tag{2}$$

Note that the state matrix $\mathbf{A}$ and input matrix $\mathbf{B}$ of the RDIP can be obtained by linearizing the nonlinear dynamics of the RDIP at the up-right such derived in (Tuvayanond & Ruangurai, 2021).

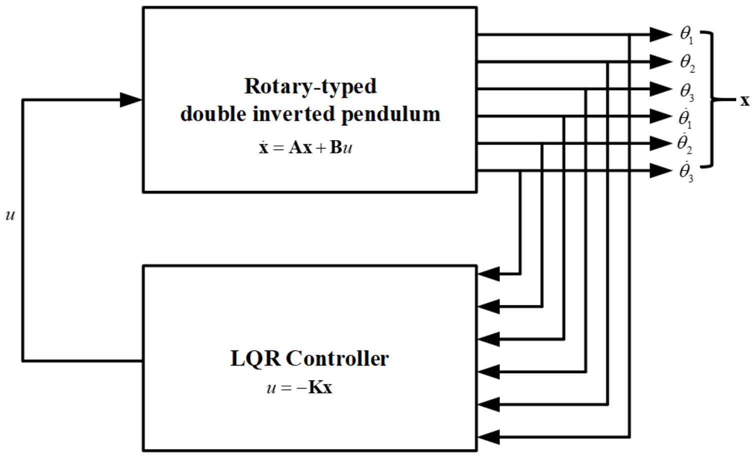

Figure 3. Linear state feedback control problem.

The LQR algorithm is implemented in the embedded control board and executed at 50 Hz while it simultaneously communicates to the user via USB serial interface.

3.3.2 User Interface Software

The commands of the RDIP are designed as the ASCII character sequences with the delimiter with linefeed character (0x0A code). All commands are summarized in Table 3. Users are required to install the USB serial communication software at the host computer to send the commands to the board. The following steps are executed to configure the LQR gains, start and stop balancing the robot.

- *Step 1:* Plug the USB cable into the control board and open the serial COM port communication software at the host computer,
- *Step 2:* Configure the serial protocol with 115200 bits/s bit rate, 8-bit data and 1 stop bit without parity check at the host computer and then connect to the board.
- *Step 3:* Put the derived LQR gains to the board by sending the serial command *'lqr'* listed in Table 3. For example, if the user would like to set $k_1 = 4.1\times 10^{-3}, k_2 = 0.5, k_3 = 0.7,$ $k_4 = 7.2\times 10^{-5}$, $k_5 = 2.0\times 10^{-7}$ and $k_6 = 8.2\times 10^{-6}$, then the following command, 'k 4.1E-3 0.5 0.7 7.2E-5 2E-7 3.1E-4 8.2E-6', is executed.
- *Step 3a:* (Optional) If the user would like to collect the position and control input data to the file,

then the *'log'* serial command is executed.

- *Step 4*: To set the zero referenced position of the robot, manually hold the pendulums in the up-right position as in Figure 4. Then, step on the footswitch or send the command 's' to start balancing while releasing your hand. Note that the zero referenced position is not necessary for the exact up-right position since the controller can stabilize some position error. If unbalancing situation occurs, then stop balancing with *Step 5* and repeat *Step 4* to initialize the up-right position again.
- *Step 5*: To stop balancing, then step on the footswitch or send the command 's'.

Figure 4. Steps for robot initialization and balancing.

Table 3. *Serial Command Interface*

Command description	Ascii sequences	Example
Start/Stop balancing	's'	's'
Update the LQR gains	'k *k1-value* …'	'k 4.1E-3 0.5 0.7 7.2E-5 2E-7 3.1E-4 8.2E-6'
Log/Stop logging data to the file	'log'	'log'

4. Conclusions

The mechanical structure of the RDIP is designed with the consideration of low cost, simple maintenance, and safety. The LQR controller is implemented to balance the RDIP in the up-right position while users or students can learn to theoretically design the controller, implement it in practice and record the experimental data for analysis as well. This can essentially help the students, who are in STEM education, to realize the relationship, limitations, and utilization from theory to practical applications. In future work, this robot programming will be integrated to the commercial MATLAB/Simulink software that are broadly used in control theory and robotic courses.

Acknowledgements

We would like to thank the King Mongkut's University of Technology North Bangkok, who has funded this research with contract no. KMUTNB-64-DRIVE-38.

References

Casanova, V., Salt, J., Piza, R., & Cuenca, A. (2012). Controlling the double rotary inverted pendulum with multiple feedback delays. *International Journal of Computers, Communications and Control*, *7*(1), 20–38. https://doi.org/10.15837/ijccc.2012.1.1420

Chen, Y., & Chang, C. C. (2018). The impact of an integrated robotics STEM course with a sailboat topic on high school students' perceptions of integrative STEM, interest, and career orientation. *Eurasia Journal of Mathematics, Science and Technology Education*, *14*(12). https://doi.org/10.29333/ejmste/94314

Gustafsson, F. (2016). *Control of Inverted Double Pendulum using Reinforcement Learning*. Retrieved from http://cs229.stanford.edu/proj2016/report/Gustafsson-ControlOfInvertedDoublePendulumUsingReinforcementLearning-report.pdf

Komine, T., Iwase, M., Suzuki, S., & Furuta, K. (2004). Rotational control of double pendulum. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *37*(14), 325–330. https://doi.org/10.1016/S1474-6670(17)31124-2

Mathworks. (2021). Linear-Quadratic Regulator (LQR) design. Retrieved from https://www.mathworks.com/help/control/ref/lqr.html

Nise, N. S. (2020). *Control Systems Engineering* (8th ed.). Wiley.

Ogata, K. (2015). *Modern Control Engineering* (5th ed.). Pearson.

Patil, M., & Kurode, S. (2018). Stabilization of rotary double inverted pendulum using higher order sliding modes. *2017 Asian Control Conference, ASCC 2017*, *2018-Janua*, 1818–1823. https://doi.org/10.1109/ASCC.2017.8287450

Quanser consulting inc. (2021). ROTARY DOUBLE INVERTED PENDULUM. Retrieved from https://www.quanser.com/products/rotary-double-inverted-pendulum/

Tuvayanond, W., & Ruangurai, P. (2021). Design, modeling and parameters identifiation of rotary-type double inverted pendulum. In *International Congress on Advanced Applied Informatics(IIAI-AAI)* (pp. 476–481). IEEE CPS. https://doi.org/10.1109/IIAI-AAI53430.2021.00082

Yamakita, M., Iwashiro, M., Sugahara, Y., & Furuta, K. (1995). Robust swing up control of double pendulum. *Proceedings of the American Control Conference*, *1*, 290–295. https://doi.org/10.1109/acc.1995.529255

YeeChin, & Chiam. (n.d.). Double Inverted Pendulum Stabilizer. Retrieved from https://engineering.purdue.edu/~zak/Publish/home.html