

Application of LQG Control for Pendubot System

Duc-Anh-Quan Nguyen¹, Luu-Quang-Thinh Nguyen^{2,*}, Hoang-Anh Nguyen³, Phu-Hung Nguyen⁴, Quoc-Thai Tran⁵, Cao-Sang Pham⁶, Hoang-Bao Tran⁷, Van-Huy Pham⁸, Anh-Quoc Nguyen⁹, Phong-Luu Nguyen¹⁰
1,2,3,4,5,6,7,8,9,10 Ho Chi Minh City University of Technology and Education (HCMUTE), Thu Duc city, HCMC, Vietnam
Email: ¹ 20151408@student.hcmute.edu.vn, ² 20151063@student.hcmute.edu.vn, ³ 20151254@student.hcmute.edu.vn, ⁴ 20145514@student.hcmute.edu.vn, ⁵ 20145608@student.hcmute.edu.vn, ⁶ 20145596@student.hcmute.edu.vn, ⁷ 20151339@student.hcmute.edu.vn, ⁸ 20145452@student.hcmute.edu.vn, ⁹ 20145431@student.hcmute.edu.vn, ¹⁰ luunp@hcmute.edu.vn
*Corresponding Author

Abstract—The paper focuses on the study of the Pendubot system, a classic system widely used in control research, particularly the 2-link Pendubot. This system is nonlinear with a single input and multiple outputs (SIMO) and is under-actuated. The paper addresses the challenges of applying optimal Linear Quadratic Regulator (LQR) control to Pendubot, requiring an accurate model without noise. The author proposes using a Kalman filter to estimate the system's state variables and mitigate the impact of noise. Subsequently, the LQR controller is applied based on the estimated state variables. Thus, the study suggests an integrated approach combining Kalman filtering and LQR control to enhance Pendubot's performance under real-world conditions. The combination of LQR and Kalman forms a new controller called LQG, which addresses issues encountered by standalone LQR and improves the system's optimal performance. The paper focuses on evaluating the performance of LQR and LQG controllers through system simulations with the input being the motor force applied to the first bar of the Pendubot system in MATLAB Simulink software.

Keywords—Pendubot, LQG Control, Kalman Filter, MATLAB/Simulink

I. INTRODUCTION

Pendubot is an underactuated system with two degrees of freedom, as illustrated in Fig. 1, consisting of one actuated joint and two non-actuated joints [1]-[4]. In various experiments, researchers have analyzed the second-order mathematical equations of the Pendubot and employed control strategies for stability and trajectory tracking. Additionally, practical applications of Pendubot have been explored in control research and educational environments.

In these studies, the authors have modeled the system starting from a stable equilibrium position with a vertically downward configuration of the two joints to an unstable balanced position with the two joints in an upright configuration using a Swing-up control scheme [4]-[11]. Once the Pendubot is brought close to the balanced state, a balancing control law is activated to maintain the Pendubot in an upright position. Notably, solutions based on energy methods have been proposed. For instance, I. Fantoni and colleagues designed a control law by computing the potential energy of the pendulum at the upright position and establishing a Lyapunov-based control law [10]. In the study by Daniel J. Block [12], a linearization technique combined with a PD controller was employed to Swing-up the Pendubot to the unstable balanced position (Top or MID) by

adjusting the parameters of K_D and K_P . An optimal control approach was also applied to achieve stabilization.

Over the past decades, several balancing control laws for Pendubot have been proposed to evaluate the system's stability [6]-[13]. These include nonlinear control laws, linear control laws, and intelligent control laws. However, the presence of noise in the system is a consideration in the development of control strategies in the field of control engineering. In a published paper in 2014, Nguyen Dinh implemented a nonlinear control law based on the Sliding Mode control technique for the underactuated system to achieve robust and stable control in a noisy environment [13]. In [14], a nonlinear disturbance observer was utilized to estimate and compensate for disturbances, enabling stable and balanced control for the Pendubot system. In [15], Adaptive sliding-mode neural network control is based on passive properties.

The Pendubot system is a complex nonlinear system, and computing its exact model can often be more challenging compared to a linear model. Hence, the authors propose the use of a Linear Quadratic Regulator (LQR) control law. However, due to the presence of noise in the system's operation, a Kalman filter is designed to estimate the system's noise. The combination of an LQR controller and a Kalman filter will result in an LQG controller. The control system's performance will be enhanced, and the LQG controller will mitigate the effects of system noise and measurement noise, which are the two primary causes of instability in the control system design. To design the LQR controller, linearization of the system around the operating point is required [4], [6]. Estimating the system's noise enables easier control of the system and reduces the energy supplied by the motor, thereby preventing potential damage.

II. MATERIAL AND METHOD

A. Modeling of Pendubot System

The Pendubot is a two-degree-of-freedom (two-DOF) underactuated system with two controlled joints, as depicted in Fig. 1. Joint one is directly linked and controlled (q_1), while joint two is coupled to joint one through a connecting joint and equipped with an encoder to measure the angle (q_2). The mass of joint one (two) is denoted as the length of link one (two) is defined as ($l_1(l_2)$), and the distance from the encoder axis to the center of mass of link one (two) is denoted as $l_{c1}(l_{c2})$. The inertia moment is located at the

center of the mass of link one (two) and is defined as $I_1(I_2)$. We apply the Euler-Lagrange equation as follows:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau \quad (1)$$

Where $q = [q_1, q_2]^T \in \mathcal{R}^2$ is the position vector and $\dot{q} = [\dot{q}_1, \dot{q}_2]^T \in \mathcal{R}^2$ is the velocity vector of the system, $\tau = [\tau_1, 0]^T$ is the input of the control variable. The obtained dynamic equation of the system is expressed as follows:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (2)$$

Where:

$$M(q) = \begin{bmatrix} \beta_1 + \beta_2 + 2\beta_3 \cos q_2 & \beta_2 + \beta_3 \cos q_2 \\ \beta_2 + \beta_3 \cos q_2 & \beta_2 \end{bmatrix}$$

$$C(q, \dot{q}) = \begin{bmatrix} -\beta_3 \sin q_2 \dot{q}_2 & -\beta_3 \sin q_2 \dot{q}_2 - \beta_3 \sin \beta \dot{q}_2 \\ \beta_3 \sin \beta \dot{q}_2 & 0 \end{bmatrix}$$

$$G(q) = \begin{bmatrix} \beta_4 g \cos q_2 + \beta_5 g \cos(q_1 + q_2) \\ \beta_5 g \cos(q_1 + q_2) \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau \\ 0 \end{bmatrix}$$

$$\beta_1 = m_1 l_{c_1}^2 + m_2 l_1^2 + I_1; \beta_2 = m_2 l_{c_2}^2 + I_2;$$

$$\beta_3 = m_2 l_1 l_{c_2}; \beta_4 = m_1 l_{c_1} + m_2 l_1; \beta_5 = m_2 l_{c_2}.$$

In addition, $M(q)$ is referred to as the inertia matrix, $C(q, \dot{q})$ is referred to as the Coriolis matrix, and $G(q)$ is referred to as the gravity matrix. The parameters used in this paper are defined as presented in Table 1.

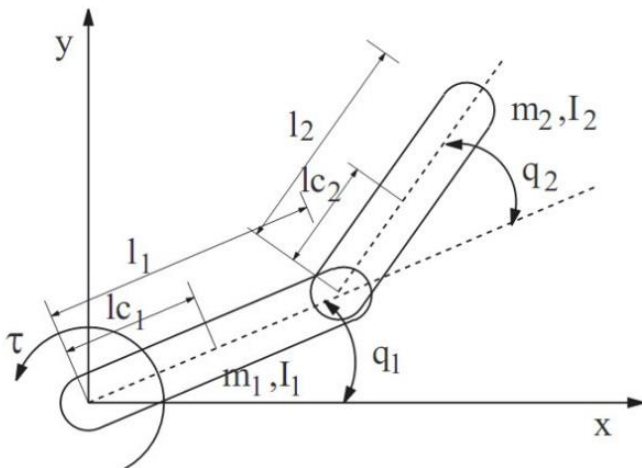


Fig. 1. Dynamics analysis of the pendubot system

B. Analysis of System Controllability

Before designing control and filtering systems for any given linear system, it is necessary to assess the controllability and observability of the system. Starting from the nonlinear equations, linearization around the operating point of the system is performed. In this case, the control is chosen at the TOP position ($q_1 = \frac{\pi}{2}; q_2 = 0$) as in Fig. 2.

1. Analysis of System Controllability

Linear System:

$$\dot{x} = Ax + Bu \quad (3)$$

When the system is stable, the state variables of the system will converge to zero:

$$x = [x_1 \ x_2 \ x_3 \ x_4]^T = \left[q_1 - \frac{\pi}{2} \quad \dot{q}_1 \quad q_2 \quad \dot{q}_2 \right]^T$$

The matrices A and B are obtained through linearization of the system around the operating point, resulting in:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 7.8312 & 0 & -0.2631 & 0 \\ 0 & 0 & 0 & 1 \\ -5.8583 & 0 & 2.8783 & 0 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 20.1247 \\ 0 \\ -21.7213 \end{bmatrix}$$

The controllability matrix is given by:

$$\phi = [B \ AB \ A^2B \ A^3B] \quad (4)$$

Using the matrices A and B, we can assess the controllability of the system. If the rank of the controllability matrix, denoted as (4), is equal to the number of state variables, then we conclude that the system is controllable. Based on this, we can design a control system for the linearized system:

$$\phi = \begin{bmatrix} 0 & 20.1247 & 0 & 163.3154 \\ 20.1247 & 0 & 163.3154 & 0 \\ 0 & -21.7213 & 0 & -180.4169 \\ -21.7213 & 0 & -180.4169 & 0 \end{bmatrix} \quad (5)$$

Therefore, based on the controllability matrix (5), we can conclude that the system is controllable.

Table 1. Parameters of the system

Parameter	Description
m_1	Mass of link 1 (kg)
m_2	Mass of link 2 (kg)
l_1	Length of link 1 (m)
l_2	Length of link 2 (m)
l_{c_1}	Distance from the encoder axis to the center of mass of link 1 (m)
l_{c_2}	Distance from the encoder axis to the center of mass of link 2 (m)
I_1	Inertia moment of link 1 ($kg.m^2$)
I_2	Inertia moment of link 2 ($kg.m^2$)
q_1	Angle of link 1 with respect to the horizontal axis (rad)
q_2	Angle of link 2 with respect to link 1 (rad)

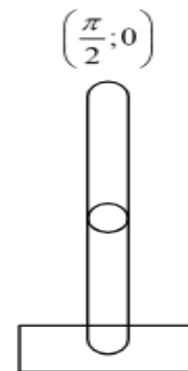


Fig. 2. Top position of the pendubot

2. Analysis of System Observability

The essence of the Kalman filter is an observer, and to construct this filter, it is necessary to assess the observability of the system. The observability matrix is presented:

$$\phi = \begin{bmatrix} 0 & 20.1247 & 0 & 163.3154 \\ 20.1247 & 0 & 163.3154 & 0 \\ 0 & -21.7213 & 0 & -180.4169 \\ -21.7213 & 0 & -180.4169 & 0 \end{bmatrix} \quad (6)$$

The output-state matrix C of the system is as follows:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Therefore, we obtain the result:

$$O = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 7.8312 & 0 & -0.2631 & 0 \\ -5.8583 & 0 & 2.8783 & 0 \\ 0 & 7.8312 & 0 & -0.2631 \\ 0 & -5.8583 & 0 & 2.8783 \end{bmatrix}$$

Hence, based on the rank of the observability matrix being equal to the number of state variables, we can conclude that the system is observable.

C. Design of the Control System

Fig. 3 illustrates the structure of the controller, and the LQG optimal control problem can be separately solved by addressing the pre-defined optimal control problem and the optimal state estimation problem. In this case, the state variables used for computation are denoted as \hat{x} , with an additional component added as the estimation gain L . The control signal $u = -K\hat{x}$ is computed based on these state variables, resulting in a control signal with reduced amplitude oscillations, thereby minimizing adverse effects on the system.

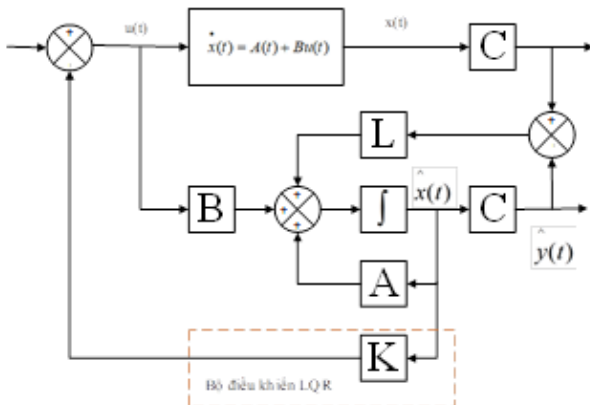


Fig. 3. LQR controller in the system

The quality criteria can be expressed through the following fitness function:

$$J = \frac{1}{2} \int_0^{\infty} (x^T(t)Qx(t) + u^T(t)Ru(t)) \quad (7)$$

The optimal control signal is designed as follows:

$$u(t) = -Kx(t) \quad (8)$$

Where K is the state feedback gain matrix determined by the formula:

$$K = R^{-1}B^T P \quad (9)$$

Where P is the positive semi-definite solution of the algebraic Riccati equation:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (10)$$

The authors simulated the discrete-time system using MATLAB software. To obtain the control gain, they utilized the `dlqr` (A , B , Q , R) function, which allowed them to

calculate the coefficient K for the control design. Where Q and R are positive definite weighting matrices.

D. Design of the Kalman Filter for the System

Consider a discrete-time linear system:

$$\begin{cases} x(k+1) = A_d x(k) + B_d u(k) + w(k) \\ y(k) = C_d x(k) + v(k) \end{cases} \quad (11)$$

Where: $W(t)$ represents the system noise and $V(t)$ represents the measurement noise.

Assuming that the system noise and measurement noise are non-Gaussian, uncorrelated, with zero mean and variances of:

$$E[ww^T] = Q_N \quad (12)$$

$$E[vv^T] = R_N \quad (13)$$

The discrete Kalman filter has the following form:

$$\begin{cases} \hat{x}(k+1) = [A_d \hat{x}(k) + B_d u(k) + L_k [y(k+1) - \hat{y}(k+1)]] \\ \hat{y}(k) = C_d \hat{x}(k) \end{cases} \quad (14)$$

Where L represents the Kalman filter gain, which is presented as:

$$L(k) = A_d \Pi(k) C_d^T (C_d \Pi(k) C_d^T + R_N)^{-1} \quad (15)$$

With Π being the solution of the Riccati equation:

$$\Pi(k+1) = A_d \Pi(k) A_d^T + Q_N - A_d \Pi(k) C_d^T R_N^{-1} C_d \Pi(k) A_d^T \quad (16)$$

III. RESULTS AND DISCUSSION

To verify the operation of the controller presented in this paper, we will conduct a simulation using MATLAB/SIMULINK software. This simulation aims to examine the responses of the signals that need to be controlled by the LQG controller. Furthermore, we will compare the performance with the optimal LQR controller when the system is subjected to undesirable components such as measurement noise and system disturbances. In this simulation, we will control the Pendubot to balance in an upright position, facing upwards, with the state vectors $q = [-\frac{\pi}{2} \ 0 \ 0 \ 0]^T$. At this position, the controllability of the system has been verified using the formula (4). We will proceed with the simulation for a duration of 4 seconds, with a system sampling time of 0.01 seconds. During the simulation, we will compare the state variables of the system's output without a filter and the system's output with a filter to determine their advantages. The main parameters of the constructed system are presented in Table 2. Controller gain matrix K :

$$K = [-59.8531 \quad -38.7811 \quad -59.8290 \quad -37.1507]$$

Observation matrix L parameters:

$$L = \begin{bmatrix} 5.5353 & -0.2079 \\ 15.536 & -0.7785 \\ -2.0792 & 3.3808 \\ -10.7535 & 5.9312 \end{bmatrix}$$

The system noise has very small parameters and is considered to be negligible:

$$Q_n = \begin{bmatrix} 1e-14 & 0 & 0 & 0 \\ 0 & 1e-14 & 0 & 0 \\ 0 & 0 & 1e-14 & 0 \\ 0 & 0 & 0 & 1e-14 \end{bmatrix}$$

The simulation control system is divided into several blocks as shown in Fig. 4:

Table 2. Parameters of the pendubot system

Parameter	Unit
m_1	0.1 (kg)
m_2	0.08 (kg)
l_1	0.3 (m)
l_2	0.4 (m)
l_{c1}	0.17 (m)
l_{c2}	0.21 (m)
I_1	0.04 ($kg.m^2$)
I_2	0.06 ($kg.m^2$)
m_1	0.1 (kg)
m_2	0.08 (kg)

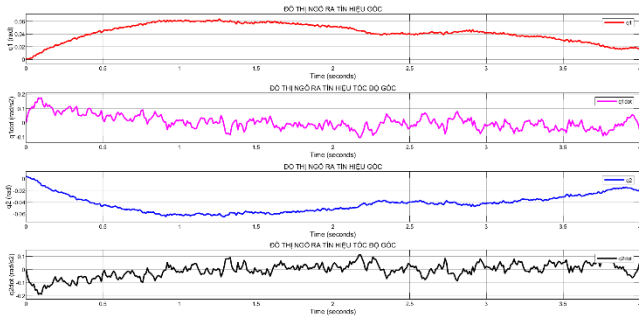


Fig. 4. The response of the output state variables of the model without the kalman filter for estimation

Measurement noise is assumed: $R_n = [0.001 \ 0; \ 0 \ 0.001]^T$

1. System Modeling Block: The mathematical model of the Pendubot system presented in (2) is implemented in MATLAB/SIMULINK.
2. Noise Generation Block: In this block, the authors utilize available blocks in MATLAB/SIMULINK to add measurement noise to the output of the model.
3. Kalman Filter Block: As described in (14), this block is responsible for estimating noise and unwanted components when using sensors for measurements.
4. Control Block: The authors employ the LQR controller and Kalman filter to bring the Pendubot to an upright balanced state, facing upwards.

The assessment of the quality of the control and filtering on the Pendubot model presented:

In Fig. 5, the state variables' outputs are presented without the Kalman filter, while Fig. 6 illustrates the improved quality of the state variables' outputs with the application of the Kalman filter. It can be observed that the output quality of the Kalman filter is no longer affected by oscillations as seen in the absence of the filter, and the control quality is smoother. In Fig. 5, the angular velocity outputs of variables and exhibit strong oscillations, which negatively impact the system. However, with the application of the filter, the signals become flatter, and the oscillations disappear.

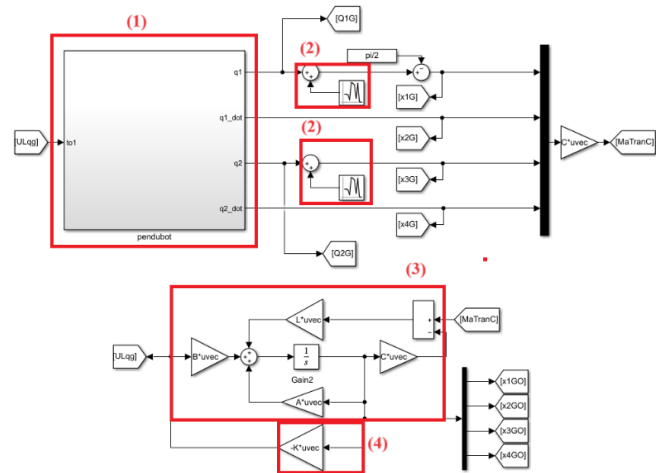


Fig. 5. The structure of the LQG control system

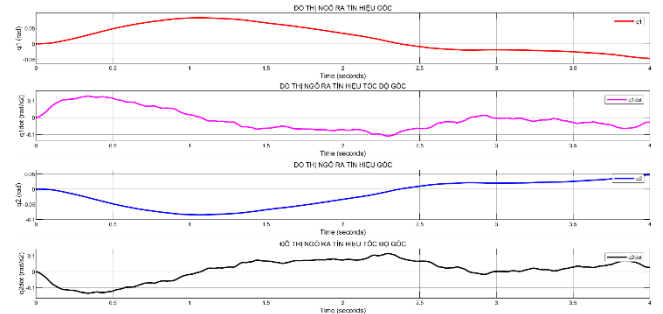


Fig. 6. The response of the output state variables of the model with the kalman filter for estimation

The angular outputs, affected by noise, are estimated remarkably well by the filter in Fig. 7 and Fig. 8. It can be observed that the continuous oscillations captured by the sensors are significantly reduced through the filter, resulting in flat and smooth signals. This indicates excellent performance of the filter.

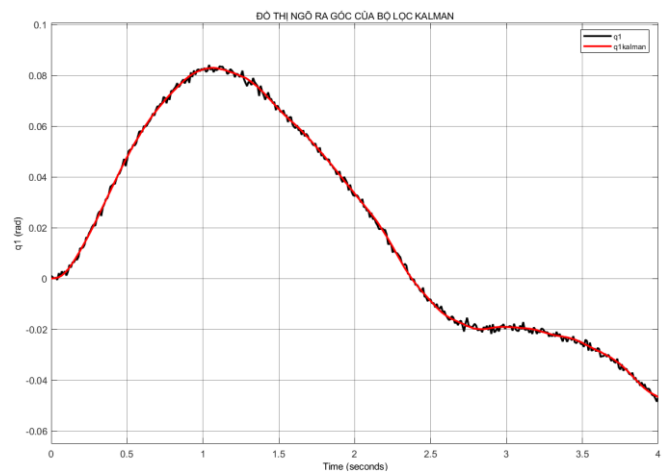


Fig. 7. The output of the estimated angle of link one, obtained through the kalman filter

The control signal supplied to the system from the motor is also crucial. In Fig. 9, when influenced by motor noise, the control signal exhibits significant oscillations. This can have a substantial impact and potentially damage the motor. However, with the implementation of the filter, the control

signal operates with reduced oscillations, minimizing potential harm to the control system.

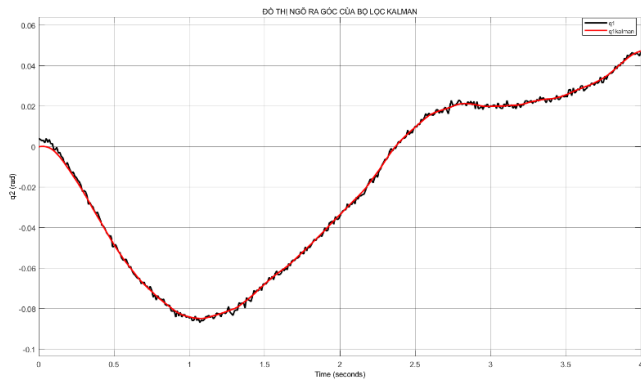


Fig. 8. The output of the estimated angle of link one, obtained through the kalman filter

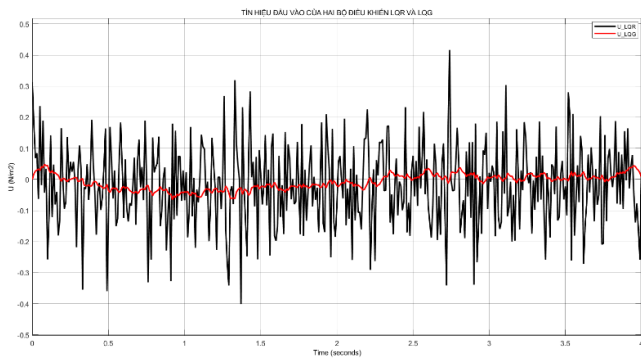


Fig. 9. The response of the control signal if significantly reduced when using the filter

IV. CONCLUSION

In this paper, the authors focus on developing a Kalman filter and combining it with an LQR controller to create an LQG controller for the Pendubot model. The goal is to minimize the influence of noise on the system and improve optimal control quality. Simulation results have shown that using the LQG controller significantly improves the system's response, enhances the quality of control signals, and increases the stability of the Pendubot. This research opens opportunities for further development in constructing parameter estimation sets to address unwanted influences on the system. The next development direction is to address noise-induced issues with systems of higher nonlinearity, such as arc robot systems.

REFERENCES

- [1] T. V. Toan, T. T. Ha and T. V. Do, "Hybrid control for swing up and balancing pendubot system: An experimental result," *2017 International Conference on System Science and Engineering (ICSSE)*, pp. 450-453, 2017, <https://doi.org/10.1109/ICSSE.2017.8030915>.
- [2] I. Fantoni, R. Lozano and M. W. Spong, "Energy based control of the Pendubot," *IEEE Transactions on Automatic Control*, vol. 45, no. 4, pp. 725-729, 2000, <https://doi.org/10.1109/9.847110>.
- [3] T. Albahkali, R. Mukherjee and T. Das, "Swing-Up Control of the Pendubot: An Impulse-Momentum Approach," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 975-982, 2009, <https://doi.org/10.1109/TRO.2009.2022427>.
- [4] X. Xin, M. Kaneda, T. Oki, "The swing-up control for the Pendubot based on energy control approach," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 461-466, 2002, <https://doi.org/10.3182/20020721-6-ES-1901.00889>.
- [5] O. Begovich, E. N. Sanchez and M. Maldonado, "Takagi-Sugeno fuzzy scheme for real-time trajectory tracking of an underactuated robot," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 1, pp. 14-20, 2002, <https://doi.org/10.1109/87.974334>.
- [6] T. D. Nguyen, "Discrete-Time Sliding mode control for a class of underactuated mechanical systems with bounded disturbances," *Journal of Computer Science and Cybernetics*, vol. 30, no. 2, pp. 93-105, 2014, <https://doi.org/10.15625/1813-9663/30/2/3569>.
- [7] M. T. Vo *et al*, "Design of Input-Output Feedback Linearization Control for Rotary Inverted Pendulum System," *Journal of Technical Education Science*, vol. 17, no. 2, pp. 26-35, 2022, <https://doi.org/10.54644/jte.69.2022.1120>.
- [8] M. Eom and D. Chwa, "Robust Swing-Up and Balancing Control Using a Nonlinear Disturbance Observer for the Pendubot System With Dynamic Friction," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 331-343, 2015, <https://doi.org/10.1109/TRO.2015.2402512>.
- [9] M. W. Spong and D. J. Block, "The Pendubot: a mechatronic system for control research and education," *Proceedings of 1995 34th IEEE Conference on Decision and Control*, vol. 1, pp. 555-556, 1995, <https://doi.org/10.1109/CDC.1995.478951>.
- [10] B. Lu, Y. Fang, "Gain-adapting coupling control for a class of underactuated mechanical systems," *Automatica*, vol. 125, p. 109461, 2021, <https://doi.org/10.1016/j.automatica.2020.109461>.
- [11] A. D. Luca, R. Mattone and G. Oriolo, "Control of underactuated mechanical systems: application to the planar 2R robot," *Proceedings of 35th IEEE Conference on Decision and Control*, vol. 2, pp. 1455-1460, 1996, <https://doi.org/10.1109/CDC.1996.572718>.
- [12] D. J. Block, M. W. Spong, "Mechanical designer and control for the Pendubot," *Journal of Commercial Vehicles*, vol. 104, 1996, <https://www.jstor.org/stable/44612095>
- [13] K. J. Åström and K. Furuta, "Swing up a pendulum by energy control," *Automatica*, vol. 36, no. 2, pp. 287-295, 2000, [https://doi.org/10.1016/S0005-1098\(99\)00140-5](https://doi.org/10.1016/S0005-1098(99)00140-5).
- [14] M. W. Spong, "The swing-up control problem for the acrobot," *IEEE Control Systems Magazine*, vol. 15, p. 49-55, 1995, <https://doi.org/10.1109/37.341864>.
- [15] V. T. Nguyen, X. T. Vu, H. B. Giap, "Adaptive neural network hierarchical sliding-mode control for pendubot based genetic algorithm optimization," *Intelligent Systems and Networks*, pp. 574-580, 2022, https://doi.org/10.1007/978-981-19-3394-3_66.