JOURNAL OF MECHANICAL ENGINEERING AND SCIENCES
ISSN: 2289-4659 e-ISSN: 2231-8380
VOLUME 18, ISSUE 2, 2024, 10017 - 10025
DOI: https://doi.org/10.15282/jmes.18.2.2024.4.0791

RESEARCH ARTICLE

Numerical and experimental investigation for swing-up control of an inverted pendulum using Arduino microcontroller

M. M. Donatoni[1]*, F. R. Chavarette[2], E. Preto[1], D. C. J. Karmouche[1,2]

[1] Faculty of Engineering of Ilha Solteira, Universidade Estadual Paulista, 15385-000 Ilha Solteira, São Paulo, Brazil
Phone: +55 (18) 3743-1085; Fax: +55 (18) 3743-1085
[2] Institute of Chemistry, Universidade Estadual Paulista, 14800-060 Araraquara, São Paulo, Brazil

ABSTRACT - The Inverted Pendulum is a classic control problem, it has non-linear dynamics, is underactuated and is naturally unstable. Thus, developing a system capable of controlling it goes through challenges such as modeling, design requirements, and implementation of the control hardware. This work proposes the swing-up of the linear inverted pendulum using the energy method with adjustable parameters, followed by a linear quadratic regulator controller stabilization. This work demonstrates how the system can be implemented using an Arduino microcontroller to acquire state variables and control commands. Furthermore, as a highlight, the implemented algorithm indicates a way to stabilize the sampling frequency, making the derivative process stable in the applied hardware and optimizing the control. The applied method was efficient in performing the swing-up, consistent with the simulations, and as effective as seen in the literature.

ARTICLE HISTORY
Received : 18th Oct. 2023
Revised : 27th Feb. 2024
Accepted : 22nd Mar. 2024
Published : 28th June 2024

KEYWORDS
Inverted Pendulum
Swing-up
Energy-based
LQR
Microcontroller
Arduino

1. INTRODUCTION

Control theory involves the implementation of autonomous dynamic systems. Historically, this process began with purely mechanical regulatory mechanisms, such as the early-level float devices dating back to the 3rd century before century. With the advent of the Industrial Revolution, a centrifugal governor for controlling the speed of a steam engine was developed [1-4]. The emergence of computers was a watershed moment in implementing controllers because they could simplify the monitoring and manipulation of variables in controlled systems, enabling previously impractical applications. One of these applications is the Linear Inverted Pendulum (LIP) system, which has become a benchmark in control theory due to its nonlinearity, under-actuation, and instability [5].

Among the existing approaches, one that adds complexity to the LIP problem is implementing the "swing-up" technique. It relies on implementing a nonlinear control law that makes the system converge from its naturally stable position (downward) to a naturally unstable position (upward). Addressing this topic, Wang et al. [6], Muskinja and Tovornik [7], and Tao et al. [8] employed Fuzzy Logic with Takagi–Sugeno models composed of consecutive "if-then" logical rules. While they achieved good results, the implementation process can be considered exhaustive and error-prone. Another approach to swing-up control involves the application of machine learning algorithms, typically using reinforcement learning, as done by Si and Wang [9], Morimoto and Doya [10], and Riedmiller [11]. Based on trial and error, this type of process can be harmful in a real system, damaging or posing risks to the people involved. However, Riedmiller [11] conducted experimental work on a Simple Inverted Pendulum, which has an actuator that acts directly on the pendulum. Finally, a third approach available in the literature involves energy-based methods. In this approach, it is considered that the acceleration of its actuator can control the pendulum's energy, and the control law aims to guide the system to a state where its energy is equivalent to the pendulum being up and at rest. In Yoshida [12], Bugeja [13], and Yang et al. [5], it is noticeable that the control laws proposed for swing-up are relatively more straightforward to implement when compared to Fuzzy Logic and machine learning, especially in the latter work where the control law is expressed by just one equation. Additionally, there is concern regarding the actuator's position in all three cases, which is of great importance in a real system due to its limitations, such as rail length.

This work performs the numerical and experimental implementation of the swing-up method developed by Yang et al. [5] using Python for real-time calculation of control commands and the Arduino Due microcontroller for state variable acquisition and control signal transmission. Similar work is seen in Bellati et al. [14], and Graham and Turkoglu [15], both use the energy-based approach for swing-up. The first stabilizes a Futura's Pendulum in the upward position with a Local Linear Regulator. In contrast, the second does it with an LIP and uses a Proportional Integral Derivative (PID) controller. The Arduino platform is used in both cases. However, they are dependent on the proprietary and high-cost Simulink platform. This paper presents an alternative implementation form using Python, an open-source software. Furthermore, after the pendulum reaches the upright position, a Linear Quadratic Regulator (LQR) controller is used for system stabilization. This robust control technique has been replicated in previous works such as Prasad et al. [16], Lee et al. [17], and Irfan et al. [18]. The control of the LIP is divided into consecutive stages: swing-up and stabilization. To

*CORRESPONDING AUTHOR | M. M. Donatoni et al. | ✉ matheus.m.donatoni@unesp.br

achieve this, mathematical modeling of the LIP system and the controllers for swing-up and LQR are performed, allowing for the numerical validation of the method. Subsequently, experimental reproduction of the method is carried out, and a comparison is made with the numerical data.

The division of this work is as follows: the section "Materials and Methods" is subdivided into "Mathematical Model of the Linear Inverted Pendulum," which presents the mathematical development of the linear inverted pendulum; "Swing-Up and Stabilization Strategy," which elaborates on the swing-up and stabilization strategy for the pendulum; "Experimental Implementation," which describes the steps used to implement the system in practice; the section "Results and Discussion" displays and discusses the parameters used and the results obtained from the implementation of the method; finally, in the section "Conclusions," the concluding remarks of the work are provided.

2. MATERIALS AND METHODS

2.1 Mathematical Model of the Linear Inverted Pendulum

As it is a system with two degrees of freedom, the mathematical model of the LIP presents two coupled differential equations. Based on the system diagram shown in Figure 1, the application of Newton's Second Law of Motion results in the following characteristic equations:

$$(J + m\,l^2)\,\ddot{\alpha} = mlg\sin\alpha + ml\ddot{x}\cos\alpha \tag{1}$$

$$(M+m)\ddot{x} = ml\ddot{\alpha}\cos\alpha - ml\,\dot{\alpha}^2\sin\alpha - B\dot{x} + F \tag{2}$$

where J represents the moment of inertia of the pendulum about its center of mass, m is its mass, l is its length from the pivot point to the center of mass, M is the mass of the cart, B is the friction between the cart and the rail, and g is the acceleration due to gravity. The angular displacement of the pendulum is represented by α, with positive values in the counterclockwise direction of rotation, and the linear displacement of the cart is represented by x, with positive values to the right.

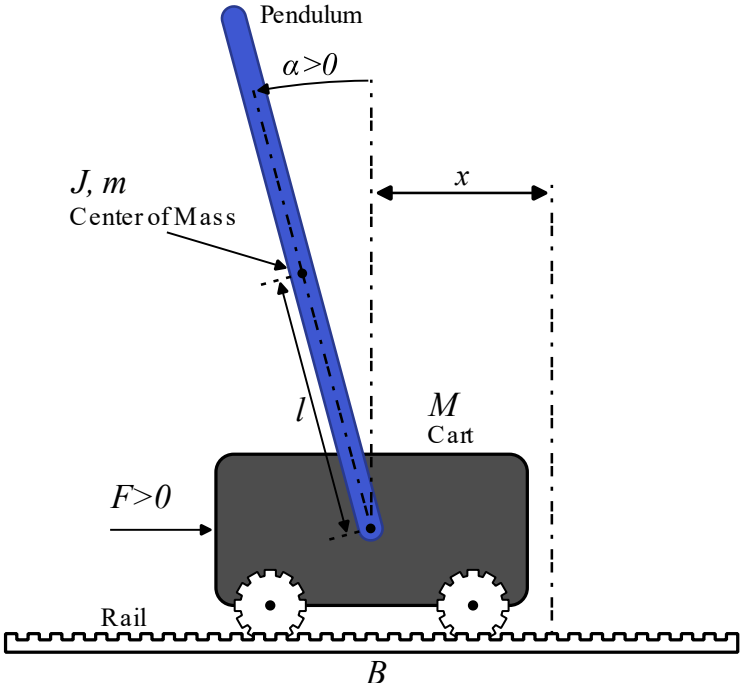

Figure 1. Schematic of the Linear Inverted Pendulum with conventions for positive directions of state variables

2.2 Swing-Up and Stabilization Strategy

Initially, the cart's acceleration is zero, $\ddot{x} = 0$, so Eq. (1) simplifies to Eq. (3).

$$(J + ml^2)\ddot{\alpha} = mlg\sin\alpha \tag{3}$$

As done by Yang et al. [5], it will be defined that when the pendulum is upright and at rest, its energy is zero. Therefore, the expression representing the pendulum's energy E_p is given by Eq. (4).

$$E_p = \frac{1}{2}(J + ml^2)\dot{\alpha}^2 + mlg(\cos\alpha - 1) \tag{4}$$

With these definitions and varying initial conditions, it is possible to visualize the orbits of constant energy of the pendulum. This is depicted in Figure 2, where the orbits are represented by purple and red lines, as well as a blue line,

respectively represent internal orbits (where E_p is between 0 and $-2\,mlg$), transition orbits (where, $E_p = 0$), and external orbits (where, E_p is greater than zero). The circular black points $(\alpha, \dot{\alpha}) = (\pm\pi, 0)$ represent the initial conditions, i.e., the pendulum down at rest, while the diamond-shaped points $(\alpha, \dot{\alpha}) = (\pm 2\pi, 0)$ and the origin represent the pendulum up at rest.

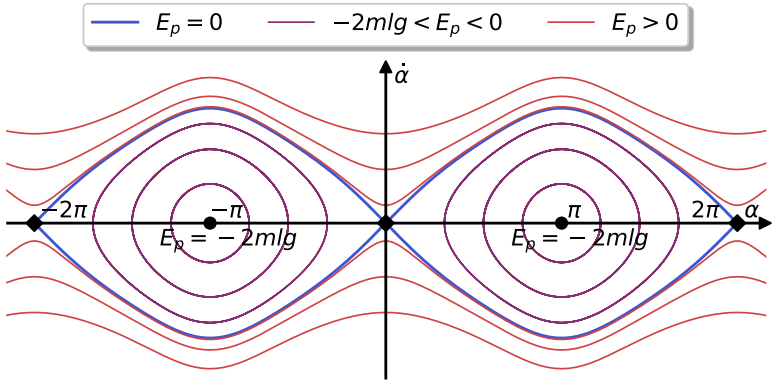

Figure 2. Orbits of constant energy for the pendulum in relation to the variation of angular position $[\alpha]$ and angular velocity $[\dot{\alpha}]$

The swing-up strategy defines a control law that can drive the velocity of the cart and the energy of the pendulum towards zero over time. The goal is achieved without the cart exceeding the limits of the track. Yang et al. [5] proposes a candidate Lyapunov function represented by:

$$V = \frac{1}{2}\left(E_p^2 + ml\lambda\dot{x}^2\right) \tag{5}$$

The parameter $\lambda > 0$ is a design value and can be adjusted to meet control requirements. Next, the derivative of V with respect to time is computed:

$$\frac{dV}{dt} = E_p\dot{E_p} + ml\lambda\dot{x}\ddot{x} \; = E_p[(J + ml^2)\dot{\alpha}\ddot{\alpha} - mlg\sin\alpha\,\dot{\alpha}] + ml\lambda\dot{x}\ddot{x}$$

and,

$$\frac{dV}{dt} = E_p ml\cos\alpha\,\dot{\alpha}\ddot{x} + ml\lambda\dot{x}\ddot{x} \; = \ddot{x}ml\left(E_p\cos\alpha\,\dot{\alpha} + \lambda\dot{x}\right) \tag{6}$$

In the development of Eq. (6), the term $(J + ml^2)\ddot{\alpha}$ was replaced by the value from Eq. (1). Next, a control law is proposed:

$$u_{as} = -u_a\left(E_p\cos\alpha\,\dot{\alpha} + \lambda\dot{x}\right) \tag{7}$$

Since the control is exerted through a force applied to the cart, it enables the control of its acceleration, so $u_{as} = \ddot{x}$. Thus, dV/dt is rewritten as:

$$\frac{dV}{dt} = -mlu_a\left(E_p\cos\alpha\,\dot{\alpha} + \lambda\dot{x}\right)^2 \tag{8}$$

The term $u_a > 0$ is a second design parameter that generates a proportional gain, while λ regulates the gain relative to the cart's velocity. If the Lyapunov function, V defined earlier tends to zero, then the terms E_p and $\dot{x}$ will also necessarily tend to zero. This was the premise used to determine the expression for u_{as} in Eq. (7). However, from the expression obtained in Eq. (8), dV/dt can assume the value of zero, which would lead the system to not converge to the transition orbit. However, Yang et al. [5] proves that for any initial condition different from $(\alpha, \dot{x}, \dot{\alpha}) = (\pm\pi, 0, 0)$, the system will reach the transition orbit. For conditions where the pendulum has any energy value greater than $-2mlg$, even if dV/dt is zero at that moment, there will be future motion, making dV/dt negative again, which leads the system to converge to the transition orbit. Therefore, an initial impulse is necessary for the system to initiate the swing-up process. Once the transition orbit is reached, there is a switch from the nonlinear controller to an LQR controller that ensures local stability. For its implementation, the system must be linearized and represented in state-space (SS) form [15]. Initially, the state variable is defined as:

$$x = [x \quad \alpha \quad \dot{x} \quad \dot{\alpha}]^T \tag{9}$$

Its derivative is represented as:

$$\dot{x} = [\dot{x} \quad \dot{\alpha} \quad \ddot{x} \quad \ddot{\alpha}]^T \tag{10}$$

It is necessary to determine the functions representing $\ddot{x}$ and $\ddot{\alpha}$. To do this, manipulating Eqs. (1) and (2), making the appropriate substitutions, yields:

$$\ddot{\alpha} = \frac{-(ml)^2(\sin\alpha\cos\alpha)\dot{\alpha}^2 - (ml\cos\alpha)B\dot{x} + (M+m)mlg\sin\alpha + (ml\cos\alpha)F}{(M+m)(J+ml^2) - (ml\cos\alpha)^2} \tag{11}$$

$$\ddot{x} = \frac{-(J+ml^2)(ml\sin\alpha)\dot{\alpha}^2 - (J+ml^2)B\dot{x} + (ml)^2 g\sin\alpha\cos\alpha + (J+ml^2)F}{(M+m)(J+ml^2) - (ml\cos\alpha)^2} \tag{12}$$

The state-space representation requires linear equations. Eqs. (11) and (12) can be linearized using Taylor series up to the first order around the system's origin $x_0 = (x, \alpha, \dot{x}, \dot{\alpha}) = (0,0,0,0)$. Therefore, these equations can be expressed in their linear form as follows:

$$\ddot{\alpha} = \frac{-mlB\dot{x} + (M+m)mlg\alpha + mlF}{(M+m)(J+ml^2) - (ml)^2} \tag{13}$$

$$\ddot{x} = \frac{-(J+ml^2)B\dot{x} + (ml)^2 g\alpha + (J+ml^2)F}{(M+m)(J+ml^2) - (ml)^2} \tag{14}$$

As the control is performed by an electric motor, the control force F exerted on the cart should be replaced by an equivalent expression in terms of the control voltage, V_m:

$$F = A_m V_m - B_m\dot{x} \tag{15}$$

where, A_m represents a motor gain constant, and B_m represents motor damping due to dissipative effects. Defining $J_{eq} = (M+m)(J+ml^2) - (ml)^2$ and $B_{eq} = B + B_m$, the state matrices can be written as follows:

$$A = \frac{1}{\mathrm{J_{eq}}}\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & (ml)^2 g & -(J+ml^2)B_{eq} & 0 \\ 0 & (M+m)mlg & -mlB_{eq} & 0 \end{bmatrix}$$
$$B = \frac{A_m}{J_{eq}}\begin{bmatrix} 0 \\ 0 \\ (J+ml^2) \\ ml \end{bmatrix},\ C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}^T,\ D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{16}$$

In this way, the state-space representation is given by:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{17}$$

where, $u = V_m$ represents the control voltage, and y is the system's output. The LQR controller is of the Full State Feedback (FSF) type, and its control law is $u = -Kx$. Therefore, the first equation in Eq. (17) is presented as:

$$\dot{x} = (A - BK)x \tag{18}$$

where, K represents the gain matrix that minimizes the performance index subject to (18), given by:

$$J_p = \int_0^\infty (\mathrm{x}^T Qx + u^T Qu) \tag{19}$$

The calculation of the gain is performed by:

$$K = R^{-1}B^T P \tag{20}$$

where, P is the unique positive definite matrix that can solve the Riccati equation:

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \tag{21}$$

The matrices Q and R are user-defined Hermitian matrices, both positive definite, which respectively dictate the importance of state errors and the inclination towards energy expenditure. In this paper, they are defined as follows:

$$Q = \begin{bmatrix} 0.75 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } R = [0.003] \tag{22}$$

Resulting in a gain matrix:

$$K = [-50 \quad 189.05 \quad -48.80 \quad 30.83] \tag{23}$$

The implementation of the physical control system involves the combination of the two controllers developed earlier. Since the control law developed in Eq. (7) expresses values in terms of acceleration, it should be converted to voltage, just as it is done with the LQR controller. Substituting Eq. (15) into Eq. (12) and equating it to Eq. (7) results in:

$$u_s = \frac{u_{as} - f_x}{g_x} \tag{24}$$

where,

$$f_x = \frac{-(J + ml^2)(ml \sin \alpha)\dot{\alpha}^2 - (J + ml^2)B_{eq}\dot{x} + (ml)^2 g \sin \alpha \cos \alpha}{(M + m)(J + ml^2) - (ml \cos \alpha)^2}$$

$$g_x = \frac{(J + ml^2)A_m}{(M + m)(J + ml^2) - (ml \cos \alpha)^2} \tag{25}$$

Figure 3 outlines the steps executed, considering the initial position of the system at $x_i = [0 \quad \pm\pi \quad 0 \quad 0]$, that means, down and at rest. The system undergoes an initial impulse, then the condition $|\alpha| > 10°$ is checked. While this condition is true, the controller will use the gain u_s. When it becomes false, the controller switches the control signal to u, stabilizing the pendulum in its upper position, and then the control process is terminated. Throughout the process, the voltage sent to the motor is subject to saturation. In other words, if the controller sends a voltage with magnitude greater than u_{sat}, it will be limited to $\pm u_{sat}$.

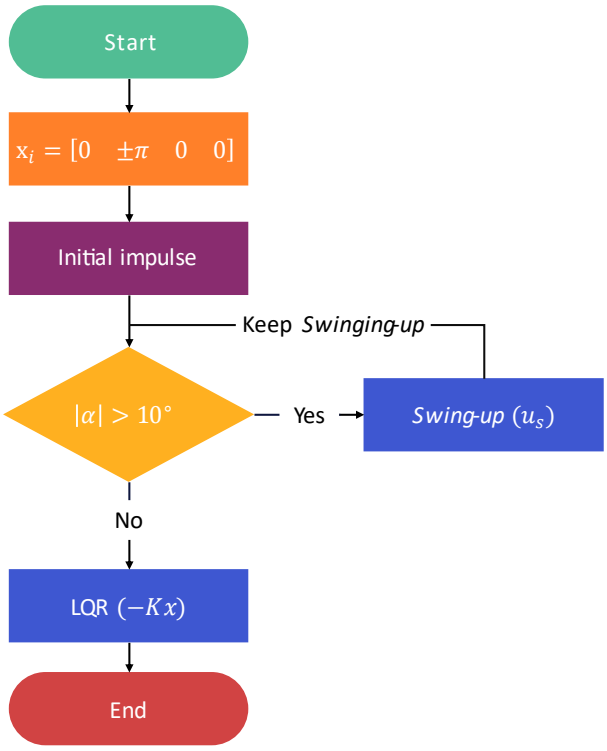

Figure 3. Flowchart of control application based on pendulum angle

2.3 Experimental Implementation

Figure 4 depicts the experimental setup, which includes the LIP and the control hardware, with a focus on the Arduino and the printed circuit board (PCB), locally made, for converting and transmitting digital signals to analog ones. The experimental process is divided into three main stages that repeat successively: signal acquisition of the state variables, computation of the control signal, and sending the control signal to the motor subjected to saturation. The acquisition of state signals is done in two sub-stages. Initially, the Arduino reads the signals sent by two incremental encoders - one attached to the cart measuring its displacement, x, and the other attached to the pendulum measuring its angular displacement, α. The *Arduino* sends this data along with the iteration time duration to a computer, where the second stage is performed. Using the history of states and time, the computer calculates the velocities of the cart, $\dot{x}$, and the pendulum, $\dot{\alpha}$, through numerical differentiation, completing the state x for a given instant.

In the implementation code using the Arduino IDE, the "attachInterrupt" command was used to acquire the signals from the encoders. To regulate the acquisition time, a "while" loop checked whether the main microcontroller loop had already reached the 6.67 millisecond period. If not, it waited using the "yield" command inside the loop. This way, the delay did not affect the interruptions. Next, the control signal is calculated. Initially, the criterion of $|\alpha| > 10°$ is analyzed. If this condition is met, the control signal will be calculated using u_s. Otherwise, it will use u. This control signal is sent back to the *Arduino*, which passes it to a 12-bit digital-to-analog converter (DAC), GY4725. The output of this DAC is a

signal between 0 and 2 V. It is then sent to an operational amplifier (OpAmp), TL071CP, configured as a rail splitter and powered by a symmetric power supply of ±12,V. This setup ensures that a control signal of 0 V corresponds to -2 V, and 2 V corresponds to 2 V. Finally, a second OpAmp, VoltPAQ-X1, triples this value and provides it to the motor with the necessary current, resulting in $u_{sat} = \pm 6$V.

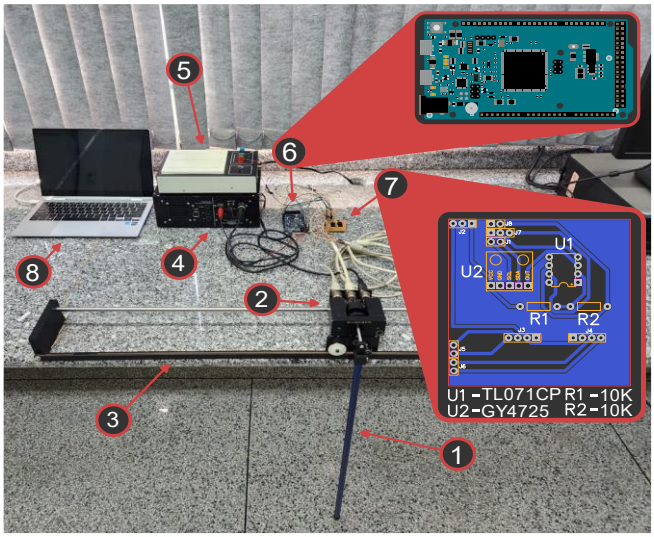

Figure 4. Experimental setup with item indications: 1 - Pendulum; 2 - Cart; 3 - Rail; 4 - Operational Amplifier VoltPAQ-X1; 5 - Symmetrical power supply of ±12 V; 6 - Arduino Due; 7 - Printed circuit board with DAC GY4725 and Operational Amplifier TL071CP; 8 - Computer

The native serial communication of the Arduino was used for the acquisition and transmission of control signals, providing more stable and higher sampling rates compared to the standard serial communication of other microcontrollers. With this implementation, the system maintained an average sampling rate of approximately 150 Hz, with a maximum variation of 0.20 Hz. Figure 5 shows the behavior of this sampling rate during one of the swing-up and stabilization processes, which can be considered stable with negligible variations.

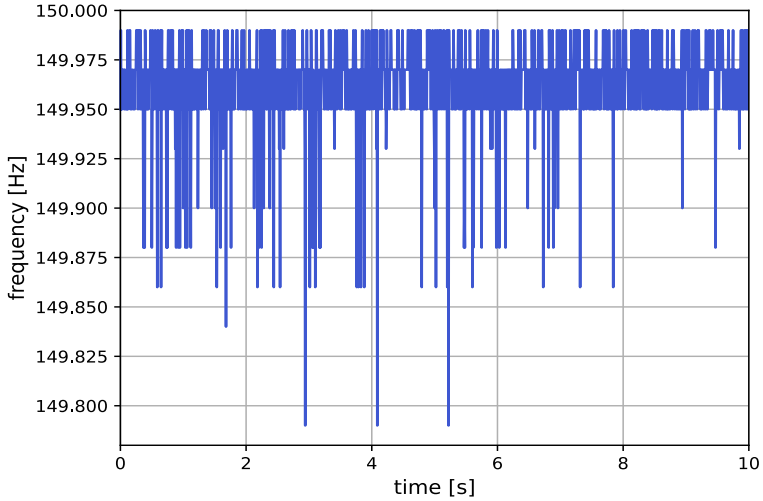

Figure 5. Variation in the sampling rate during a swing-up and stabilization process

3. RESULTS AND DISCUSSION

The system parameter values used for simulation are listed in Table 1. The results obtained experimentally and numerically for each case are shown in Figure 6, with paired graphs for the pendulum's angular position and the cart's displacement. In all cases, $u_a = 1.6$, and λ is varied as $[3.0 \quad 4.5 \quad 6.0]$.

Table 1. Physical parameters of the inverted pendulum

Parameter	Value	Parameter	Value
J	$7.88 \times 10^{-}3$ kg.m^2	M	0.94 kg
m	0.23 kg	B	1.5 N. s/m
l	0.32 m	B_m	4.81 kg. m^2
g	9.81 m/s^2	A_m	1.07 N/V

In the graphs shown in Figure 6, the pendulum gradually increases its oscillation amplitude until it reaches stability at 0 or 2π rad (where $E_p = 0$), depending on the direction of oscillation. Simultaneously, the cart loses its maximum velocity with each oscillation, resulting in smaller displacements in its position, which tends towards the origin. As highlighted by Yang et al. [5], the parameter λ can be adjusted to meet the cart's displacement limit in exchange for a greater number of oscillations. For example, in the experimental cases with $\lambda = 3.0$, the maximum amplitude of the cart's movement was 0.32 m, whereas for $\lambda = 6.0$, it was 0.26 m, with an increase of two oscillations.

In all cases, the simulation required one less oscillation for the swing-up. Initially, this is attributed to two sources of uncertainty: the physical parameters of the system model and the simulation of the initial impulse. In the lab, the initial impulse was manually applied, while in the simulation, it was done by modifying the boundary conditions, adding initial velocity to the pendulum. Specifically, $\dot{\alpha}_{\lambda=3.0,\,4.5} = 3.0$ and $\dot{\alpha}_{\lambda=6.0} = 1.4$ rad/s. It's worth noting that the existence of saturation did not harm the system functionality, in every test the pendulum converged to the upward position. What is possible to abstract is that in a scenario without saturation the convergence might happen faster. Figure 7 shows how the voltage was applied to the motor in every experiment with $u_a = 1.6$, during the swing-up there were instants in which the upper or lower limits are reached. It is also noticeable that the voltage pattern applied for stabilization under the LQR controller shows a repetitive polarity alternation pattern, a necessary effect to keep the pendulum upright.

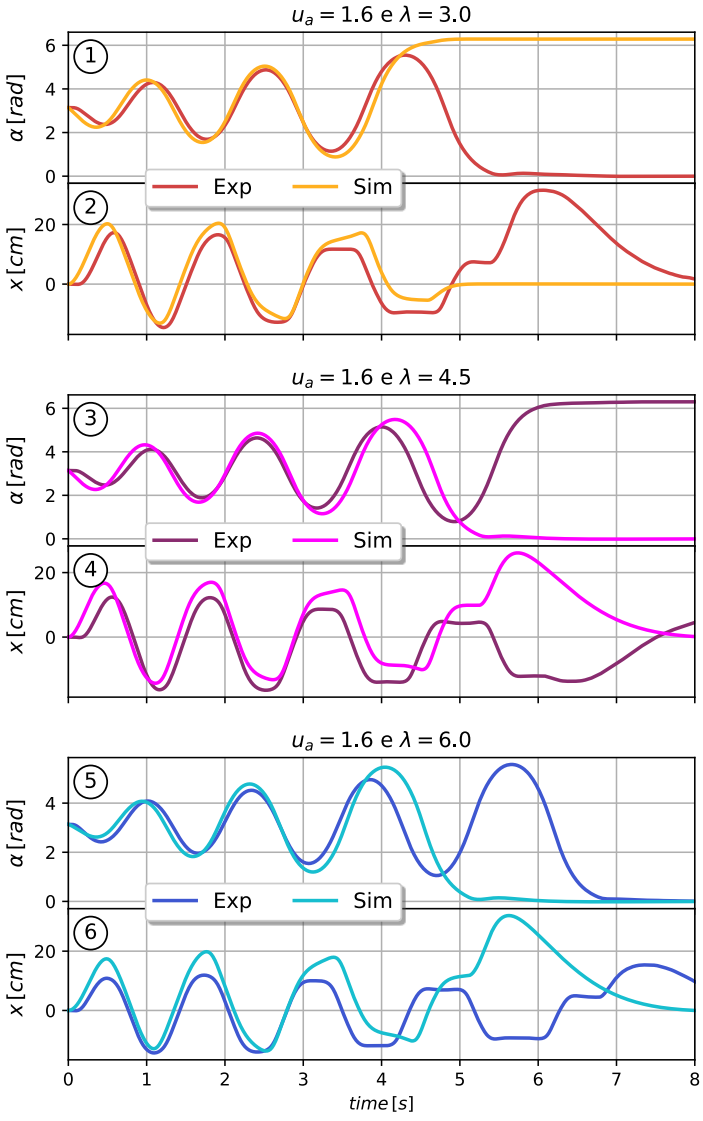

Figure 6. Experimental and simulated results of swing-up and stabilization with $u_a = 1.6$ and varying λ. 1 and 2 - Pendulum and cart position, respectively, for $\lambda = 3.0$; 3 and 4 - Pendulum and cart position, respectively, for $\lambda = 4.5$; 5 and 6 - Pendulum and cart position, respectively, for $\lambda = 6.0$. Exp and Sim stand for Experimental and Simulation, respectively

Figure 8 illustrates how the swing-up increases the maximum energy of the pendulum with each cycle, moving from its initial stationary orbit, where its energy is minimal at $-2mlg$, to reaching the transition orbit with zero energy. Throughout this process, the control law used is u_s; only upon reaching the minimum energy is it switched to u, which corresponds to the LQR controller's control law. The results obtained in this work validate the real-time performance

achievable through the combination of the open-source platforms Python and Arduino. Additionally, the results, both numerically and experimentally, attest to the functionality of the method presented by Yang et al. [5]. Furthermore, the results are qualitatively comparable to those obtained by other authors who implemented similar control rules, such as Bellati et al. [14] and Graham and Turkoglu [15].

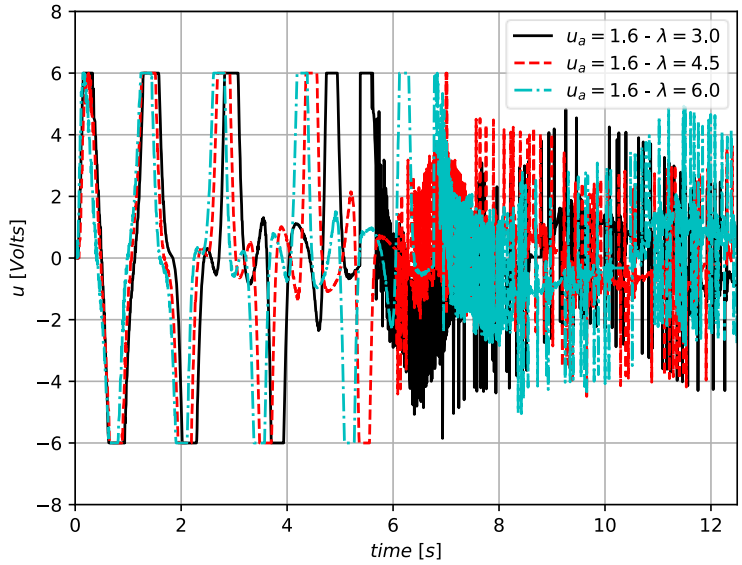

Figure 7. Voltage applied to the motor during experiments with $u_a = 1.6$ and $\lambda = [3.0 \quad 4.5 \quad 6.0]$

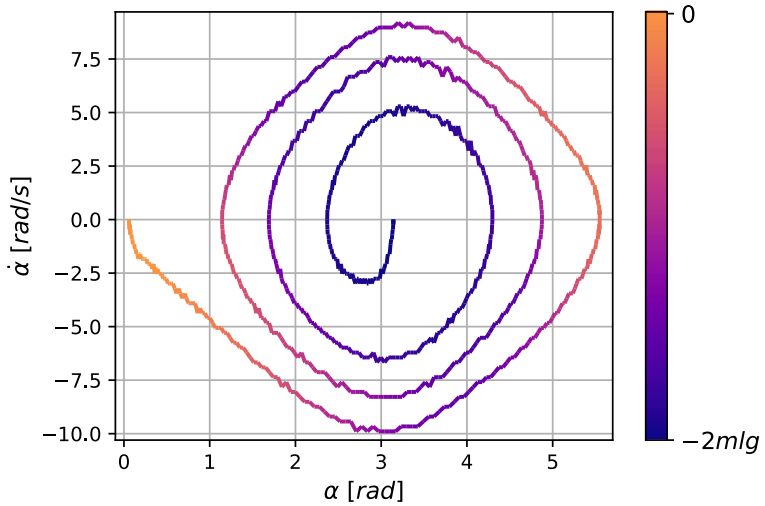

Figure 8. Phase portrait of angular position [α] and angular velocity [$\dot{\alpha}$] with the pendulum's energy gradient for the swing-up result with $u_a = 1.6$ and $\lambda = 3.0$

4. CONCLUSIONS

This work successfully implemented the referenced swing-up method, obtaining equivalent results when compared to the existing literature and demonstrating that, through adjustments of control parameters, the approach is suitable for analogous systems. The experimental results were consistent with simulations, even in the presence of uncertainties in physical parameters and boundary conditions. Furthermore, the application of microcontrollers, such as Arduino, for LIP control with swing-up was validated as an alternative to commercial data acquisition and control boards. The necessary steps and techniques to achieve sampling frequencies with minimal fluctuations were also presented.

ACKNOWLEDGEMENTS

We thank UNESP Ilha Solteira for providing the control laboratory and equipment used to run the tests. This work was funded by CAPES under grant number 88887.678338/2022-00.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

AUTHORS CONTRIBUTION

M. M. Donatoni (Methodology; Data curation; Writing - original draft; Resources)
F. R. Chavarette (Conceptualization; Formal analysis; Visualisation; Supervision)
E. Preto (Methodology; Writing - original draft)
D. C. J. Karmouche (Methodology; Writing - original draft)

REFERENCES

[1] J. J. da Cruz, "Controle automático," *SBA: Controle & Automação Sociedade Brasileira de Automatica*, vol. 22, no. 4, pp. 425–427, 2011.

[2] R. C. Dorf, R. H. Bishop. Modern Control Systems, 13th ed. Boston: Pearson, 2016.

[3] K. Ogata. Modern Control Engineering, 5th ed. Boston: Pearson, 2010.

[4] T. Yoneyama, "Alguns aspectos relevantes em projeto," in *Engenharia de Controle: Teoria e Prática*, pp. 439–464, 2022.

[5] J.-H. Yang, S.-Y. Shim, J.-H. Seo, Y.-S. Lee, "Swing-up control for an inverted pendulum with restricted cart rail length," *International Journal of Control, Automation and Systems*, vol. 7, no. 4, pp. 674–680, 2009.

[6] H. O. Wang, K. Tanaka, M. F. Griffin, "An approach to fuzzy control of nonlinear systems: Stability and design issues," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 1, pp. 14–23, 1996.

[7] N. Muskinja, B. Tovornik, "Swinging up and stabilization of a real inverted pendulum," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 2, pp. 631–639, 2006.

[8] C. W. Tao, J. S. Taur, Tzuen Wuu Hsieh, C. L. Tsai, "Design of a fuzzy controller with fuzzy swing-up and parallel distributed pole assignment schemes for an inverted pendulum and cart system," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1277–1288, 2008.

[9] J. Si, Y-T. Wang, "Online learning control by association and reinforcement," *IEEE Transaction on Neural Networks*, vol. 12, no. 2, pp. 264–276, 2001.

[10] J. Morimoto, K. Doya, "Robust reinforcement learning," *Neural Computation*, vol. 17, no. 2, pp. 335–359, 2005.

[11] M. Riedmiller, "Neural reinforcement learning to swing-up and balance a real pole," in *IEEE International Conference on Systems, Man and Cybernetics*, pp. 3191–3196, 2005.

[12] K. Yoshida, "Swing-up control of an inverted pendulum by energy-based methods," in *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, vol. 6, pp. 4045–4047, 1999.

[13] M. Bugeja, "Non-linear swing-up and stabilizing control of an inverted pendulum system," in *IEEE Region 8 EUROCON2003. Computer as a Tool*, vol. 2, pp. 437–441, 2003.

[14] A. Bellati, N. P. Blengio, F. Cancela, P. Monzon, N. Perez, "Modeling and control of a Furuta pendulum," in *IEEE URUCON*, pp. 334–338, 2021.

[15] K. Turkoglu, B. Graham, "Design, build and integration of a low-cost self-erecting inverted pendulum mechanism," in *AIAA Modeling and Simulation Technologies Conference*, Grapevine, Texas, 9 – 13th January, 2017.

[16] L. B. Prasad, B. Tyagi, H. O. Gupta, "Optimal control of nonlinear inverted pendulum system using PID controller and LQR: Performance analysis without and with disturbance input," *International Journal of Automation and Computing*, vol. 11, no. 6, pp. 661–670, 2014.

[17] J. Lee, R. Mukherjee, H. K. Khalil, "Output feedback stabilization of inverted pendulum on a cart in the presence of uncertainties," *Automatica*, vol. 54, pp. 146–157, 2015.

[18] S. Irfan, A. Mehmood, M. T. Razzaq, J. Iqbal, "Advanced sliding mode control techniques for inverted pendulum: Modelling and simulation," *Engineering Science and Technology, an International Journal*, vol. 21, no. 4, pp. 753–759, 2018.