SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 20, No. 3, October 2023, 361-390

UDC: 681.511.4 **DOI: https://doi.org/10.2298/SJEE2303361T**

A Comparative Study Between Sliding Mode and Fuzzy Sliding Mode Control for Pendubot

Minh-Duc Tran[1], Minh-Tai Vo[1,2], Van-Dong-Hai Nguyen[3]

Abstract: The pendubot is a classical highly non-linear system that has been widely used in many laboratories to demonstrate the responses of the system when applying control algorithms, to analyse dynamical equations, and to study the parameter identification algorithms. The objective of this paper is to design the controller to stabilize the pendubot at equilibrium points (TOP and MID) and to track the defined trajectory. Firstly, mathematical equations of the pendubot are derived by the Euler–Lagrange method. Thence, a genetic algorithm (GA) is employed to identify the parameters of the system based on the collected data of the output states of the system when applying impulse inputs. After that, sliding mode control (SMC) is designed to balance the system at the equilibrium points and track the defined trajectory. The chattering caused by SMC is reduced by fuzzy-sliding mode control (FSMC). The proposed FSMC method solves the problem induced by SMC by applying fuzzy logic. Additionally, the partial feedback linearisation (PFL) method is introduced to design the swing-up system. Finally, both simulation and experimental results are provided to show the effectiveness and robustness of the proposed methods.

Keywords: Pendubot, Identification system, Sliding Mode Control, Fuzzy-Sliding Mode Control, Swing-up.

1 Introduction

The pendulum robot, which is also known as the pendubot, is a classical non-linear system and is widely used in laboratories to demonstrate the quality control algorithm, identification methods, and other studies on its kinetic equations. The mechanical part includes a rotating arm which is the actuated part producing torque for the pendulum to swing, a pendulum controlled by the rotating arm, a DC motor, an H-bridge, and other important components such as encoders, a micro-processing board, the pendubot frame built of steels, and power supply.

[1]Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, 268-Ly Thuong Kiet, Ward 14, District 10, Ho Chi Minh City, Vietnam; E-mail: tmduc.sdh222@hcmut.edu.vn
[2]School of Science, Engineering & Technology, RMIT University Vietnam, 702 Nguyen Van Linh Blvd., Tan Phong ward, District 7, Ho Chi Minh City, Vietnam; E-mail: tai.vo3@rmit.edu.vn
[3] Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Education (HCMUTE), 01-Vo Van Ngan, Thu Duc city, Ho Chi Minh City, Vietnam; E-mail: hainvd@hcmute.edu.vn

The balancing control, trajectory tracking control, and swing-up control for the pendubot system are considered a tricky research topic in control engineering because of their nonlinearity. Many control methods have been proposed to address these topics, which can be viewed in the literature. The pendubot is the research subject to apply motion control strategy based on the integrated trajectory method [1]. In this work a motion control methodology is proposed to accomplish the above control objective with a single controller. In [2], the authors design robust sliding mode controllers for a pendubot with 2 DOF. In [3, 4], the authors use the partial feedback linearisation method to swing up the pendubot from a vertical downward initial position to a vertical upright position. Thence, they use the LQR algorithm to stabilize the system at the desired equilibrium position. The method for the swing-up is also used in [5, 6]. In [7, 8], the authors propose energy-based control of the pendubot and show how they choose the proper coefficient values for the controller. In [9], the same method has been used but in a different system. Besides, in the studies [10, 11], the authors present the SMC design and optimise the controller's coefficients. They also show how robust it is not only in balancing the system at the equilibrium points but also retaining the ability to track the trajectory when the desired point changes in time. In [12], the authors introduce the new method by applying a Takagi–Sugeno fuzzy model to the pendubot system to track the sine wave. Besides, the application of feedforward control has been studied for the pendubot system and other systems [13 – 15]. Moreover, many applications have been applied successfully to the pendubot, such as open-loop control schemes [16, 17], feedback control [5, 18, 19], adaptive control [20] and [26], and so on.

Sliding mode control (SMC) is popular due to its robustness to uncertainties [21]. There are some advantages of SMC such as the robustness of the controller against external disturbances and parameter variations, giving fast dynamic response convergence to the system, and its implementation in a straightforward manner [21]. SMC has been applied successfully to applications in many different majors. Hierarchical sliding mode control has also been proposed and applied successfully in [22]. The combination of SMC and LQR is studied and applied to a rotary double-inverted pendulum [21]. Applications of SMC for under-actuated inverted pendulum [23], an induction motor [24], DC-DC buck converters [25], and so on are also reported.

In general, there are many methods of controlling the pendubot, and the above methods have many advantages, but the cost is quite high. One of them is the chattering phenomenon when applying the sliding mode controller, which causes heat in the DC motor and forcefully oscillates the pendubot at the equilibrium points. This idea is quite similar to that reported in [23], but we identify the system's parameters before applying the controllers and evaluate the two controllers' quality on the trajectory signal by simulation and experiment.

Moreover, in this paper, we also provide a method to identify system parameters that are not presented in [1 – 27], then we provide a wider operation range of the pendubot system when it is not only balanced at the equilibrium points but also follows predefined trajectories, which are lacking in some of the papers listed [3 – 5, 10, 11, 13, 22, 23, 26]. The common swing-up method used for the pendubot system, which is used in the same way as that in [4, 5], is also presented. In [22] and [23], the authors show very good results in comparing FSMC and HSMC in simulation in different systems, so with the same idea, we provide the procedure to adopt the final control law of the SMC and the fuzzy rules for the FSMC, then we perform a comparison of both the simulation results and practical results of the two controllers. This will provide the reader with the procedure to be able to make the pendubot system work in simulation as well as in practice, and these implementation steps can also be employed to other under-actuated systems such as rotary inverted pendulum systems, inverted wedge pendulum systems, and many more.

Weakness of SMC is chattering. The chattering phenomenon can damage the system because it causes heat loss and creates loud noise in most of the electrical power circuits and wear on moving mechanical components [27]. Each contribution in this article is listed as follows : i) to design two controllers, which include SMC and FSMC for a nonlinear pendubot system; ii) to design the FSMC to solve the chattering problem that is caused by SMC and iii) to make a comparison between FSMC and SMC to assess which is the better controller. In addition, we also provide the method that we used: iv) to identify the system's parameters by using a genetic algorithm (GA), and v) to swing up the pendubot system by using the partial feedback linearisation method (PFL).

This paper is organised in 5 sections. Section 1 introduces the model and the key objectives of this work. Section 2 presents the pendubot dynamic model. Section 3 gives the design of pendubot system identification and controllers for the pendubot. The results of the simulation and experiment are shown in Section 4. Lastly, Conclusions of the work are given in Section 5.

2 Mathematical Model of the Pendubot System

The pendubot system is presented in Fig. 1. According to **Error! Reference source not found.**, 4], the mathematical equations of the pendubot are

$$D_{11}\ddot{q}_1 + D_{12}\ddot{q}_2 + C_{11}\dot{q}_1 + C_{12}\dot{q}_2 + G_1 = t_1 - b_1\dot{q}_1, \quad (1)$$

$$D_{21}\ddot{q}_1 + D_{22}\ddot{q}_2 + C_{21}\dot{q}_1 + G_2 = -b_2\dot{q}_2, \quad (2)$$

where

$$D_{11} = \left(\theta_1 + \theta_2 + 2\theta_2\theta_3\right),\ D_{12} = \left(\theta_2 + \cos(q_2)\theta_3\right),$$

$$D_{21} = \left(\theta_2 + \cos(q_2)\theta_3\right),\ D_{22} = \theta_2,$$

$$C_{11} = -\theta_3 \sin(q_2)\dot{q}_2,\ C_{12} = -\left(\dot{q}_1 + \dot{q}_2\right)\sin(q_2)\theta_3,\ C_{21} = \theta_3 \sin(q_2)\dot{q}_1,$$

$$G_1 = \theta_4 g\cos(q_1) + \theta_4 g\cos(q_1 + q_2),\ G_2 = \theta_5 g\cos(q_1 + q_2),$$

$$\theta_1 = m_1 l_{c1}^2 + m_2 l_{c2}^2 + I_{c1},\ \theta_2 = m_2 l_{c2}^2 + I_{c2},\ \theta_3 = m_1 l_1 l_{c2},$$

$$\theta_4 = m_1 l_{c1} + m_2 l_1,\ \theta_5 = m_2 l_{c2},\ \theta_6 = C + k_t^2/R,\ \theta_7 = k_t/R.$$

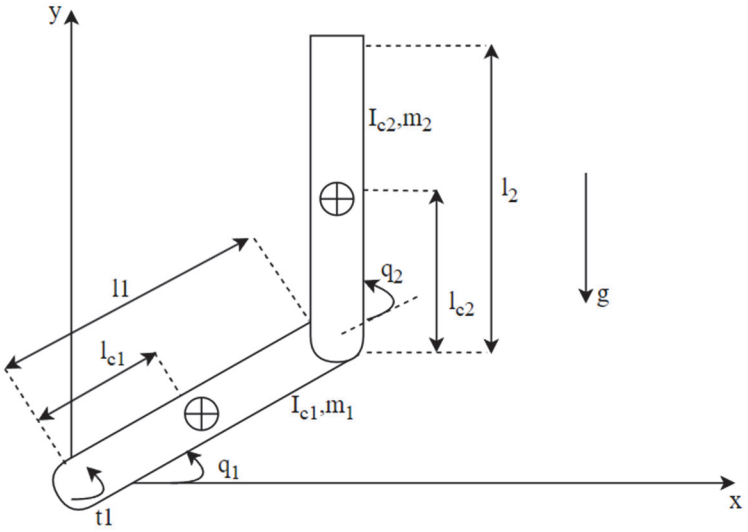


Fig. 1 – *Pendubot system in x-y coordinates.*

The system parameters are shown in **Table 1**.

Since the actual input signal for the pendubot system is the voltage applied to the DC motor, the transformation of the first mathematical equation represented in (1) of the system to the new one is presented by the formula below:

$$v = \frac{\tau_1 + J_m \ddot{q}_1 + \theta_6 \dot{q}_1}{\theta_7}. \tag{3}$$

By substituting (3) into (1), the new mathematical equations of the pendubot system using v as input are as follows:

$$(D_{11} + J_m)\ddot{q}_1 + D_{12}\ddot{q}_2 + (C_{11} + \theta_6 + b_1)\dot{q}_1 + C_{12}\dot{q}_2 + G_1 = \theta_7 v, \tag{4}$$

$$D_{21}\ddot{q}_1 + D_{22}\ddot{q}_2 + C_{21}\dot{q}_1 + b_2\dot{q}_2 + G_2 = 0. \tag{5}$$

Table 1
Parameters of the pendubot.

No	Parameter	Description	Unit
1	m_1	Mass of rotating arm	kg
2	l_1	Length of rotating arm	m
3	l_{c1}	Centre of mass length of rotating arm	m
4	m_2	Mass of pendulum	kg
5	l_2	Length of pendulum	m
6	l_{c2}	Centre of mass length of pendulum	m
7	g	Gravitation acceleration	m/s^2
8	b_1	Friction of rotating arm	1/s
9	b_2	Friction of pendulum	1/s
11	I_{c1}	Inertia of rotating arm	$kg\ m^2$
12	I_{c2}	Inertia of pendulum	$kg\ m^2$
13	R	Motor armature coil resistance	Ω
14	k_t	Torque constant	Nm/A
15	C	Friction	1/s
16	J_m	Motor inertia	$kg\ m^2$
17	v	DC motor voltage	V
18	q_1	Angle of rotating arm	rad
19	q_2	Angle of pendulum	rad
20	$\dot{q}_1$	Angular velocity of rotating arm	rad/s
21	$\dot{q}_2$	Angular velocity of pendulum	rad/s
22	$\ddot{q}_1$	Angular acceleration of rotating arm	rad/s^2
23	$\ddot{q}_2$	Angular acceleration of pendulum	rad/s^2

3 Parameters Identification and Controller Design

3.1 Implementing GA to identify system parameters

Five steps are used to conduct parameter identification by GA:

S1: Determining parameters that need to be optimized. Then, randomly generating a population consists of individuals based on experience or non-experience (e.g. a set includes parameters that need to be optimized).

S2: Defining optimal function J to estimate individuals whether individuals satisfy termination criteria or not.

S3: Choosing the best individuals and deleting the bad individuals.

S4: Now, crossover operation is involved recombining genetic material between two parents individuals to generate new offspring. Then, rely on probability to create mutations for individuals to ensure the search can be spread out without getting stuck at local point.

S5: After completing the loop "Selection – Crossover – Mutation" operations or satisfying termination criteria, we list down the best individuals and finish program.

3.2 Pseudo-code of GA

1. Generate sets of initial values:

$$\vec{x}_{l,G} = \left[J_m, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, b_1, b_2\right].$$

2. Determine an objective function $J=\|e\|$. such that e is the error of output signals compared with the simulation ones

$$\boldsymbol{e} = \left[(q_{1,simu} - q_{1,real}) \quad (\dot{q}_{1,simu} - \dot{q}_{1,real}) \quad (q_{2,simu} - q_{2,real}) \quad (\dot{q}_{2,simu} - \dot{q}_{2,real})\right]^{\mathrm{T}}$$

3. For $G = 1$ to *Max_Gene.*
4. Sorting the objective function values of the sets from bad to good.
5. Implementing the linear ranking selection operation.
6. Implementing crossover BLX–α operation.
7. Implementing the non-uniform mutation operation.
8. Calculate the cost function $(x_i, t) = |e|$.
9. If $J(x_i, G, best) \leq \epsilon$ **,** then
10. *BREAK*;
11. Else
12. *Max_Gen* += 1;
13. End if.
14. End for.

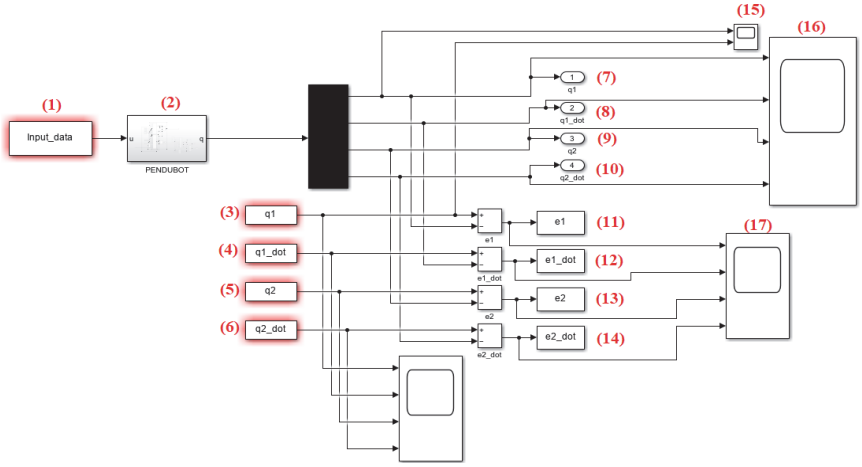


Fig. 2 – *The simulation program generates sample data for the pendubot.*

Fig. 2 shows the simulation program which generates a sample data set for identifying parameters $J_m, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7$.

Table 3
The explanation for Fig. 2.

Block	Description
1	Applied voltage for the pendubot system.
2	Pendubot simulation achieved from (1) and (2)
3,4,5,6	Output states of real pendubot.
7,8,9,10	Output states of simulation pendubot.
11,12,13,14	Errors in the comparison of each output state.
15,16,17	Scope in MATLAB/Simulink monitors each output state between simulation and real pendubot.

3.3 Design of swing-up controller

3.3.1 *Design of swing-up controller at TOP* $q_1 = \pi/2;\ q_2 = 0$.

From (1) we derive

$$\ddot{q}_2 = \frac{-\left(D_{21}\ddot{q}_1 + C_{21}\dot{q}_1 + G_2 + b_2\dot{q}_2\right)}{D_{22}}. \tag{6}$$

Substituting (6) into (2) yields

$$\bar{D}_{11}\ddot{q}_{11} + \bar{C}_{11}\dot{q}_1 + \bar{C}_{12}\dot{q}_2 + \bar{G}_1 = \tau_1, \tag{7}$$

where

$$\bar{D}_{11} = D_{11} - \frac{D_{12}D_{21}}{D_{22}}, \quad \bar{C}_{11} = C_{11} + b_1 - \frac{D_{12}C_{21}}{D_{22}},$$

$$\bar{C}_{12} = C_{12} - \frac{D_{12}b_2}{D_{22}}, \quad \bar{G}_1 = G_1 - \frac{D_{12}G_2}{D_{22}}.$$

By applying torque from (7) to the system, we obtain

$$\ddot{q}_1 = v_1, \ \ D_{21}v_1 + D_{22}\ddot{q}_2 + C_{21}\dot{q}_1 + G_2 = -b_2\dot{q}_2. \tag{8}$$

The desired trajectory outer loop for the rotary arm with angle $q_1 q_1$ is designed as follows:

$$v_1 = \ddot{q}_{1d} + K_V(q_{1d} - q_1) + K_D(\dot{q}_{1d} - \dot{q}_1). \tag{9}$$

Substituting (8) into (7), we have the torque equation for swing-up of the pendubot as

$$\tau_1 = \bar{D}_{11}\left(\ddot{q}_{1d} + K_P\left(q_{1d} - q_1\right) + K_D\left(\dot{q}_{1d} - \dot{q}_1\right)\right) + \bar{C}_{11}\dot{q}_1 + \bar{C}_{12}\dot{q}_2 + \bar{G}_1. \tag{10}$$

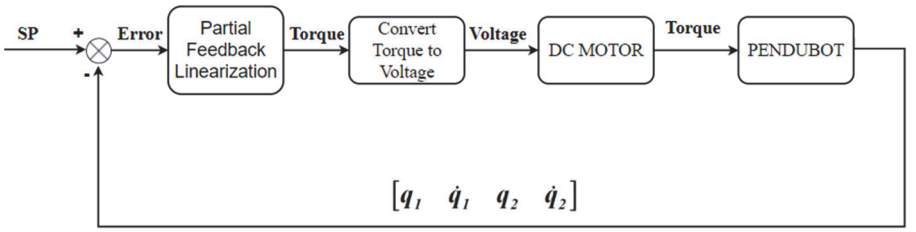


Fig. 3 – *The swing-up controller diagram.*

3.3.2 *Design of swing-up controller at MID* $q_1 = -\pi/2; q_2 = \pi$.

The design of the swing-up controller for the MID position is the same as the TOP position. However, the equilibrium point is different in this case.

3.4 Design of SMC controller and FSMC

3.4.1 Sliding mode control (SMC)

According to [8, 10], we transform (4) and (5) into the form as described in (11). Hence, we have new equations as given below:

$$\begin{aligned} \ddot{q}_1 &= \mathrm{F}_1(q,\dot{q}) + \mathrm{B}_1(q,\dot{q})u \\ \ddot{q}_2 &= \mathrm{F}_2(q,\dot{q}) + \mathrm{B}_2(q,\dot{q})u. \end{aligned} \tag{11}$$

We define the sliding surfaces as

$$S_1 = \mathrm{d}_1 e_1 + \mathrm{d}_2 e_2, \quad S_2 = \mathrm{d}_3 e_3 + \mathrm{d}_4 e_4, \quad S_3 = S_1 + S_2, \tag{12}$$

where:

$$e_1 = q_1 - q_{1d}, \quad e_2 = \dot{q}_1, \quad e_3 = q_2 - q_{2d}, \quad e_4 = \dot{q}_2, \quad u = v$$

and d_i are parameters of the controller, $i = 1,2,3,4$.

The Lyapunov function is chosen as follows:

$$V = \frac{S_3^2}{2} \tag{13}$$

and its derivative is

$$\begin{aligned} &\dot{V} = S_3\dot{S}_3 \quad \Leftrightarrow \quad \dot{V} = S_3(\dot{S}_1 + \dot{S}_2) \quad \Leftrightarrow \\ &\dot{V} = S_3(\mathrm{d}_1\dot{q}_1 + \mathrm{d}_2\ddot{q}_1 + \mathrm{d}_3\dot{q}_2 + \mathrm{d}_4\ddot{q}_2) \quad \Leftrightarrow \\ &\dot{V} = S_3(\mathrm{d}_1\dot{q}_1 + \mathrm{d}_2\mathrm{F}_1(q,\dot{q}) + \mathrm{d}_2\mathrm{B}_1(q,\dot{q})u + \mathrm{d}_3\dot{q}_2 + \mathrm{d}_4\mathrm{F}_2(q,\dot{q}) + \mathrm{d}_4\mathrm{B}_2(q,\dot{q})u). \end{aligned} \tag{14}$$

In order to ensure that $\dot{V} < 0$, we need $\dot{S}_3 = -\mathrm{k}S_3 - \eta\,\mathrm{sign}(S_3)$, so that we have the control law as follows:

$$\begin{aligned} &\mathrm{d}_1\dot{q}_1 + \mathrm{d}_2\mathrm{F}_1(q,\dot{q}) + \mathrm{d}_2\mathrm{B}_1(q,\dot{q})u + \mathrm{d}_3\dot{q}_2 + \mathrm{d}_4\mathrm{F}_2(q,\dot{q}) + \mathrm{d}_4\mathrm{B}_2(q,\dot{q})u = \\ &\qquad = -\mathrm{k}S_3 - \eta\mathrm{sign}(S_3) \quad \Leftrightarrow \\ &u = \frac{-\mathrm{d}_4\mathrm{F}_2(q,\dot{q}) - \mathrm{d}_3\dot{q}_2 - \mathrm{d}_2\mathrm{F}_1(q,\dot{q}) - \mathrm{d}_1\dot{q}_1}{\mathrm{d}_2\mathrm{B}_1(q,\dot{q}) + \mathrm{d}_4\mathrm{B}_2(q,\dot{q})} + \frac{-\mathrm{k}S_3 - \eta\mathrm{sign}(S_3)}{\mathrm{d}_2\mathrm{B}_1(q,\dot{q}) + \mathrm{d}_4\mathrm{B}_2(q,\dot{q})}. \end{aligned} \tag{15}$$

The control input for swinging up the system is separated as follows:

$$u = u_{eq3} + u_{sw3}, \tag{16}$$

where

$$u_{sw3} = \frac{-\mathrm{k}S_3 - \eta\mathrm{sign}(S_3)}{\mathrm{d}_2\mathrm{B}_1(q,\dot{q}) + \mathrm{d}_4\mathrm{B}_2(q,\dot{q})}, \tag{17}$$

$$u_{eq3} = \frac{-\mathrm{d}_4\mathrm{F}_2(q,\dot{q}) - \mathrm{d}_3\dot{q}_2 - \mathrm{d}_2\mathrm{F}_1(q,\dot{q}) - \mathrm{d}_1\dot{q}_1}{\mathrm{d}_2\mathrm{B}_1(q,\dot{q}) + \mathrm{d}_4\mathrm{B}_2(q,\dot{q})}. \tag{18}$$

We expand the denominator $\mathrm{d}_2\mathrm{B}_1(q,\dot{q}) + \mathrm{d}_4\mathrm{B}_2(q,\dot{q})$ to find the condition that ensures it must not reach zero.

$$\begin{aligned} &\mathrm{d}_2\mathrm{B}_1(q,\dot{q}) + \mathrm{d}_4\mathrm{B}_2(q,\dot{q}) \neq 0 \quad \Leftrightarrow \\ &\theta_7\left(\mathrm{d}_4\theta_2 - \mathrm{d}_2\theta_2 + \mathrm{d}_4\theta_3\cos(q_2)\right) \neq 0 \quad \Leftrightarrow \\ &\frac{\mathrm{d}_2\theta_2\theta_7 - \mathrm{d}_4\theta_2\theta_7}{\mathrm{d}_4\theta_3\theta_7} \neq \cos(q_2) \quad \Leftrightarrow \\ &\frac{\mathrm{d}_2\theta_2\theta_7 - \mathrm{d}_4\theta_2\theta_7}{\mathrm{d}_4\theta_3\theta_7} \notin [-1,1]. \end{aligned} \tag{19}$$

Note that $\mathrm{d}_2, \mathrm{d}_4$ are the parameters of the SMC controller and $\theta_7, \theta_2, \theta_3$ are the system constants.

3.4.2 Stability analysis

Based on previous works [10, 22, 23, 27], two theorems will be presented to show the stability of the system and the conditions of the system's parameters.

Theorem 1. *Consider the class of under-actuated systems* (11)*. If the control law adopted is* (16) *and the final layer sliding surface* S_3 *is defined as* (12)*, then* S_3 *is asymptotically stable.*

Proof. Consider the derivative of the defined Lyapunov function (14), then the control law (16) is applied to the system (11):

$$\dot{V} = S_3\dot{S}_3 \quad \Leftrightarrow \quad \dot{V} = S_3\left(-\mathrm{k}S_3 - \eta\mathrm{sign}(S_3)\right) \quad \Leftrightarrow \quad \dot{V} = -\mathrm{k}S_3^2 - \eta|S_3|. \tag{20}$$

Taking the integration of (19) yields

$$\int_0^t \dot{V}\mathrm{d}t = \int_0^t \left(-\mathrm{k}S_3^2 - \eta\left|S_3\right|\right)\mathrm{d}t \quad \Leftrightarrow \quad V(t) - V(0),$$
$$V(0) = V(t) + \int_0^t \left(\mathrm{k}S_3^2 + \eta\left|S_3\right|\right)\mathrm{d}t. \tag{21}$$

Hence,

$$V(0) = V(t) + \int_0^t \left(\mathrm{k}S_3^2 + \eta\left|S_3\right|\right)\mathrm{d}t \geq \int_0^t \left(\mathrm{k}S_3^2 + \eta\left|S_3\right|\right)\mathrm{d}t \quad \Rightarrow$$
$$\lim_{t\to\infty} \int_0^t \left(\mathrm{k}S_3^2 + \eta\left|S_3\right|\right)\mathrm{d}t \leq V(0) < \infty. \tag{22}$$

According to Barbalat's lemma,

$$\lim_{t\to\infty} \left(\mathrm{k}S_3^2 + \eta\left|S_3\right|\right) = 0\,. \tag{23}$$

From (22), we achieve the convergence of the total sliding surface S_3 as follows:

$$\lim_{t\to\infty} S_3 = 0\,. \tag{24}$$

Theorem 2. *Consider the class of under-actuated systems* (11)*. If the control law adopted is* (16) *and the subsystem sliding surfaces of* S_3*, which are* S_1 *and* S_2*, are defined in* (12)*, then they are asymptotically stable.*

Proof. When the control law (16) is applied to the system (11), we have

$$\dot{V} = S_3\dot{S}_3 < \infty\,. \tag{25}$$

As the result,

$$S_3, \dot{S}_3 < L^\infty\,, \tag{26}$$

where L^∞ denotes the space of all bounded functions.

Since S_3 is a linear combination of S_1 and S_2, we also have that $\dot{S}_3$ is a linear combination of $\dot{S}_1$ and $\dot{S}_2$, which yields

$$S_1, S_2 < L^\infty\,, \dot{S}_1, \dot{S}_2 < L^\infty\,. \tag{27}$$

Thus, we have

$$\sup_{t\geq 0}\left|S_i\right| = \left\|S_i\right\|_\infty < \infty\,, \tag{28}$$

where $i = 1, 2$.

Finally, the convergence of both sliding surfaces is

$$\lim_{t\to\infty} S_1 = \lim_{t\to\infty} S_2 = 0\,. \tag{29}$$

From (29), we define another Lyapunov function to determine the condition of the system's parameters:

$$V_{\mathrm{i}} = \frac{1}{2}(q_i - q_{id})^2, \tag{30}$$

where $i = 1,2$ and q_{id} is the desired set point of each state.

Differentiating (30) with respect to time (t), we have

$$\dot{V}_{\mathrm{i}} = (q_i - q_{id})\dot{q}_i. \tag{31}$$

In order to make $q_i \to q_{id}$ and $\dot{q}_i \to 0$, $\dot{V}_i$ must be less than zero and from (29), we have both $S_{1,2} \to 0$, which yields

$$\begin{cases} S_1 = \mathrm{d}_1 e_1 + \mathrm{d}_2 e_2 = 0 \\ S_2 = \mathrm{d}_3 e_3 + \mathrm{d}_4 e_4 = 0 \end{cases} \Leftrightarrow \begin{cases} \mathrm{d}_1(q_1 - q_{1d}) + \mathrm{d}_2 \dot{q}_1 = 0 \\ \mathrm{d}_3(q_2 - q_{2d}) + \mathrm{d}_4 \dot{q}_2 = 0 \end{cases} \Leftrightarrow$$

$$\Leftrightarrow \begin{cases} q_1 - q_{1d} = -\dfrac{\mathrm{d}_2}{\mathrm{d}_1}\dot{q}_1, \\ q_2 - q_{2d} = -\dfrac{\mathrm{d}_4}{\mathrm{d}_3}\dot{q}_2. \end{cases} \tag{32}$$

As results, the SMC's parameters d_j (where $j = 1,2,3,4$) must be positive so that each state of the system is able to converge to its set point or zero.

3.4.3 Fuzzy-sliding mode control (FSMC)

When the control law (16) is applied to the pendubot system, the "chattering" phenomenon is produced in the DC motor. This phenomenon is caused by the switch control part of the control law (16), which is mentioned in (17). Intuitively, the two coefficients are the main components contributing to this phenomenon. By trial and error adjustment of each coefficient in (17), which are k and η to verify, the most effective coefficient on the intensity of the "chattering" phenomenon is η. In the scope of this paper, the authors propose simple but efficient fuzzy logic rules based on the empirical experiment of the practical pendubot system.

The fuzzy rules are designed to eliminate chattering from the system. Fig. 4 depicts the inputs of the fuzzy rules and Fig. 5 shows their outputs. **Table 4** shows the fuzzy control rules.

The input membership function is defined as $|S_3|$ and the output membership function is defined as η.

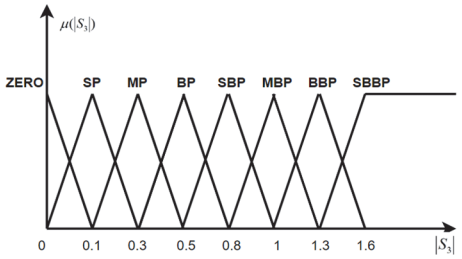


Fig. 4 – *The input rules.*

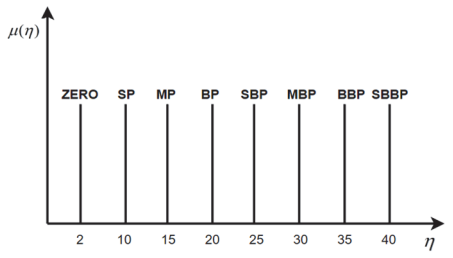


Fig. 5 – *The output rules.*

Table 4

Fuzzy rules for pendubot.

Rule	$\|S_3\|$	η
1	ZERO	ZERO
2	SP	SP
3	MP	MP
4	BP	BP
5	SBP	SBP
6	MBP	MBP
7	BBP	BBP
8	SBBP	SBBP

The notation means: SP (small positive), MP (medium positive), BP (big positive), SBP (small big positive), MBP (medium big positive), BBP (big big positive), SBBP (small big big positive).

The fuzzy rules of the SMC controller are as follows:

– Rule 1: If $|S_3|$ is ZERO, then η is ZERO;

– Rule 2: If $|S_3|$ is SP, then η is SP;

– Rule 3: If $|S_3|$ is MP, then η is MP;

– Rule 4: If $|S_3|$ is BP, then η is BP;

– Rule 5: If $|S_3|$ is SBP, then η is SBP;

– Rule 6: If $|S_3|$ is MBP, then η is MBP;

– Rule 7: If $|S_3|$ is BBP, then η is BBP;

– Rule 8: If $|S_3|$ is SBBP, then η is SBBP.

The fuzzy operators used to calculate the final output η with the input $|S_3|$ are as follows:

– And method: PRODUCT;

– Or method: PROBABILISTIC OR;

– Implication: MIN;

– Aggregation: MAX;

– Defuzzification method: weighted average method.

The Matlab built-in fuzzy logic block in Fig. 6 is used to increase the process of testing the fuzzy rules based on the user's empirical data. Moreover, the fuzzy logic block simulation program combined with SMC in Matlab Simulink is shown in Fig. 7.

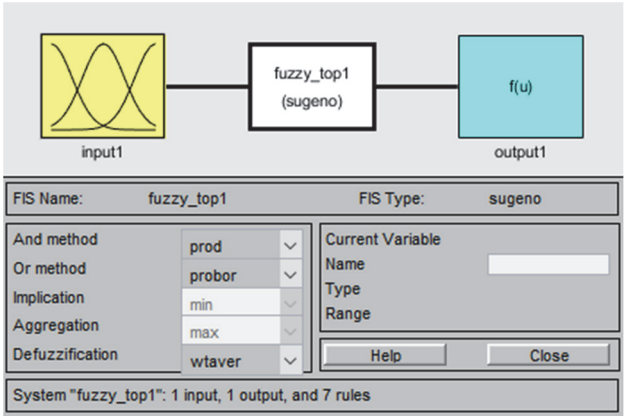


Fig. 6 – *Matlab built-in fuzzy logic block.*

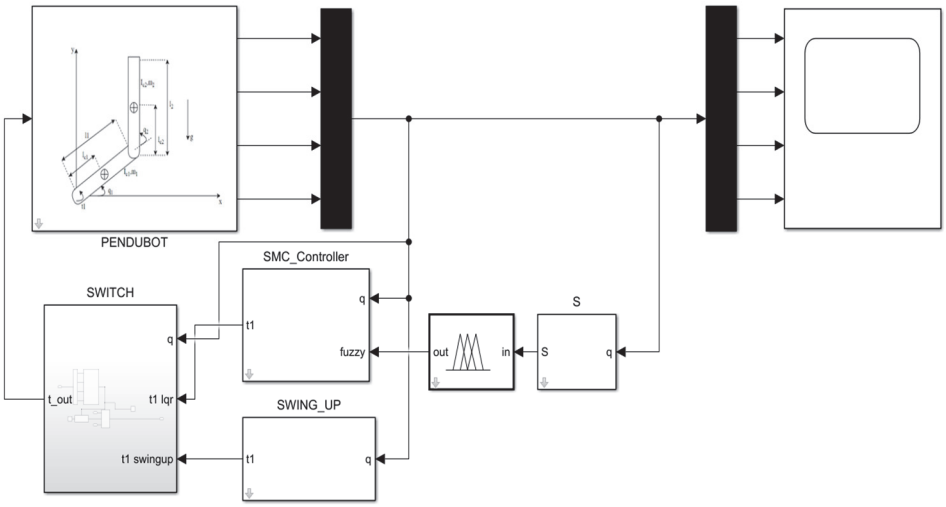


Fig. 7 – *Simulation of FSMC for the pendubot.*

4 Simulation and Experiment Results

4.1 Parameters identification using GA

We initialise the parameters for GA to train the initialisation dataset:

– Population: $N = 50$;

– Linear ranking selection parameter: $\eta_{GA} = 0.5$;

– BLX-α crossover: $\alpha = 0.5$;

– Mutation parameter: $\beta = 0.5$;

– The number of trainings: $\text{Max\_gen} = 30$;

– Crossover probability: $P_c = 0.5$;

– Mutation probability: $P_m = 0.5$.

After 10000 training iterations, we obtain a new dataset; $J = f - f_1$ is the error between the actual data and simulation data for the first training; $J_1 = f - f_2$ for the second, and $J_2 = f - f_3$ for the last training.

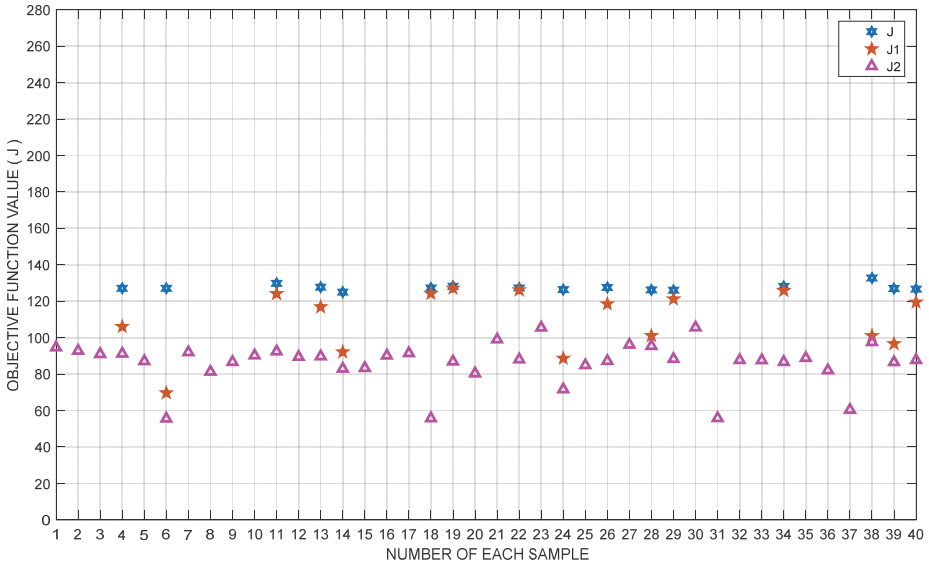


Fig. 8 – *The result after training.*

In Fig. 8, the training results are shown for each training J, J_1, J_2, respectively. The fitness selected dataset of the three-training (in **Tables 5a** and **5b**) gives the

simulation results (red line in Fig. 9) compared to real-time collected data (blue line in Fig. 9).

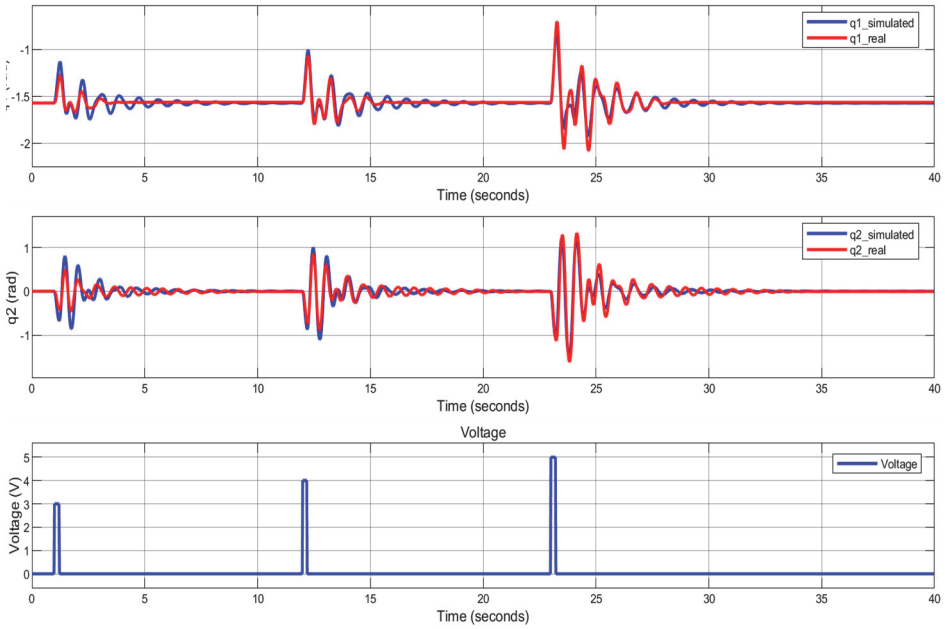


Fig. 9 – *The fitting data of the simulation pendubot using the identified parameters compared to the real data.*

Table 5a
The real parameters.

Initial parameters	Value
J_m	1.52E-05
θ_1	0.003933
θ_2	0.006974
θ_3	0.000698
θ_4	0.004832
θ_5	0.006784
θ_6	0.007788
θ_7	0.005811

Table 5b
The objective value.

Objects	Value
J	127.1241
J_1	69.61168
J_2	55.54988

4.2 Implementation of swing-up and balancing control

The parameters for the controllers are as follows:

$$\text{TOP}: \begin{cases} d_1 = 84.5310, d_2 = 5.8689; \\ d_3 = 62.9853, d_4 = 3.5502; \\ k = 5, \eta = 40; \end{cases}$$
$$\text{MID}: \begin{cases} d_1 = 81.0048, d_2 = 3.2886; \\ d_3 = 184.5449, d_4 = 10.8109; \\ k = 25, \eta = 5. \end{cases} \quad (33)$$

The fuzzy rules in Figs. 4 and 5 are designed based on the range in which the DC motor can generate enough torque at their specific voltage.

4.3 Simulation results

4.3.1 TOP position

Both output states of the pendubot when applying the swing-up controller combined with the SMC and FSMC at the TOP equilibrium point are mostly the same in the simulation (Fig. 10), but the input states are significantly different. We implement fuzzy logic for SMC to reduce the chattering phenomenon, which causes heating of the mechanical actuator parts of the real pendubot.

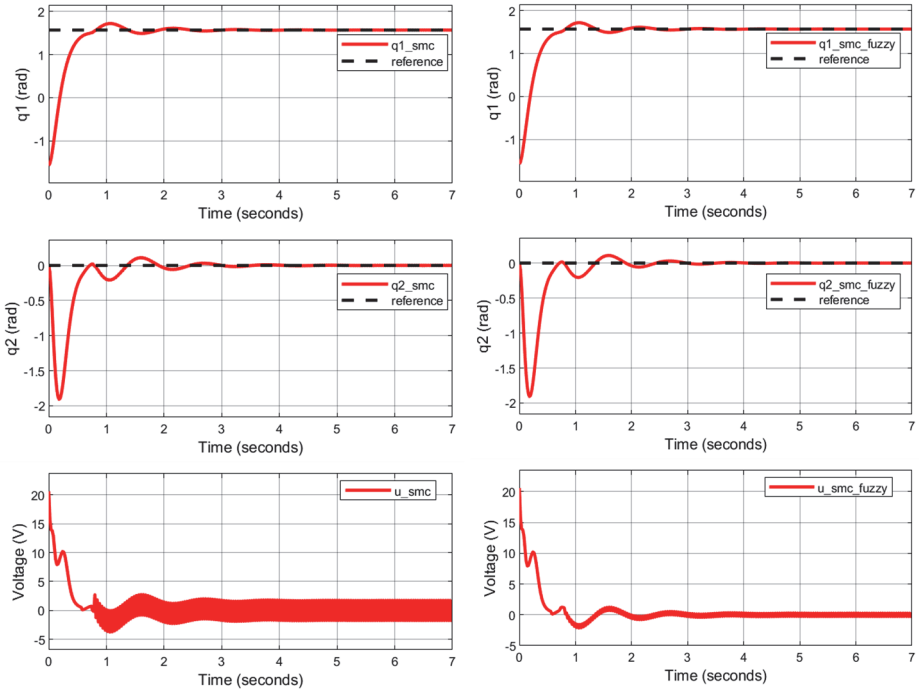


Fig. 10 – *Output and input states under swing-up controller and SMC (left) and FSMC (right) controllers to move pendubot to TOP position and keeping balanced.*

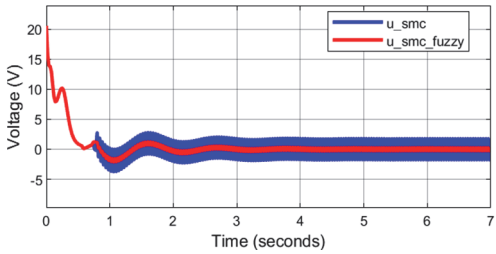


Fig. 11 – *Input states under swing-up controller and SMC (blue) and with FSMC (red) controllers to move pendubot to TOP position and keeping balanced.*

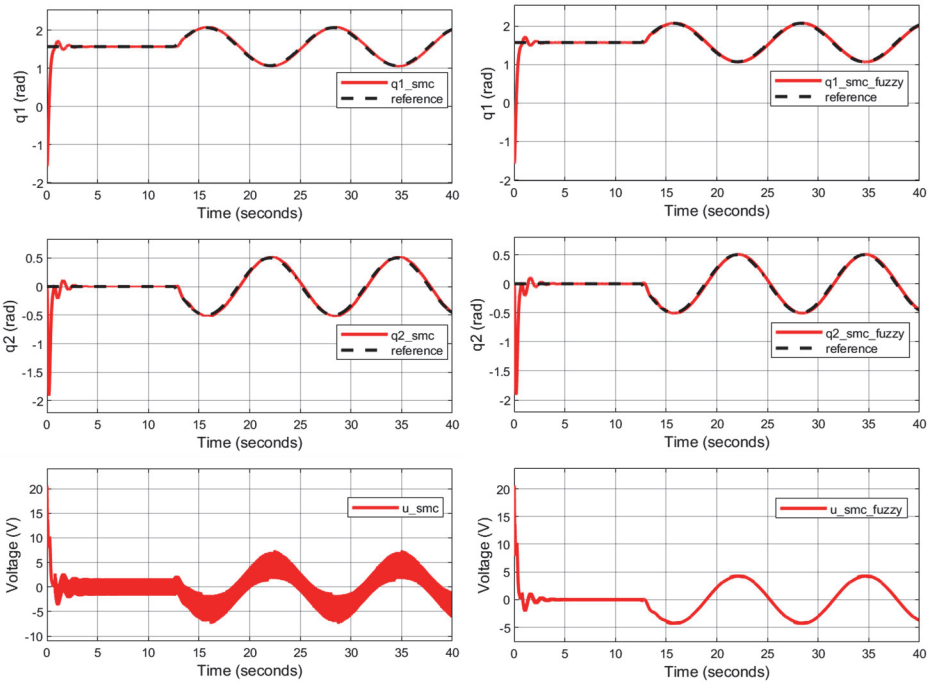


Fig. 12 – *Output and input states under swing-up controller and SMC (left) and FSMC (right) controllers to move pendubot to TOP position and tracking trajectory.*

In Fig. 12, after 12 s of balancing at the TOP position, the pendubot tracks the defined trajectory as follows to show its robustness:

$$\begin{cases} q_1 = \pi/2 + 0.5\sin(0.5t), \\ \dot{q}_1 = 0, \\ q_2 = -0.5\sin(0.5t), \\ \dot{q}_2 = 0, \end{cases} \quad \text{when} \quad (t > 4\pi). \tag{34}$$

Both SMC and FSMC are able to drive the system to track the defined trajectories. The control signal of FSMC shows the effectiveness of its main

purpose. In contrast to FSMC, the control signal of SMC is larger, and it causes the chattering signal.

In Fig. 13, the signals of two controllers are shown, which are the control signal of the SMC controller and the control signal of the FSMC controller, respectively. It can be seen that the control signal value of the FSMC controller is much smaller than the control input of the SMC when the system is in the balanced position, and tracking the sine wave trajectory.

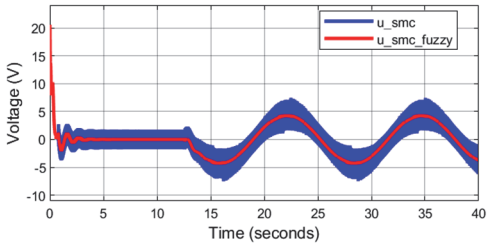


Fig. 13 – *Input states under swing-up controller and SMC (blue) and with FSMC (red) controllers to move pendubot to TOP position and tracking trajectory.*

4.3.2 MID position

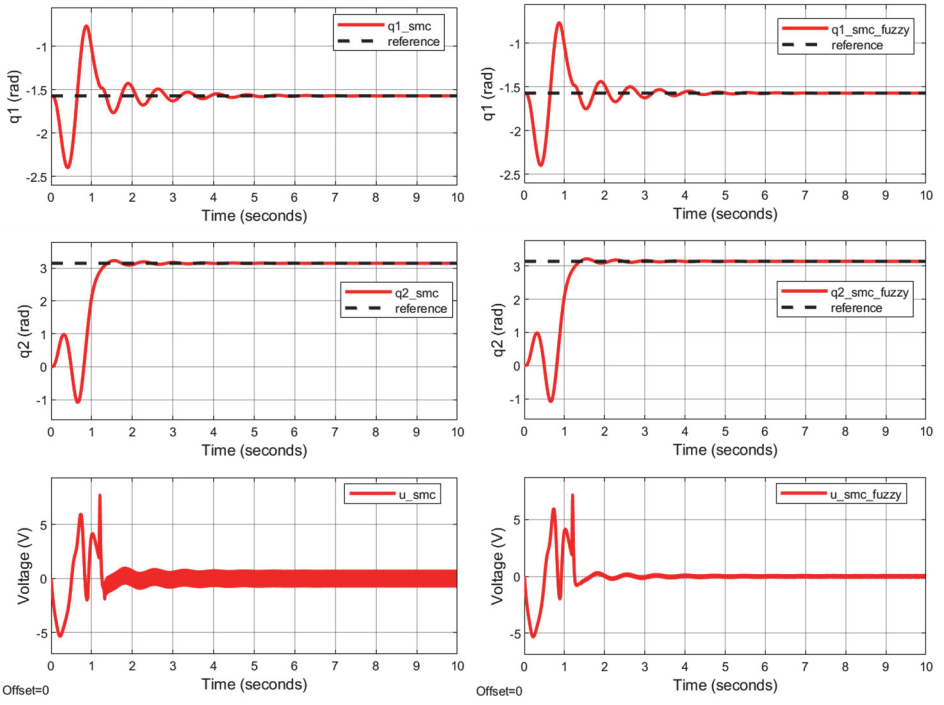


Fig. 14 – *Output and input states under swing-up controller and SMC (left) and FSMC (right) controllers to move pendubot to MID position and keeping balanced.*

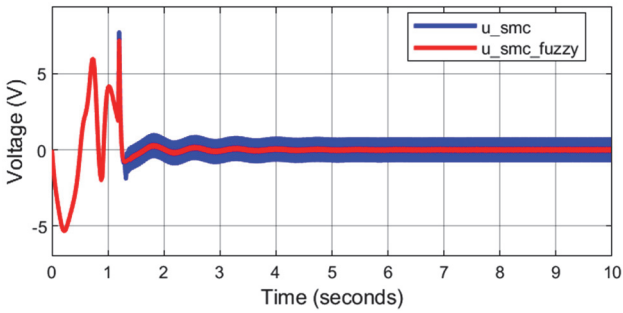


Fig. 15 – *Input states under swing-up controller and SMC (blue) and with FSMC (red) controllers to move pendubot to TOP position and keeping balanced.*

The outputs of the pendubot are primarily the same in Fig. 14. The chattering induced in the input of SMC is reduced when applying the fuzzy mode. The result is shown clearly in Fig. 15.

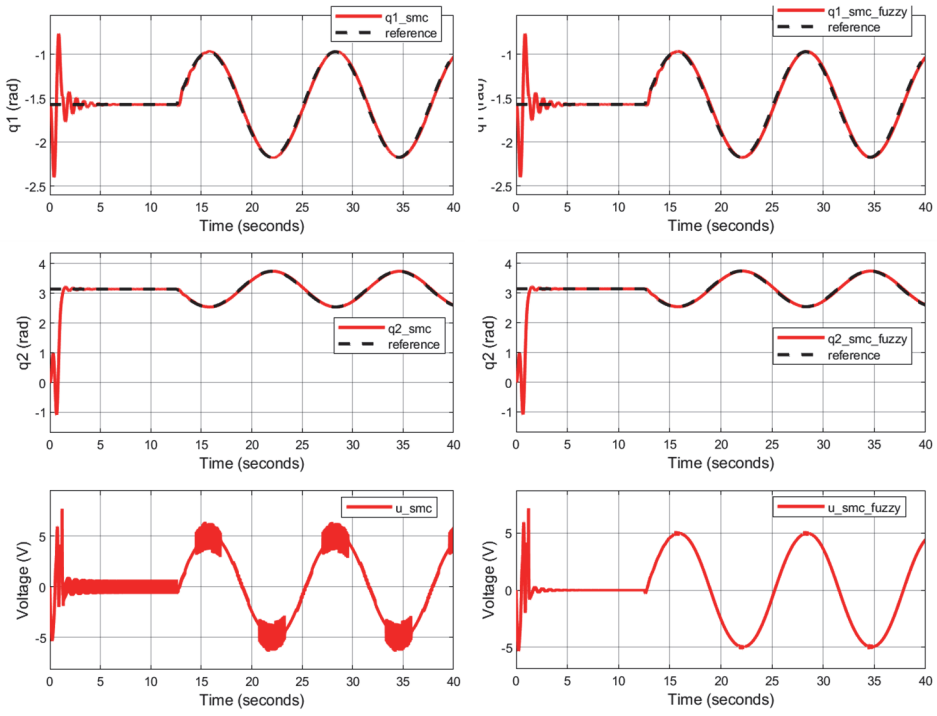


Fig. 16 – *Output and input states under swing-up controller and SMC (left) and FSMC (right) controllers to move pendubot to TOP position and tracking trajectory.*

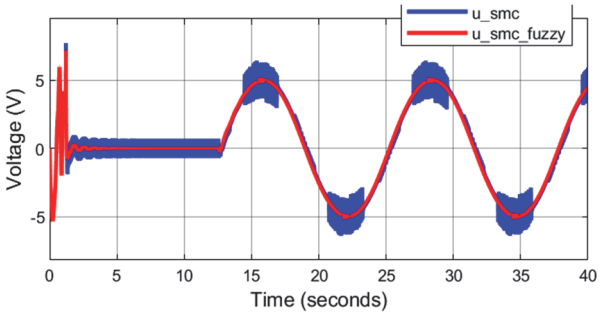


Fig. 17 – *Input states under swing-up controller and SMC (blue) and with FSMC (red) controllers to move pendubot to TOP position and tracking trajectory.*

The trajectory is defined as follows:

$$\begin{cases} q_1 = -\pi/2 + 0.6\sin(0.5t), \\ \dot{q}_1 = 0, \\ q_2 = \pi - 0.6\sin(0.5t), \\ \dot{q}_2 = 0, \end{cases} \quad \text{when} \quad (t > 4\pi). \tag{35}$$

Both balance controllers can track the trajectory and, as mentioned above for the main purpose of the proposed method, the chattering phenomenon is reduced by FSMC.

4.4 Experiment results

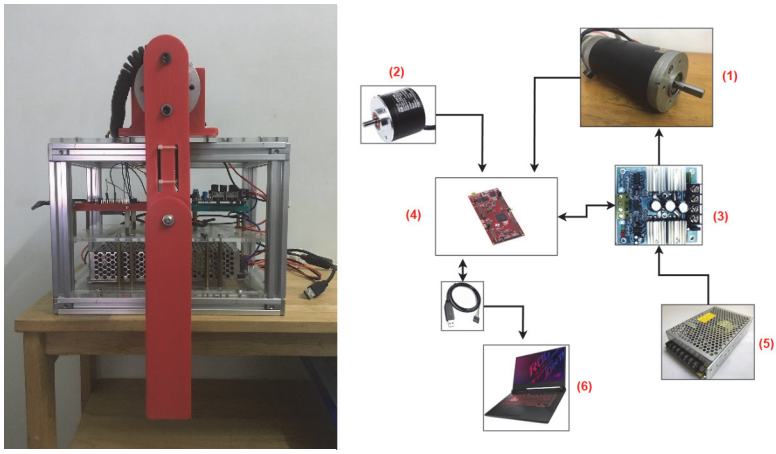


Fig. 18 – *Real experimental pendubot and hardware platform.*

Our real pendubot consists of:

– (1) DC motor 24VDC M.A.E with 2000 pulses encoder;

– (2) 1000 pulse per round Omron encoder;
– (3) H-bridge;
– (4) Microcontroller TMSF28379D;
– (5) DC power supply;
– (6) Laptop to communicate with the microcontroller.

4.4.1 TOP position

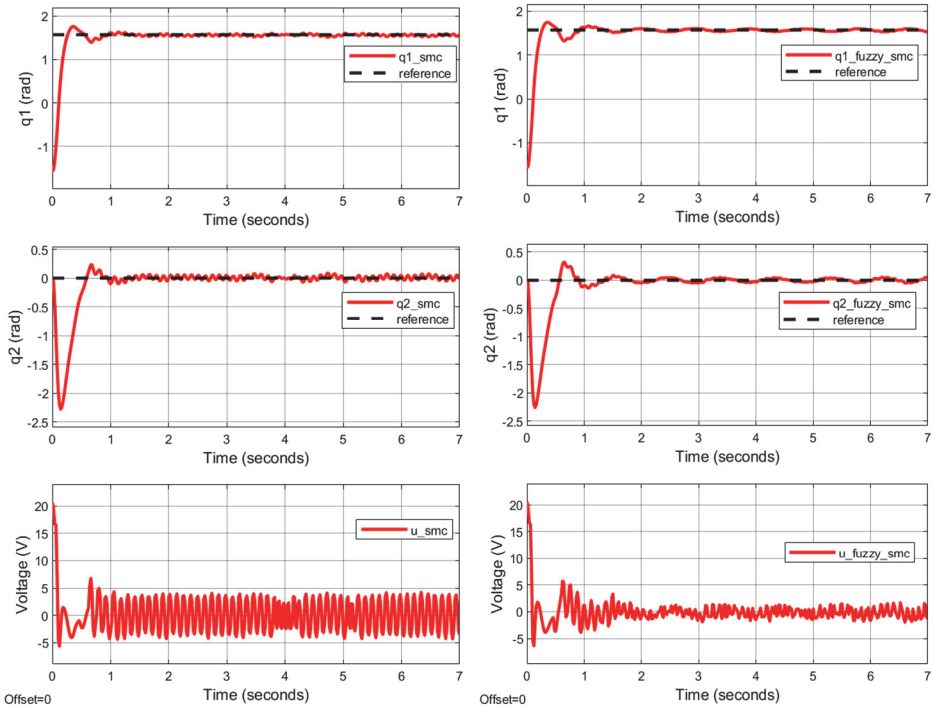


Fig. 19 – *The output and input states of pendubot when the swing-up controller is associated with SMC (left) and with FSMC (right) controllers at the TOP position.*

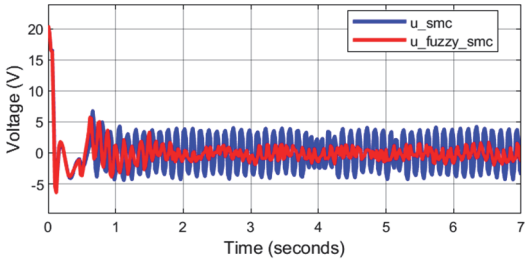


Fig. 20 – *Input states of pendubot when swing-up controller is associated with SMC (blue) and with FSMC (red) controllers at the TOP position.*

The experiment results show the effectiveness of the FSMC. The voltage applied to the motor is reduced (see Fig. 20) and the quality remains the same as with the classical SMC (Fig. 19). Furthermore, the chattering in the real pendubot almost vanishes when applying the proposed design to improve the SMC.

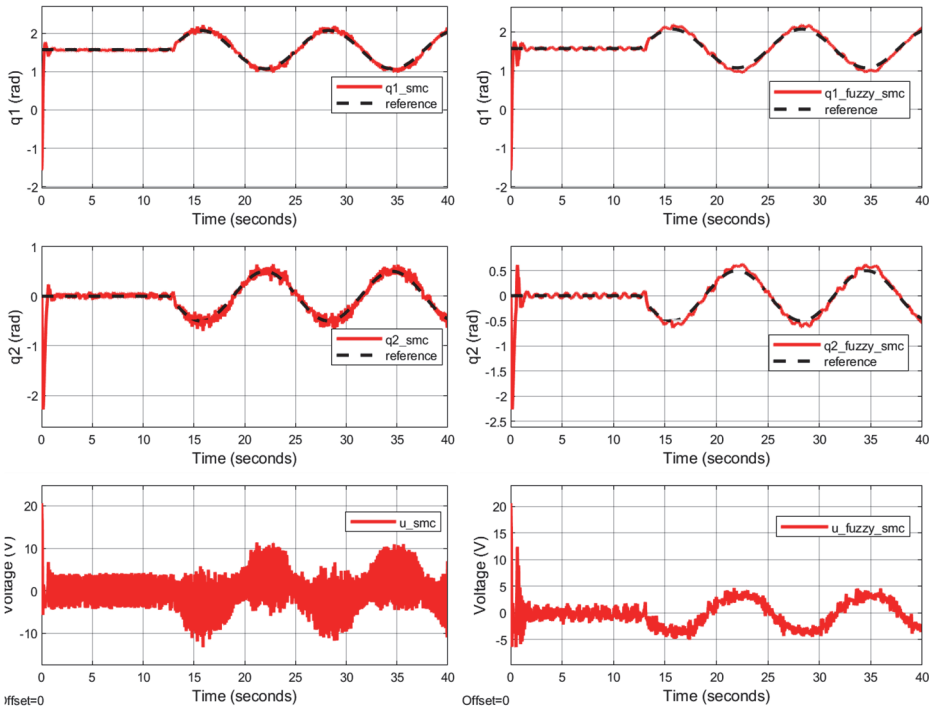


Fig. 21 – *Output and input states of pendubot when swing-up controller is associated with SMC (left) and with FSMC (right) controllers at the TOP under tracking control.*

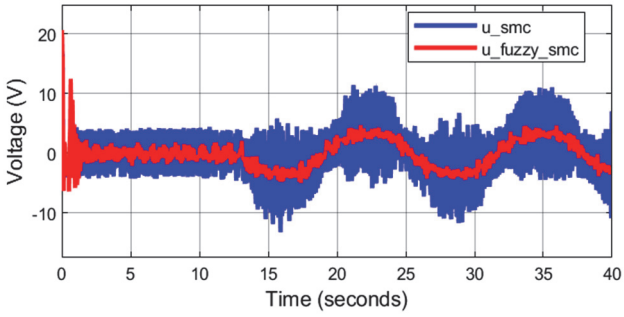


Fig. 22 – *Input states of the pendubot swing-up controller associated with SMC (blue) and FSMC (red) controllers at the TOP under tracking control.*

Fig. 22 shows very clearly how the robustness and effectiveness of FSMC compare with SMC. While SMC needs at most 10 V to enable the system to track the defined trajectory, FSMC needs approximately 5 V, which is less than half of the voltage of SMC and stabilises the system with the same quality as SMC (Fig. 21).

4.4.2 MID position

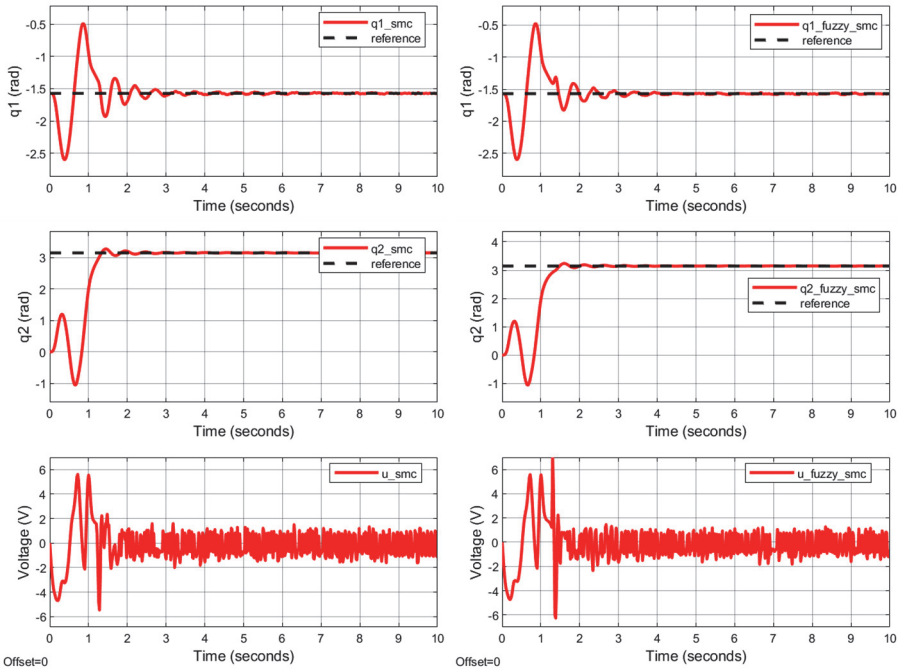


Fig. 23 – *Output and input states of pendubot when the swing-up controller is associated with SMC (left) and with FSMC (right) controllers at MID position.*

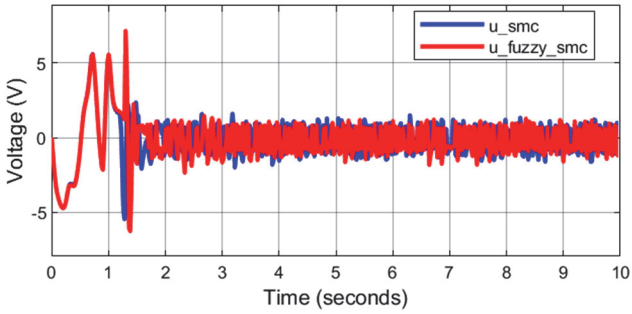


Fig. 24 – *Input states of pendubot when the swing-up controller is associated with SMC (blue) and with FSMC (red) controllers at MID position.*

Both controllers give good output states responses to the references. The voltage from FSMC is slightly better than SMC when balancing at the MID position, but this will not be the point at which to consider that FSMC is better than SMC, because the next results will show the effectiveness and robustness of the proposed method when the equilibrium points are no longer fixed.

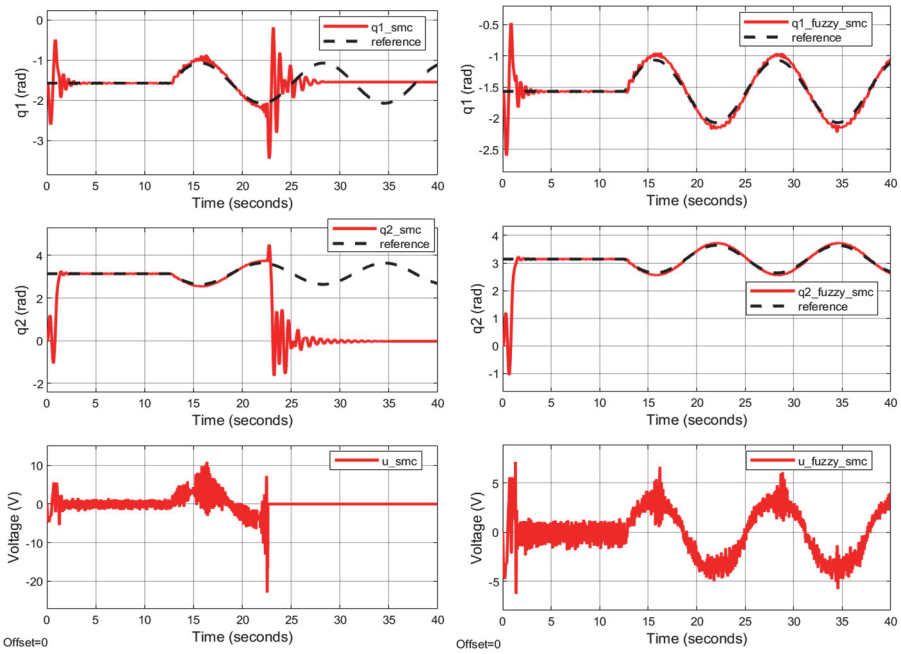


Fig. 25 – *Output and input states of pendubot when the swing-up controller is associated with SMC (left) and FSMC (right) controllers under balancing at MID position and tracking trajectory.*

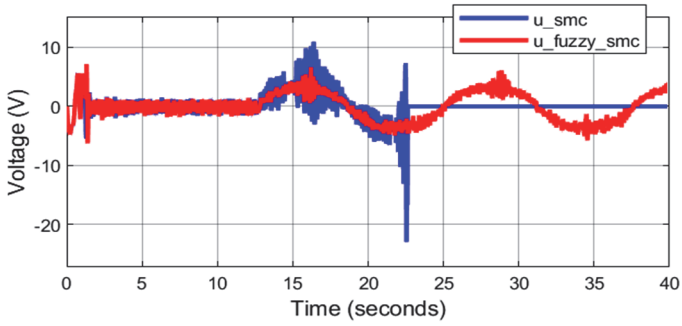


Fig. 26 – *Input states of pendubot when the swing-up controller is associated with SMC (blue) and FSMC (red) controllers under balancing at MID position and tracking trajectory.*

The results in Fig. 25 show that SMC is able to balance the system, but it is not successful when controlling the pendubot tracking the trajectory. It is successful in the first half of the sine wave but false in the second half and then completely out of control. Considering the first half of the trajectory that SMC is able to track, the voltage input (Fig. 26) is up to 10 V, which is much higher than the FSMC voltage.

To make the comparison more reliable, we re-select a new coefficient of the SMC. With the coefficient $k = 25$ the SMC is not able to track the trajectory. By reducing to $k = 5$, through trial and error, this is the largest k that enables the SMC controller to keep the system to both the balancing and tracking trajectory.

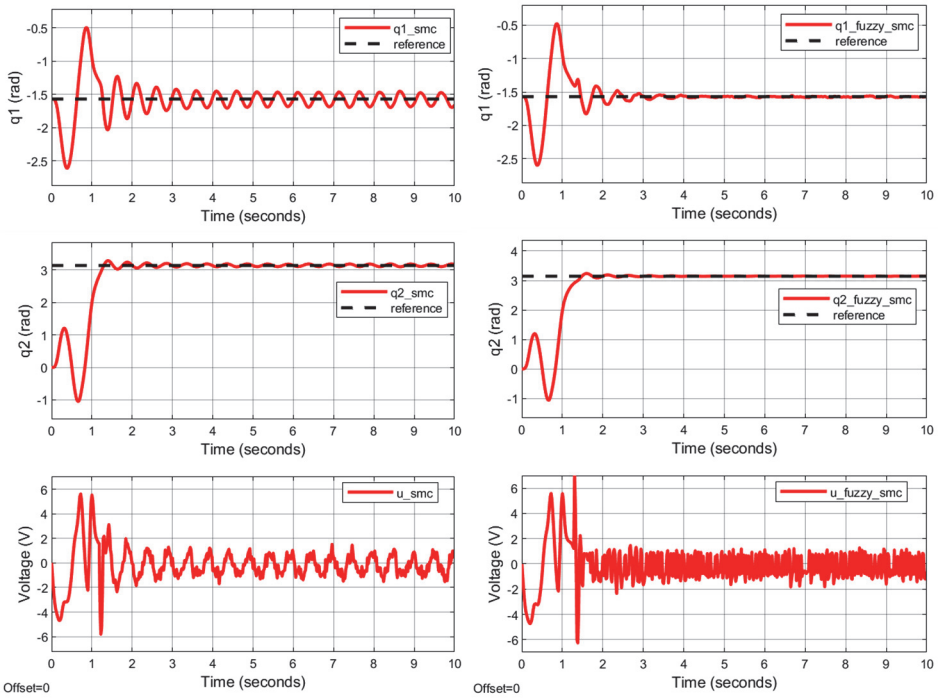


Fig. 27 – *Output and input states of pendubot when the swing-up controller is associated with SMC (left) and FSMC (right) controllers at the MID position.*

In Fig. 27, the voltage part of the SMC, at first glance, seems much better than that of the FSMC, but the states and of the SMC controller fluctuate back and forth around the set point, which is shown more clearly in Figs. 29 and 30, while for the FSMC, they stick to the set point. Therefore, for this case, the FSMC is more robust than the SMC.

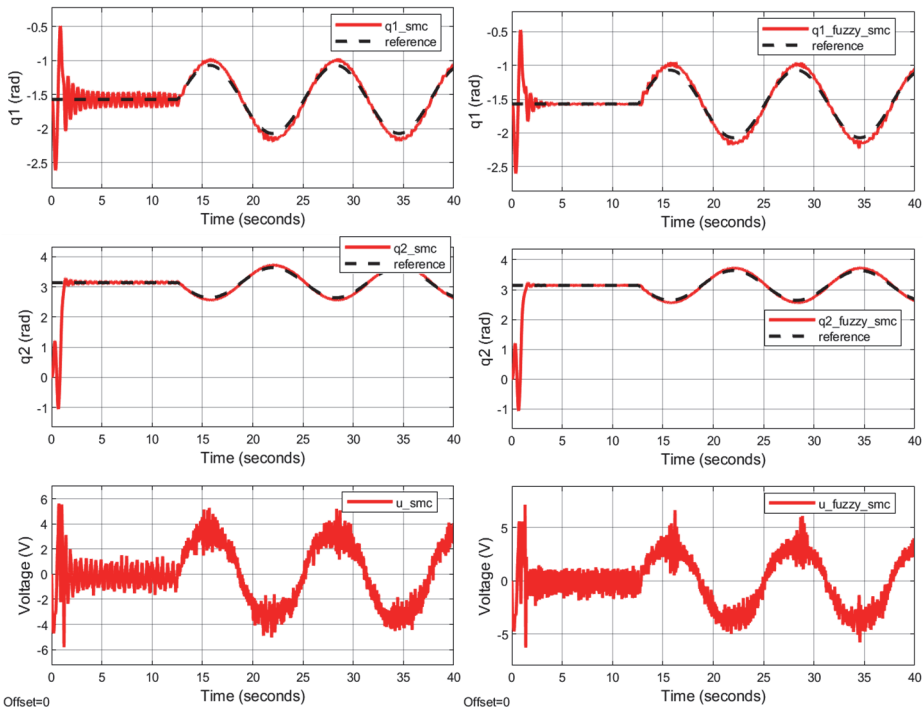


Fig. 28 – *Output and input states of pendubot when the swing-up controller is associated with SMC (left) and FSMC (right) controllers under balancing at MID position and tracking trajectory.*

In Fig. 28, given the wider operating range of the system, we can see that the states and of the FSMC and SMC controller are mostly the same when the pendubot system is tracking the defined trajectory. Overall, the FSMC gives a wider operating range than the SMC, however.

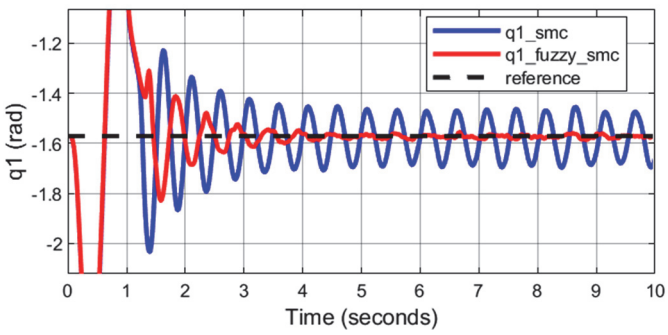


Fig. 29 – *Output states of the rotational arm of pendubot when the swing-up controller is associated with SMC (blue) and FSMC (red) controllers at the MID position.*

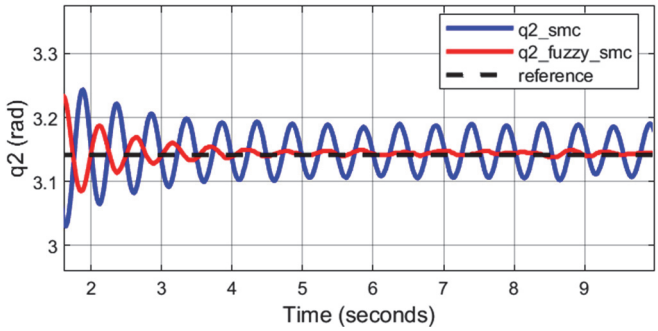


Fig. 30 – *Output states of the pendulum of pendubot when the swing-up controller is associated with SMC (blue) and with FSMC (red) controllers at the MID position.*

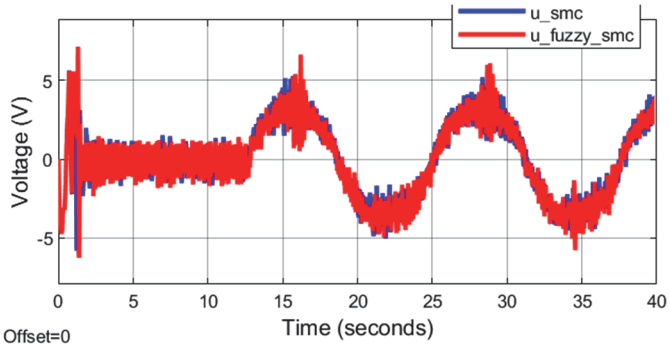


Fig. 31 – *Input states of pendubot system when the swing-up controller is associated with SMC (blue) and FSMC (red) controllers at the MID position.*

In Fig. 31, the voltage supplied to the motor when using SMC and FSMC is compared and it can be seen that SMC gives a slightly better voltage level compared to FSMC, but, as explained in Figs. 27 and 28, and shown in Figs. 29 and 30, the voltage of the SMC gives the tracking results of the state variables to the set points, which are quite bad and worse than those of the FSMC. Moreover, while the SMC is able to balance the pendubot at the MID position, it gives poor stability of the outputs shown in Figs. 29 and 30 on the rotational arm and pendulum. The proposed FSMC solves these problems.

Both SMC and FSMC are able to stabilize when tracking the defined MID trajectory. But FSMC gives a wider range in balancing at the fixed equilibrium point and tracking the defined trajectory (Figs. 27, 28, and 31).

5 Conclusion

In this study, the identification method for the pendubot, SMC, and FSMC for the stability control of the system at the equilibrium points and trajectory

tracking are proposed. The experiment and simulation results show that the SMC algorithm makes the system stable at the equilibrium points and defined trajectories. The "chattering" phenomenon caused by SMC is reduced by fuzzy logic. The swing-up controller is designed based on the partial feedback linearization algorithm. The experimental and simulation results have proved the effectiveness and robustness of the proposed methods.

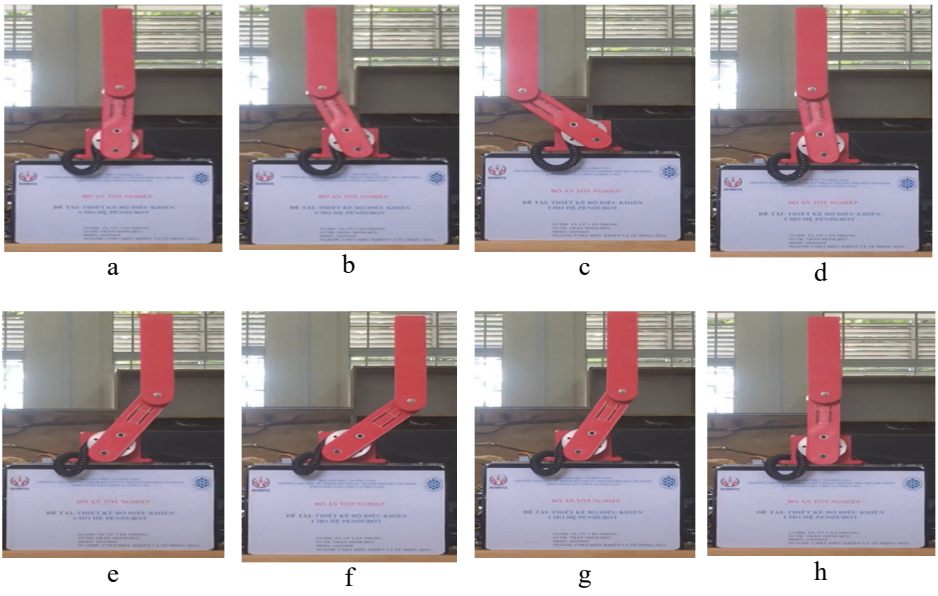
In future research, adaptive fuzzy sliding mode control or adaptive neuron network sliding mode control can be studied for comparison with the fuzzy sliding mode control and classical sliding mode control in both simulation and practical experiments.

6 Appendix

The experimental results in the laboratory are listed in different cases:

– TOP position:

swing-up, balancing, and trajectory tracking (a, b, c, d, e, f, g, and h, respectively).

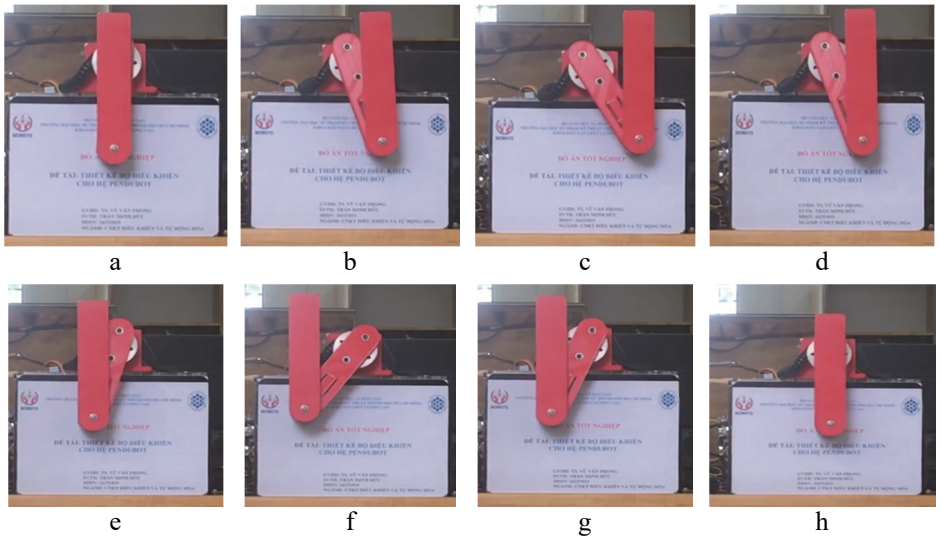


a b c d

e f g h

– MID position:

swing-up, balancing, and trajectory tracking (a, b, c, d, e, f, g, h, respectively).



a b c d

e f g h

7 References

[1] L. Wang, X. Lai, P. Zhang, M. Wu: Motion Control Strategy Based on Integrated Trajectory for the Pendubot, Proceedings of the European Control Conference (ECC), Delft, Netherlands, June 2021, pp. 2151 – 2156.

[2] R. Jain, S. J. Mija, D. Kumar: Design of Robust Sliding Mode Controllers for a Class of Underactuated Robotic Systems, Proceedings of the 2nd International Conference on Intelligent Technologies (CONIT), Hubli, India, June 2022, pp. 1 – 6.

[3] M. W. Spong, D. J. Block: The Pendubot: A Mechatronic System for Control Research and Education, Proceedings of the 34th IEEE Conference on Decision and Control, New Orleans, USA, December 1995, pp. 555 – 556.

[4] T. V. Toan, T. T. Ha, T. V. Do: Hybrid Control for Swing Up and Balancing Pendubot System: An Experimental Result, Proceedings of the International Conference on System Science and Engineering (ICSSE), Ho Chi Minh City, Vietnam, July 2017, pp. 450 – 453.

[5] M. Zhang, T.- J. Tarn: Hybrid Control of the Pendubot, IEEE/ASME Transactions on Mechatronics, Vol. 7, No. 1, March 2002, pp. 79 – 86.

[6] M. W. Spong: The Swing Up Control Problem for the Acrobot, IEEE Control Systems Magazine, Vol. 15, No. 1, February 1995, pp. 49 – 55.

[7] I. Fantoni, R. Lozano, M. W. Spong: Energy Based Control of the Pendubot, IEEE Transactions on Automatic Control, Vol. 45, No. 4, April 2000, pp. 725 – 729.

[8] X. Xin, M. Kaneda, T. Oki: The Swing Up Control for the Pendubot Based on Energy Control Approach, Proceedings of the 15th Triennial World Congress of the International Federation of Automatic Control, Barcelona, Spain, July 2002, pp. 461 – 466.

[9] K. J. Astrom, K. Furuta: Swinging Up a Pendulum by Energy Control, Automatica, Vol. 36, No. 2, February 2000, pp. 287 – 295.

[10] D. Qian, J. Yi, D. Zhao: Hierarchical Sliding Mode Control for a Class of SIMO Under-Actuated Systems, Control and Cybernetics, Vol. 37, No. 1, January 2008, pp. 159 – 175.

[11] D. Zehar, K. Benmahammed: Optimal Sliding Mode Control of the Pendubot, Global Journal of Computer Science and Programming, Vol. 1, No. 2, June 2014, pp. 8 – 14.

[12] O. Begovich, E. N. Sanchez, 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, January 2002, pp. 14 – 20.

[13] K. Graichen, M. Zeitz: Nonlinear Feedforward and Feedback Tracking Control with Input Constraints Solving the Pendubot Swing Up Problem, IFAC Proceedings Volumes, Vol. 38, No. 1, 2005, pp. 1112 – 1117.

[14] K. Graichen, V. Hagenmeyer, M. Zeitz: A New Approach to Inversion-Based Feedforward Control Design for Nonlinear Systems, Automatica, Vol. 41, No. 12, December 2005, pp. 2033 – 2041.

[15] S. Borisenok, S. Ullah: Linear Feedforward Control of Two-Level Quantum System by Modulated External Field, Optics Communications, Vol. 284, No. 14, July 2011, pp. 3562 – 3567.

[16] R. W. O'Flaherty, R. G. Sanfelice, A. R. Teel: Hybrid Control Strategy for Robust Global Swing-Up of the Pendubot, Proceedings of the American Control Conference, Seattle, USA, June 2008, pp. 1424 – 1429.

[17] S. Ullah: Two-Level Decay Quantum System with Open-Loop Control Optical Field, Journal of the Optical Society of America B, Vol. 30, No. 2, February 2013, pp. 428 – 430.

[18] S. Rudra, R. K. Barai: Design of Block Backstepping Based Nonlinear State Feedback Controller for Pendubot, Proceedings of the IEEE 1st International Conference on Control, Measurement and Instrumentation (CMI), Kolkata, India, January 2016, pp. 1 – 5.

[19] S. Ullah: Feedback Control of Probability Amplitudes for Two-Level Atom in Optical Field, Optics Communications, Vol. 281, No. 4, February 2008, pp. 640 – 643.

[20] C. Wei, Y.- H. Chen, T. Chai, J. Fu: Adaptive Robust Control for Pendubot with Matched–Mismatched Uncertainty via Constraint-Following, Robotica, Vol. 41, No. 5, May 2023, pp. 1550 – 1567.

[21] S. D. A. Sanjeewa, M. Parnichkun: Control of Rotary Double Inverted Pendulum System Using LQR Sliding Surface Based Sliding Mode Controller, Journal of Control and Decision, Vol. 9, No. 1, April 2021, pp. 89 – 101.

[22] M. Idrees, S. Muhammad, S. Ullah: Robust Hierarchical Sliding Mode Control with State-Dependent Switching Gain for Stabilization of Rotary Inverted Pendulum, Kybernetika, Vol. 55, No. 3, June 2019, pp. 455 – 471.

[23] M. Idrees, S. Ullah, S. Muhammad: Sliding Mode Control Design for Stabilization of Underactuated Mechanical Systems, Advances in Mechanical Engineering, Vol. 11, No. 5, May 2019, pp. 1 – 10.

[24] S. Abderazak, N. Farid: Comparative Study Between Sliding Mode Controller and Fuzzy Sliding Mode Controller in a Speed Control for Doubly Fed Induction Motor, Proceedings of the 4th International Conference on Control Engineering & Information Technology (CEIT), Hammamet, Tunisia, December 2016, pp. 1 – 6.

[25] L. Wu, P. Mei, B. Lei, Z. Lin: Dead-Beat Terminal Sliding Mode Control with Application to DC-DC Buck Converters, IEEE Transactions on Circuits and Systems II: Express Briefs, Vol. 70, No. 7, July 2023, pp. 2470 – 2474.

[26] W. Liu, S. Chen, H. Huang: Actor-Critic Learning Hierarchical Sliding Mode Control for a Class of Underactuated Systems, Proceedings of the Chinese Automation Congress (CAC), Hangzhou, China, November 2019, pp. 1 – 6.

[27] B. Majout, B. Bossoufi, M. Bouderbala, M. Masud, J. F. Al-Amri, M. Taoussi, M. El Mahfoud, S. Motahhir, M. Karim: Improvement of PMSG-Based Wind Energy Conversion System Using Developed Sliding Mode Control, Energies, Vol. 15, No. 5, February 2022, p. 1625.