A COMPARISON OF TWO SWING-UP ALGORITHMS: ENERGY-BASED AND ADVANCED SLIDING MODE METHODS

Minh-Duy Tran [1], Ngoc-Trung Nguyen [1], Tuan-Anh Le [1], Ngoc-Huy Huynh [1], Quang-Truong Ninh [1], Hoang-Anh-Vu Phi [1], Khanh-Hung Bui [1], Hoang-Chinh Tran [2, *]

[1] Ho Chi Minh City University of Technology and Education (HCMUTE)
Vo Van Ngan Street, No. 01, Ho Chi Minh City, Vietnam
[2] Cao Thang Technical College
Huynh Thuc Khang Street, No. 65, Ben Nghe Ward, District 1, Ho Chi Minh City
***** Corresponding author. E-mail: tranhoangchinh@caothang.edu.vn

Abstract: Pendubot is a popular model in control engineering, often used in laboratories as a typical under-actuated system to study nonlinear control algorithms. It is an accessible model for learners and is widely chosen by laboratories for research purposes. Numerous control algorithms for balancing this system have been developed by researchers. Additionally, the task of swinging the system up from a static equilibrium position to a dynamic equilibrium position is necessary for the model to autonomously reach a suitable position for balance control. In this paper, the authors will compare simulation results of two swing-up control methods: the energy-based (EBM) method and the advanced sliding mode control (ASMC). This comparison of swing-up control algorithms will serve as a foundation for future experimental research on the model.
Keywords: LQR, Energy based method, sliding mode control, pendubot.

1. Introduction

The Pendubot model, fully known as the Pendulum Robot, is a mechanical system that is underactuated, having fewer control inputs than control actuators. It is considered a classical model but plays a crucial role in scientific research, particularly in automatic control theory. This serves as the foundation for creating self-balancing systems in practice. Many issues have been studied using this model, mainly focusing on balance control and swing-up algorithms. Numerous swing-up control algorithms for the pendubot system have been introduced in recent years, but this paper will present two algorithms: EBM and ASMC. In document [1], the author provides detailed insights into the identification and design of the pendubot system and proposes a LQR balance control algorithm. Following this, in document [2], author Hung.V.P investigates two controllers PID and LQR applying a Genetic Algorithm (GA) to optimize the controller parameters at an equilibrium point. The authors of both papers [1] and [2] focus on individual components and have not specifically assessed which control algorithm will provide long-term stability after swing-up. An energy-based swing-up method is presented in document [3]. In [4], the author employs the EBM swing-up algorithm to simulate and compare the evaluation of balance control algorithms post swing-up. However, the author delves only into one swing-up technique, making it difficult to ascertain which swing-up method is better suited for the pendubot system. Document [5] studies the use of sliding mode swing-up control for the pendubot. The author of document [6] builds upon these sliding control methods, further enhancing them by adding rrr to follow the pre-defined trajectory. This upgrade is quite beneficial as it minimizes energy during free rotation. For successful swing-up, it is necessary to combine balance algorithms to maintain the stable state of the system. Comparative studies of balance algorithms also assist the authors in selecting the appropriate balance control algorithm for this research. This paper focuses solely on evaluating and comparing swing-up algorithms, hence investigating only one balance control algorithm: LQR, to stabilize the system after swing-up. We present a comparison of the two swing-up control techniques, including EBM and ASMC. The swing-up balance controller for the pendubot system is implemented using LQR algorithm. The simulation results are presented for a comprehensive evaluation.

2. The Dynamic Mathematical Equations

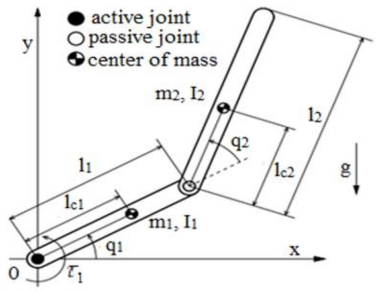

Fig. 1. Structure of the pendubot description

The dynamic equations of the system based on the Euler-Lagrange formulation are as follows [1]:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \quad (1)$$

where:

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \qquad \tau = \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (2)$$

$$D(q) = \begin{bmatrix} \beta_1 + \beta_2 + 2\beta_3 \cos q_2 & \beta_2 + \beta_3 \cos q_2 \\ \beta_2 + \beta_3 \cos q_2 & \beta_2 \end{bmatrix} \quad (3)$$

$$C(q,\dot{q}) = \begin{bmatrix} -\beta_3 \sin(q_2)\, \dot{q}_2 & -\beta_3 \sin q_2\, \dot{q}_2 - \beta_3 \sin q_2 \dot{q}_1 \\ \beta_3 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix} \quad (4)$$

$$G(q) = \begin{bmatrix} \beta_4 g \cos q_1 + \beta_5 \cos(q_1 + q_2) \\ \beta_5 \cos(q_1 + q_2) \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} \ddot{q_1} \\ \ddot{q_2} \end{bmatrix} = D(q)^{-1}\tau - D(q)^{-1}C(q,\dot{q})\dot{q} - D(q)^{-1}g(q) \quad (6)$$

$$\beta_1 = m_1 {l_{c1}}^2 + m_2 {l_1}^2 + I_1;\ \beta_2 = m_2 {l_{c2}}^2 + I_2$$
$$\beta_3 = m_2 l_1 l_2;\ \beta_4 = m_1 l_1 + m_2 l_1;\ \beta_5 = m_2 l_2 \quad (7)$$

The external torque is converted to the input voltage value as follows:

$$\tau = a_1 u - a_2 \dot{q_1} - a_3 \ddot{q_1} \quad (8)$$

With: $a_1 = \frac{K_t}{R_m} \qquad a_2 = \frac{K_t K_b}{R_m} \qquad a_3 = J_m$ (9)

In this paper, the authors focus on examining the Pendubot system at the TOP position. x_1 and x_3 correspond to the angular deviations of link 1 and link 2 relative to the Oy axis, respectively. x_2 and x_4 are the angular velocities of link 1 and link 2, respectively. The state variables are redefined as follows:

$$x_1 = q_1 - \frac{\pi}{2} \qquad x_2 = \dot{x_1} = \dot{q_1}$$
$$x_3 = q_2 \qquad x_4 = \dot{x_3} = \dot{q_2} \quad (10)$$

Tab 1. System parameters.

Symbol	Value	Unit	Description
m_1	0.155	kg	Mass of link 1
l_1	0.16	m	Length of link 1
l_{c1}	0.076	m	Distance from active joint to center of mass of link 1
I_1	0.0021	$kg.m^2$	Moment of inertia of link 1
m_2	0.078	kg	Mass of link 2
l_2	0.2	m	Length of link 2
l_{c2}	0.073	m	Distance from passive joint to center of mass of link 2
I_2	0.0012	$kg.m^2$	Moment of inertia of link 2
K_t	0.0926	Nm	Torque constant
K_b	0.0061	V	Back EFM constant (Force constant of motor)
R_m	2.826	Ω	Resistor of rotor
J_m	0.0023	$kg.m^2$	Moment of inertia of rotor

3. Control Algorithms

3.1. Balancing Control LQR

The LQR controller is built based on the principle of state feedback. LQR is commonly used to control SIMO (Single Input, Multiple Output) systems. In general, the controller takes the system's state as input, performs calculations, and converts it into a control signal for the target system.

Consider a system with a control signal $(u \neq 0)$:

$$\dot{x} = Ax + Bu \quad (11)$$

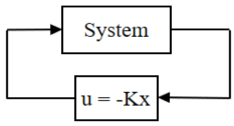

Fig. 2. Block diagram of the principle of the system controlled by the LQR controller.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} q_1 - \frac{\pi}{2} \\ \dot{q_1} \\ q_2 \\ \dot{q_2} \end{bmatrix} \quad (12)$$

$$A = \begin{bmatrix} \frac{\partial \dot{q_1}}{\partial x_1} & \frac{\partial \dot{q_1}}{\partial x_2} & \frac{\partial \dot{q_1}}{\partial x_3} & \frac{\partial \dot{q_1}}{\partial x_4} \\ \frac{\partial \ddot{q_1}}{\partial x_1} & \frac{\partial \ddot{q_1}}{\partial x_2} & \frac{\partial \ddot{q_1}}{\partial x_3} & \frac{\partial \ddot{q_1}}{\partial x_4} \\ \frac{\partial \dot{q_2}}{\partial x_1} & \frac{\partial \dot{q_2}}{\partial x_2} & \frac{\partial \dot{q_2}}{\partial x_3} & \frac{\partial \dot{q_2}}{\partial x_4} \\ \frac{\partial \ddot{q_2}}{\partial x_1} & \frac{\partial \ddot{q_2}}{\partial x_2} & \frac{\partial \ddot{q_2}}{\partial x_3} & \frac{\partial \ddot{q_2}}{\partial x_4} \end{bmatrix};\ B = \begin{bmatrix} \frac{\partial \dot{q_1}}{\partial \tau_1} \\ \frac{\partial \ddot{q_1}}{\partial \tau_1} \\ \frac{\partial \dot{q_2}}{\partial \tau_1} \\ \frac{\partial \ddot{q_2}}{\partial \tau_1} \end{bmatrix}. \quad (13)$$

At the equilibrium state with positions $x_1 = 0$; $x_2 = 0$; $x_3 = 0$; $x_4 = 0$; $u = 0$ the system parameters in Tab 1 can be used to calculate matrices A and B. To ensure that the simulation **Error! Reference source not found.**calculations in this paper closely reflect reality, it is necessary to convert matrices A and B from a continuous to a discrete system. In other words, this system will be transformed into a discrete system. In this study, a sampling time of 10ms is chosen. The conversion of matrices A and B from a continuous to a discrete system is supported by MATLAB using the command below, resulting in the new matrices A and B as follows:

```
[Ad, Bd] =c2d (A, B,0.01)
```
(14)

$$Ad = \begin{bmatrix} 1.0026 & 0.01 & -0.0002 & 0 \\ 0.5259 & 1.0026 & -0.0438 & -0.0002 \\ -0.002 & 0 & 1.0029 & 0.01 \\ 0.4789 & -0.002 & 0.3974 & 1.0020 \end{bmatrix} \quad (15)$$

$$Bd = \begin{bmatrix} 0.0188 \\ 3.7707 \\ -0.029 \\ -5.8003 \end{bmatrix} \quad (16)$$

The matrices Q and R are selected to define control quality. The Q matrix represents the desired quality of control, while the R matrix reflects the energy consumption associated with the control process.

Weighting matrix Q:

$$Q = \begin{bmatrix} Q_1 & 0 & 0 & 0 \\ 0 & Q_2 & 0 & 0 \\ 0 & 0 & Q_3 & 0 \\ 0 & 0 & 0 & Q_4 \end{bmatrix} \quad (17)$$

Energy coefficient R (R > 0)

Solving the Riccati equation, the solution is the matrix S:

$$A^T S + SA - SBR^{-1}B^T S + Q = 0 \quad (18)$$

Control matrix K:

$$K = R^{-1}B^T S \quad (19)$$

Calculate the matrix K in MATLAB:

$$K = dlqr(Ad, Bd, Q, R) \quad (20)$$

3.2. Swing-up Control

The swing-up controller will guide the two links of the pendubot system from the DOWN position ($q_1 = -\frac{\pi}{2}$; $q_2 = 0$) to the TOP equilibrium position ($q_1 = \frac{\pi}{2}$; $q_2 = 0$). This section presents two swing-up methods: EBM and ASMC. The first part discusses the EBM as referenced in [3], followed by the ASMC in [6].

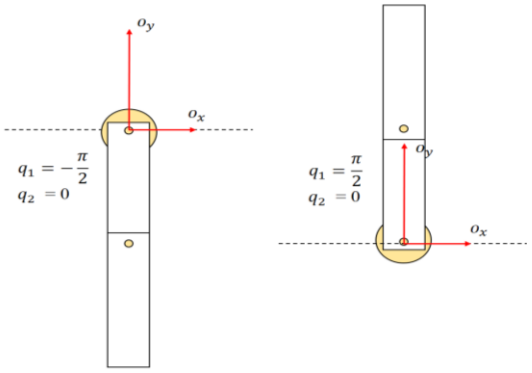

Fig. 3. DOWN/TOP equilibrium positions of pendubot

The essence of a swing-up controller is to initially use a swing-up controller (either EBM or ASMC) to generate rotational kinetic energy in link 1. As the system approaches the equilibrium position, it switches to a balance controller once the Switch Condition is met. The combined algorithm in this paper consists of EBM-LQR and ASMC-LQR.

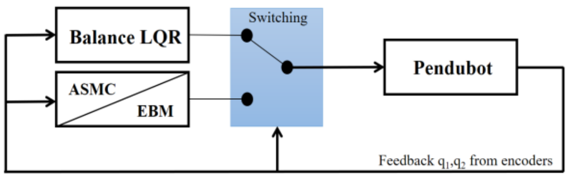

Fig. 4. Swing-up control structure

The angle values of link 1 and link 2 are feed back to determine the Switch condition . To select an appropriate Switch condition, we need to observe the state variables around the equilibrium position to determine the suitable Switch condition.

3.3. Energy-Based Method

The EBM swing-up control algorithm is based on the energy method and the passive properties of the pendubot. The author in document [3] demonstrated that link 1 converges to the upright position, and link 2 oscillates to the TOP position. Energy calculations are presented as follows:

$$\lim_{t\to\infty} E(q,\dot{q}) = E_p$$
$$\lim_{t\to\infty} E(q_1) = \frac{\pi}{2} \quad (21)$$
$$\lim_{t\to\infty} E(\dot{q}_1) = 0$$

With

Total energy $E(q,\dot{q}) = \frac{1}{2}\dot{q}^T D(q)\dot{q} + P(q)$ (22)

Potential of pendubot:

$$P(q) = \beta_4 g sin(q_1) + \beta_5 g sin(q_1 + q_2) \quad (23)$$

Energy of pendubot at upright position:

$$E_p = (\beta_4 + \beta_5)g \quad (24)$$

Lyapunov function:

$$V = \frac{1}{2}K_E(E - E_p)^2 + \frac{1}{2}K_p(q_1 - \frac{\pi}{2})^2 + \frac{1}{2}K_D\dot{q}_1{}^2 \quad (25)$$

where K_E, K_P, K_V is positive contant parameters. Derivative V respect to time t

$$\dot{V} = K_E\dot{E}(E - E_p) + K_p\,\dot{q}_1\left(q_1 - \frac{\pi}{2}\right) + K_D\ddot{q}_1\dot{q}_1 \quad (26)$$

$$\dot{E} = \dot{q}^T B\tau_1 = \dot{q}_1\tau_1 \quad (27)$$

Substitute (27) into (26), then we obtain:

$$\dot{V} = \dot{q}_1\left(K_E(E - E_p)\tau_1 + K_p\left(q_1 - \frac{\pi}{2}\right) + K_D\ddot{q}_1\right) \quad (28)$$

From (1), after some calculations it gives

$$\ddot{q}_1 = B^T\ddot{q} = B^T D^{-1}(B\tau_1 - C\dot{q} - G) \quad (29)$$

Substituting (29) into (28), it yields

$$\Psi\tau_1 = -K_p\left(q_1 - \frac{\pi}{2}\right) + K_D B^T D^{-1}(C\dot{q} + G) - K_V\dot{q}_1 \quad (30)$$

Where $\Psi = K_E(E - E_p) + K_D B^T D^{-1}B \neq 0$ (31)

Thence, swing up controller is

$$\tau_1 = \frac{-K_p\left(q_1 - \frac{\pi}{2}\right) + K_D B^T D^{-1}(C\dot{q} + G) - K_V\dot{q}_1}{\Psi} \quad (32)$$

3.4. Advanced Sliding Mode Controller

Sliding mode control has several advantages over other controllers. It does not require an exact model, as it can respond effectively to dynamic models with modeling components and disturbances. As a nonlinear controller, it offers greater efficiency compared to linear controllers. The advanced sliding mode controller is a minor enhancement of the standard sliding mode controller, allowing q_1 to track a pre-defined trajectory signal r for Swing-up. This approach provides more stability than simply allowing q_1 to rotate with an undefined angle. The state-space equations of the pendubot system are defined as follows (6):

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= f_1 + g_1\tau \\ \dot{x_3} &= x_4 \\ \dot{x_4} &= f_2 + g_2\tau \end{aligned} \qquad (33)$$

The error e is now defined as follows:

$$e = x_1 + x_3 - r; \dot{e} = \dot{x_1} + \dot{x_3} - \dot{r} \qquad (34)$$

Construct the sliding surface:

$$s = ce + \dot{e} = c(x_1 + x_3 - r) + (x_2 + x_4 - \dot{r}) \qquad (35)$$

Derivative with respect to c is a constant:

$$\dot{s} = c\dot{e} + \ddot{e} = c(x_2 + x_4) + f_1 + f_2 + (g_1 + g_2)\tau \qquad (36)$$

Lyapunov function:

$$V = \frac{1}{2}s^2 \qquad (37)$$

To ensure that V decreases towards 0, it is necessary to choose:

$$\dot{V} = s.\dot{s} < 0 \qquad (38)$$

Choose: $\dot{s} = -Ksign(s)$ (39)

From (36)(36) to (39) we have:

$$\tau = \frac{-Ksign(s) - cx_2 - cx_4 + c\dot{r} + \ddot{r} - f_1 - f_2}{g_1 + g_2} \qquad (40)$$

At the TOP position, the reference signal r that q_1 needs to track is:

$$r = \begin{cases} \pi \sin(4.35t) - \frac{\pi}{2} : t < \frac{\pi}{8.7} \\ \frac{\pi}{2} : t \geq \frac{\pi}{8.7} \end{cases} \qquad (41)$$

4. Simulation Program and Result

To ensure that the simulation results closely reflect reality, the authors selected system parameters as shown in Table 1. For accurate application of control algorithms and simulation, as well as to provide evaluation results, the entire study was conducted using MATLAB and the SIMULINK tool. The Saturation block is used to limit the motor supply voltage to a range of -24V to 24V, making it suitable for future experimental studies. To optimize the control parameters in the simulation, GA is also used in this study to determine the control parameters that yield the best performance. For the LQR controller, the control matrix K needs to be optimized. K is influenced by the weighting matrix Q and the energy coefficient R. GA is used to find the optimal values for Q and R.

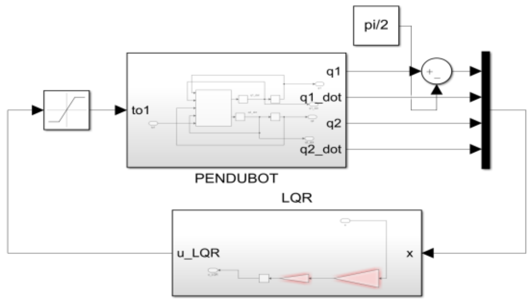

Fig. 5. LQR controller program for the Pendubot.

The value of K is calculated based on the optimal values of Q and R found through GA, as presented above to achieve the best control quality. Parameters of the matrix K: **K= [-6.3453 -1.3245 -5.7979 -0.9292]**

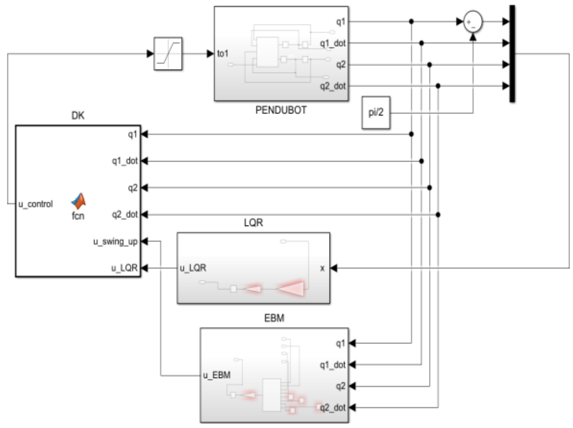

Fig. 6. EBM – LQR controller program for the Pendubot.

According to the document [7] Switching condition is chosen as:

$$\left(\frac{\pi}{2} - q_1\right) + 0.1\dot{q}_1 + q_2 + 0.1\dot{q}_2 < \xi \qquad (42)$$

The control parameters sought by the GA algorithm are as follows: K_e = 246.3, K_v = 187, K_p = 652.1 and K_d = 3. The choice of parameter ξ also greatly impacts the system's control quality. If ξ is too large, the timing for switching may prevent the LQR controller from stabilizing the system, because the angle error q1compared to the set value remains too large, and the angular velocity has not yet met the conditions needed for the control algorithm to stabilize the system. When ξ is smaller, the system's control quality improves after the swing-up. However, if ξ is too small, it becomes challenging to achieve this condition during the swing-up process. In this paper, after some simulation trials, we chose ξ =1.5.

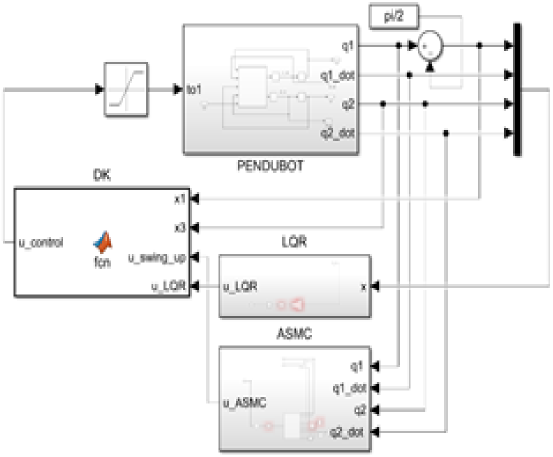

Fig. 7. ASMC – LQR controller program for Pendubot.

The control parameters found by GA are as follows: c = 628 and K = 9547 . For the ASMC, the switching condition is chosen based on the angles of link 1 and link 2 with the conditions $|q_1|<0.3$ & $|q_2|<0.3$. After multiple simulation trials, the value 0.3 was found to be the most suitable for triggering the switch.

4.1. Simulation Results of the LQR

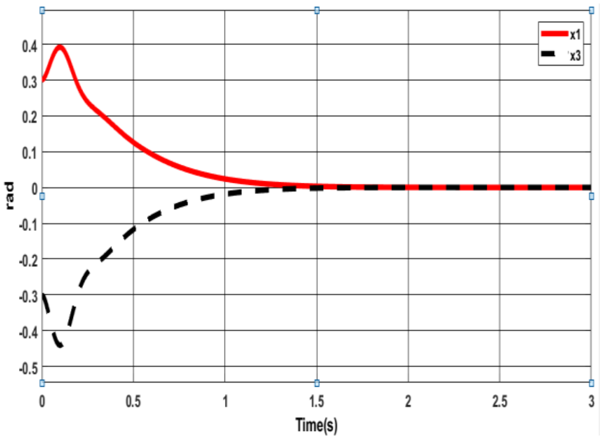

Fig. 8. The simulation results of the LQR algorithm.

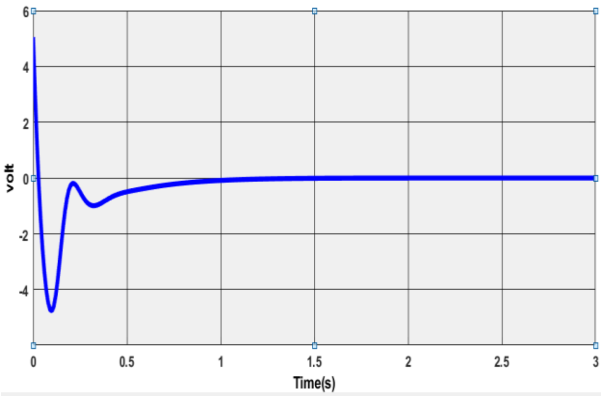

Fig. 9. The simulation results of the input voltage supplied to the motor using the LQR algorithm

Link 1 reaches a vertical position and stops moving after 1.5 seconds, as does Link 2. Therefore, the LQR controller stabilizes the Pendubot system in 1.5 seconds. Simulation results for the LQR controller indicate that the system maintains good stability with minimal oscillations in the links during control. This results in a fast settling time of 1.5 seconds and a low steady-state error of 0.02 rad. Although there is some initial overshoot at the start of the control process, but the system quickly stabilizes, thus minimally impacting control quality. This is partly due to the large initial set values, allowing the system to effectively respond during the swing-up process. The results confirm that LQR is a suitable control algorithm for experimental applications with the Pendubot. With high control quality, LQR can be combined with other control algorithms to create Swing-Up controllers, such as ASMC-LQR and EBM-LQR.

4.2. Simulation Results of the EBM Controller

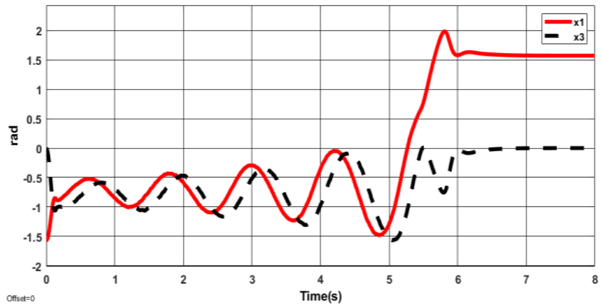

Fig. 10. Output response of the Pendubot's two links during the swing-up phase using the EBM algorithm.

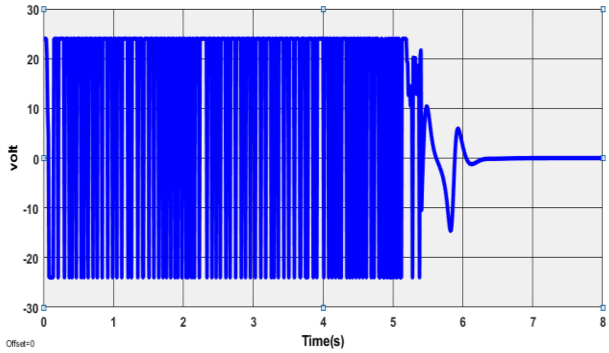

Fig. 11. The simulation results of the input voltage to the motor during the swing-up process of the EBM

Link 1 reaches a vertical position and stops moving after 7.7 seconds, while Link 2 does so after 6 seconds. Therefore, the EBM-LQR controller completes the swing-up and stabilizes the Pendubot system in 7.7 seconds. For the EBM-LQR controller, simulation results show that the system oscillates more strongly during the swing-up phase, reaching the TOP position within 6 seconds before switching to the LQR controller once the required conditions are met. Despite significant oscillations at the beginning of the control process, the system gradually stabilizes as it approaches the TOP position, resulting in a final steady-state error of 0.05

rad. The energy used during the swing-up process is large due to strong and prolonged oscillations.

4.3. Simulation Results of the ASMC Controller

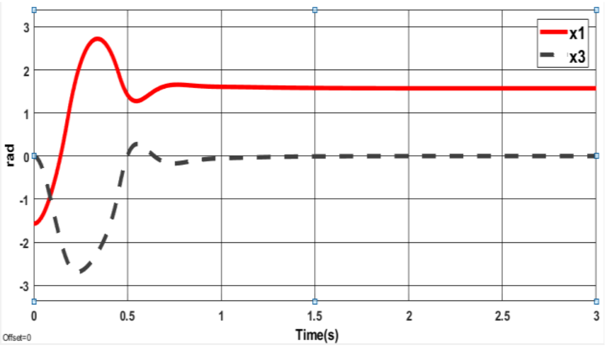

Fig. 12. Output response of the Pendubot's two links during the swing-up phase using the ASMC algorithm.

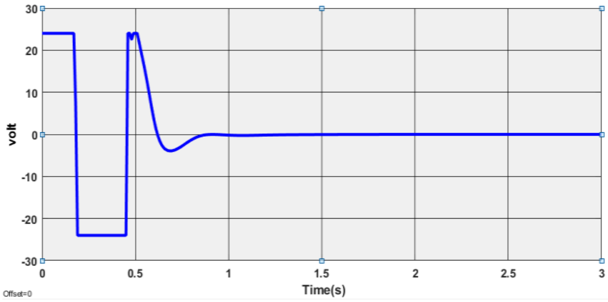

Fig. 13. The simulation results of the input voltage to the motor during the swing-up process of EBM algorithm

Links 1 and 2 reach a vertical position and stop moving after 1.2 seconds. Thus, the ASMC-LQR controller stabilizes Pendubot in 1.2 seconds. Both links exhibit minimal oscillation during the swing-up phase, with the energy generated for Link 1 being just enough to swing up the system, resulting in a fast settling time of 1.2 seconds and a low steady-state error of 0.03 rad. Although there is significant initial overshoot in both links during the swing-up process, they quickly follow the pre-set trajectory and stabilize at the TOP equilibrium point. The energy required to generate the kinetic energy to rotate Link 1 during the swing-up is not excessively high.

Simulation results indicate that the control quality of ASMC is superior to that of EBM for swinging up Pendubot. Not only does ASMC-LQR achieve a shorter settling time and smaller error, but it also requires less energy to swing up the system compared to EBM-LQR, as shown in Fig. 11 and Fig. 13.

Tab 2. Comparison of Results

Algorithm	Error(rad)	Settling time (s)
EBM-LQR	0.05	7.7
ASMC-LQR	0.03	1.2

5. Conclusion

Through simulation and comparison, it can be seen that both controllers successfully perform the swing-up and balance Pendubot. Among them, the ASMC-LQR controller stabilizes the system the fastest and with the least oscillation. For the EBM controller, the parameters found through GA provide a good output response, but there is still too much oscillation, and the settling time is slower than that of the other controller. The energy required to swing up the system is also a significant factor in controlling the Pendubot. For these reasons, it is clear that the ASMC-LQR algorithm offers better control quality and is the most viable choice when tested on an experimental model.

Acknowledgement

This paper belongs to project for students of Ho Chi Minh City University of Technology and Education (HCMUTE) in year 2025. Mr. Minh-Duy Tran is key person of this project. The project is funded by HCMUTE. We also want to give thanks to PhD. Van-Dong-Hai Nguyen and Ms. Eng. Thi-Thanh-Hoang Le (HCMUTE) due to their supervision.

6. References

[1] Block D.J. et al: "Mechanical Design and Control of the Pendubot," SAE Transactions, vol. 104, pp. 36-43, 1995.

[2] Hùng P.V.: "Nghiên cứu điều khiển cánh tay robot thiếu dẫn động hai bậc tự do – Pendubot", Luận văn Thạc sĩ, Trường Đại học Đà Nẵng, 2013.

[3] Fantoni et al: "Energy Based Control of the Pendubot," IEEE Transactions on Automatic Control, vol. 45, 2000.

[4] Duc T.M. et al: "A Comparison of Control Schemes for Under-Actuated Pendubot System", Robotica & Management, Vol. 28, No. 1, pp. 53-58, 2023.

[5] Qian D., Yi J., Zhao D.: "Hierarchical Sliding Mode Control to Swing up a Pendubot", Proc. of the American Control Conf., pp. 5254-5259, 2007.

[6] Kien C.V. et al: "Swing Up and Balancing Implementation for the Pendubot UsingAdvanced Sliding Mode Control," International Conference on Artificial Intelligence and Control Automation, 2015.

[7] Bao H.G. et al: "Comparison between two swing-up algorithms: partial feedback linear and energy based method", Journal of Technical Education Science, No.55, 2019.