processes

Article

Validation of a Classical Sliding Mode Control Applied to a Physical Robotic Arm with Six Degrees of Freedom

Andres González-Rodríguez, Rogelio E. Baray-Arana *, Abraham Efraím Rodríguez-Mata, Isidro Robledo-Vega and Pedro Rafael Acosta Cano de los Ríos

Tecnólógico Nacional de México, IT de Chihuahua, Ave. Tecnólógico #2909, Chihuahua 31310, Mexico
* Correspondence: rogelio.ba@chihuahua.tecnm.mx

Abstract: The control of robotic manipulators has become increasingly difficult over recent years due to their high accuracy, performance, speed, and reliability in a variety of applications, such as industry, medicine, research, etc. These serial manipulator systems are extremely complex because their dynamic models include perturbations, parametric variations, coupled nonlinear dynamics, and non-modular dynamics, all of which require robust control for trajectory tracking. This paper compares two control techniques: computational torque control (CTC) and sliding mode control (SMC). In this study, the latter was used for a physical robotic arm with six degrees of freedom (DOF) and online experiments were conducted, which have received little attention in the literature. As a result, the contribution of this work was based on the real-time application of this controller via a self-developing interface. The great resilience of sliding mode controllers to disturbances was also demonstrated in this study.

Keywords: sliding mode control; six degrees of freedom; manipulator; real-time application

check for updates

Citation: González-Rodríguez, A.; Baray-Arana, R.E.; Rodríguez-Mata, A.E.; Robledo-Vega, I.; Acosta Cano de los Ríos, P.R. Validation of a Classical Sliding Mode Control Applied to a Physical Robotic Arm with Six Degrees of Freedom. *Processes* **2022**, *10*, 2699. https://doi.org/10.3390/pr10122699

Academic Editors: Francisco Ronay López-Estrada and Guillermo Valencia-Palomo

Received: 19 March 2022
Accepted: 24 April 2022
Published: 14 December 2022

1. Introduction

Serial manipulators can perform complex tasks that human beings cannot perform because they could be harmed or because they do not have the precision or force, or either, that are required to perform those tasks. However, to fully fulfill this ability, serial manipulators require robust control. Thus, the design of manipulators needs to consider nonlinear dynamical models, parameter uncertainties, perturbations, and non-model dynamics. PID and PD controllers are some of the most widely applied control schemes for robot manipulators due to the simplicity of their implementation. However, these types of controllers have some drawbacks. A PD controller needs a gravitational term in the control law for nonplanar manipulators [1]. Meanwhile, a PID controller requires a tuning procedure to allow good performance [2]. These controllers also lack precision for non-modeled dynamics, such as friction or unknown external torques. Finally, they can only be used to track set points [3].

Trajectory tracking and motion control refer to when a manipulator follows a proposed path, which is usually obtained using inverse kinematics. One of the common solutions is the CTC (which is a feedback linearization method). This method has the disadvantages of knowing the dynamical model a priory and being not robust. To counter these problems, control researchers have combined CTC with other techniques, such as adaptive control [4], fuzzy logic [5], and neural networks [6]. Although these methods can estimate the dynamics of the proposed model, they are not robust to perturbations.

SMC is a robust control method that has been extensively studied by control researchers and demonstrates qualities such as parametric in-variance, dynamic collapse, and asymptotic convergence in the presence of perturbations [7,8]. One way to design an SMC is to use the equivalent control method (ECM) [9]. This method introduces a known part of the model system into the control law to reduce the chattering phenomena, which are a well-known drawback of SMC systems. By doing this, the SMC only deals with the nonmodeled dynamics and the estimation errors of the parameters [10–12].

In the field of robotics, the known part of an ECM is the dynamic model of a robotic arm [13]. These dynamic models are complex due to the coupling of the chain of masses and inertia and the large number of terms that appear for arms with more than three DOFs [14]. For this reason, in the literature, various authors have combined different control techniques and computer algorithms to estimate the known part of the ECM. In [15], Bailey and Arapostathis used a classical SMC surface with the known part of the ECM in a 2DOF manipulator. Kumar and Prasad [16–18] compared the use of CTC and SMC to the known part of the ECM, which showed the advantage of using SMC over the ECM in a 3DOF manipulator. The paper [19] proposed a novel sliding mode control (NSMC) that was based on an extended gray wolf optimizer (EGWO). The NSMC employed a PD surface with an exponential term that was combined with the ECM in a 2DOF manipulator and surface gains were selected using the EGWO. However, as in CTC, it is necessary to know the dynamic model in the ECM. In [20], Thuan et al. combined intelligent control with second-order sliding mode control (SOSMC). A radial basis function network (RBFN) calculated the dynamic model and nonmodeled dynamics of a 4DOF dual arm (DAM) (2DOF in each arm). Furthermore, Thuan et al. in [21] used model reference adaptive control (MRAC) instead of RBFN with SOSMC for the same DAM. An optimal sliding mode control approach, which was used in [22], combined optimal control with SMC for optimal robustness properties. In this approach, an observer of the disturbance was in charge of the adaptive part of the control and applied this control to the 2DOF planar manipulator.

There are two main problems with the techniques that were presented in [9–14]. One is that SMC is a simple control law that depends only on the selection of a gain and a surface, which, in turn, depends on the state of the system [23]. The use of this technique makes the control law and its application more difficult. Second, they only simulate the control strategy. For implementation, ref. [24] proposed an SMC with a sliding perturbation observer (SPO). The SPO estimated the reaction force of a 6DOF DAM (3DOFs in each arm). Jeong et al. [25] applied a super-twisting algorithm (STA) with an adaptive law to an industrial 4DOF robot using only 2DOF and an XY planar manipulator. Paper [26] presented an adaptive sliding mode neural network control (ASMNN). This control approach coupled the SMC with a radial basis function neural network (RBFNN), which was implemented in a three-link robot manipulator (3DOF). Similarly, ref. [27] used the same control strategy for the same manipulator, taking into account the dead zone.

Control techniques for other types of system can also be found in the literature. In [28], an NN in combination with a fractional order SMC was used to control an active power filter. The NN was in charge of the estimation of the uncertainties and nonlinearities, while the fractional order SMC improved the precision and performance of the control. The controller was implemented in real time and produced excellent results.

Ref. [29] combined a pole placement control, time delay estimation (TDE), and adaptive sliding mode control. The authors implemented the controller with a 6DOF Mitsubishi robot that used only 2DOF. To increase robustness in the reaching phase of the SMC, article [30] proposed an adaptive integral sliding mode control (AISMC) with a TDE, which was implemented with 3DOF of a 6DOF PUMA robot. The articles [15–20] only implemented controllers with a few DOFs of the manipulator. Hee et al. in [31] implemented a PID controller for the gravitational term of a physical 7DOF DAM. An SPO was used to estimate the evaluation force of the manipulator. However, as mentioned above, the PID controller could not be used for trajectory tracking.

The literature has proposed a variety of control techniques. Most of them involve the simulation of the controllers. For the implemented controllers, the majority used fewer DOFs than were available in the corresponding system. One example was an implementation of a 7DOF controller, but only for set-point tracking. Another point to address is the gain selection of the sliding-mode control. For a robot driven by DC motors, the limit gain is the maximum voltage in the power converter. When the controller needs more energy, the robot cannot reach or stay on the surface. The main focus of this paper was to find the voltage (SMC gain) that is needed for trajectory tracking. The novelty of this paper was the procedure that was used to validate the SMC gain in an n-DOF manipulator driven by DC motors. To the authors' knowledge, there has not yet been a discussion regarding the verification of the above.

In [32], a suggested strategy was compared to achieve finite-time convergence, flicker-free control input, superior tracking performance, and resilience of the robotic manipulator. The difficulty of this type of method is implementation in real time, as in this project, since the programming of fractional control algorithms is quite complex to carry out in an embedded way. In [33], an unmanned aerial vehicle (UAV) system with 6 degrees of freedom (6DOF) and external disturbances that corresponded to sensor failure was discussed. This form of pure SMC controller is still employed in robotics, and we utilized it in this paper to control the manipulator robot in a robust trajectory, following our proposed work. Using Lyapunov's theory, see Ref. [34] showed that a well-designed control could ensure that transnational and rotational tracking errors converge at the origin in a finite amount of time. However, only numerical simulations were carried out to demonstrate that the developed control scheme had a high level of robustness and a quick convergence time and demonstrated elimination of entry saturation and suppression of chattering. In this study, we proposed the use of real-time control through an SMC.

In this paper, a classical SMC dealt with the nonmodeled dynamics for the error estimation of the parameters and perturbations of a 6DOF manipulator that was driven by DC motors. The dynamical model was not applied to the control law (the known part of the ECM), which made the control law of the SMC simpler. When the numerical simulations were compared to those of CTC and SMC, SMC showed better robustness with easy implementation. The SMC experiments were performed using self-developed hardware that sent and received data between MATLAB and the robot. SMC gain was validated using Lyapunov stability analysis and actuator dynamics.

The paper is organized as follows. The mathematical model of the actuator manipulator is presented in Section 2. Section 3 deals with the CTC and the SMC. Numerical simulations of the CTC and SMC are presented in Section 4. Section 5 details the experimental results of the obtained SMC gain. Finally, Section 6 discusses the conclusions and areas for further development.

2. Mathematical Problem and Model

In this section, we introduce the mathematical model that was used to construct the robust control that was based on sliding modes. We rely on the most popular mathematical models from the literature, since this type of dynamic modeling is still used for the design of automatic control technology [11]. Therefore, the classical dynamic model of an n-DOF manipulator is expressed as follows.

$$D(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + C(\mathbf{q}(t), \dot{\mathbf{q}}(t))\dot{\mathbf{q}}(\mathbf{t}) + \mathbf{g}(\mathbf{q}(t)) = \mathbf{u}(t) \tag{1}$$

The inertia, Coriolis, and centrifugal matrices and the gravity vector are given by $D(\mathbf{q})$, $C(\mathbf{q}, \dot{\mathbf{q}})$, and $\mathbf{g}$, respectively, and $\mathbf{u}(\mathbf{t})$ is the control input of the system. By representing (1) as a separation of equations for each DOF, it yielded the following.

$$\sum_{j=1}^{n} d_{kj}(\mathbf{q})\ddot{q} + \sum_{i,j}^{n} c_{ijk}(\mathbf{q})\dot{q}_i\dot{q}_j + g_k(\mathbf{q}) = u_k \tag{2}$$

where k is the DOF of the manipulator.

Much of the previous research has focused on electrical positions, which was helpful as it allowed us to model more accurately and helped us develop our controller. The control input of the dynamic Equation (1) was the torque that was produced by the actuators in each DOF. In this case, the manipulator was driven by DC motors. The system was the input for the voltage in the motor terminals. For this reason, we could obtain the input of the model in voltage terms by combining the dynamical models of the manipulator and the actuators. The linear model of a DC motor was given by:

$$\begin{aligned} v_a(t) &= L_a\frac{\mathrm{d}i_a(t)}{\mathrm{d}t} + R_a i_a(t) + K_e\frac{\mathrm{d}\theta(t)}{\mathrm{d}t} \\ K_t i_a(t) &= J\frac{\mathrm{d}^2\theta(t)}{\mathrm{d}t^2} + B\frac{\mathrm{d}\theta(t)}{\mathrm{d}t} + n\tau(t) \end{aligned} \tag{3}$$

where the inductance L_a, electrical resistance R_a, and electrical constant K_e are the electrical parameters and the rotor inertia J, viscous friction B, and mechanical constant K_t are the mechanical parameters. The current and voltage are $v_a = v_a(t)$ and $i_a = i_a(t)$, the motor shaft angular displacement is $\theta = \theta(t)$, n is the transmission ratio, and $\tau = \tau(t)$ is the toque load, which is also known as the disturbance.

When we coupled Equations (1) and (3), there were three state variables (angular position, angular velocity, and current). In the literature, we found that Equation (3) could be reduced to one state variable. To achieve this, we compared the magnitude of the electrical time constant (ETC) with the mechanical time constant (MTC) by defining these constants as:

$ETC = \frac{L_a}{R_a}$

$MTC = \frac{J}{B}$

For the manipulator used in this article, the MTC was 168 times higher than the ETC in the arm motor and 40 times higher in the wrist motors. Dividing Equation (3) by R_a, i.e., $L_a/R_a = 0$, and writing Equation (3) according to its components yielded:

$$J_k\ddot{\theta}_k + (B + \frac{K_{ek}K_{tk}}{R_{ak}})\dot{\theta}_k = \frac{K_{tk}}{R_{ak}}v_{ak} - n_k\tau_k \tag{4}$$

By dividing Equation (4) by n_k and expressing $B = (K_{ek}K_{tk}/R_{ak})$, we obtained:

$$\frac{J_k}{n_k}\ddot{\theta}_k + \frac{B}{n_k}\dot{\theta}_k = \frac{K_{tk}}{R_{ak}n_k}v_{ak} - \tau_k \tag{5}$$

where θ_k is the angular displacement that the motor saw. When we wanted to combine (5) and (2), θ_k was necessary in terms of the manipulator. By substituting $\theta_k = \frac{q_k}{n_k}$ into Equation (5), it yielded:

$$\frac{J_k}{n_k^2}\ddot{q}_k + \frac{B}{n_k^2}\dot{q}_k = \frac{K_{tk}}{R_{ak}n_k}v_{ak} - \tau_k \tag{6}$$

When comparing Equation (2) with (6), the control input of (2) was the load torque of (6). By combining these equations, we obtained a model for the manipulator that was driven by a DC motor, which was presented as follows:

$$\frac{J_k}{n_k^2}\ddot{q}_k + \sum_{j=1}^{n} d_{kj}(\mathbf{q})\ddot{q} + \sum_{i,j}^{n} c_{ijk}(\mathbf{q})\dot{q}_i\dot{q}_j + \frac{B}{n_k^2}\dot{q}_k + g_k(\mathbf{q}) = \frac{K_{tk}}{R_{ak}n_k}v_{ak}$$

or as matrices:

$$(D(\mathbf{q}) + J)\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + B\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{u} \tag{7}$$

where

$D(\mathbf{q}), C(\mathbf{q},\dot{\mathbf{q}}), B, J \in \mathbb{R}^{nxn}$

$\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{g}, \mathbf{u} \in \mathbb{R}^{nx1}$

where J and B are diagonal matrices with terms J_k/n_k^2 and B_k/n_k^2, respectively. The control input was given by a vector with terms:

$$u_k = \frac{K_k}{R_{ak}n_k}v_{ak} \tag{8}$$

As in the analysis that was performed in [1], we could compute the voltage input that was needed for the control.

3. Computed Torque Control and Sliding Mode Control

3.1. CTC

CTC is a feedback linearization technique that uses the dynamic model of a manipulator in the control law. To achieve this, we write Equation (7) as follows:

$$M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{u} \tag{9}$$

where

$M(\mathbf{q}) = (D(\mathbf{q}) + J)$
$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + B\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})$

As $M(\mathbf{q})$ was full-rank and square, its inverse also existed. Thus, we selected a control law as follows:

$$\mathbf{u} = M(\mathbf{q})\mathbf{v} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \tag{10}$$

where

$$\mathbf{v} = -\alpha\mathbf{q} - \beta\dot{\mathbf{q}} + \mathbf{r(t)} \tag{11}$$

where α and β are diagonal matrices and $\mathbf{r(t)}$ is the trajectory of each DOF, which was defined as:

$$\mathbf{r(t)} = \ddot{\mathbf{q}}_d(t) + \alpha\dot{\mathbf{q}}_d(t) + \beta\mathbf{q}_d(t) \tag{12}$$

Equation (11) contained the position ($\mathbf{q}_d(t)$), velocity ($\dot{\mathbf{q}}_d(t)$), and acceleration ($\ddot{\mathbf{q}}_d(t)$) of the trajectory that was proposed. Substituting Equation (11) into (10) yielded the following:

$$\mathbf{v} = \ddot{\mathbf{q}}_d(t) + \alpha\dot{\mathbf{e}}(t) + \beta\mathbf{e}(t) \tag{13}$$

where $\mathbf{e}(t) = \mathbf{q}_d(t) - \mathbf{q}$ is the error of the trajectory tracking.

Applying the control (10) to the dynamic model (9), we obtained the following.

$$\ddot{\mathbf{e}}(t) + \alpha\dot{\mathbf{e}}(t) + \beta\mathbf{e}(t) = \mathbf{0} \tag{14}$$

Equation (14) showed a homogeneous second-order linear differential equation. Another property exhibited by Equation (14) was that the set of equations did not lump together. In this sense, we could treat each differential equation separately. With this in mind, we added a disturbance to Equation (9) and, using the same control law as (10), we obtained the following:

$$\ddot{\mathbf{e}}(t) + \alpha\dot{\mathbf{e}}(t) + \beta\mathbf{e}(t) = \mathbf{f}(t) \tag{15}$$

where $\mathbf{f}(t) \in \mathbb{R}^{nx1}$

As the equations were not lumped, we could use one equation for the Lyapunov stability analysis. To do this, we passed the first equation into state-space. By selecting $x_1(t) = e_1(t)$ and $x_2(t) = \dot{x}_1(t) = \dot{e}_1(t)$, we wrote the state-space equation as:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -\alpha_{11}x_2(t) - \beta_{11}x_1(t) + f_1(t) \end{aligned} \tag{16}$$

Then, using the following Lyapunov candidate function:

$$V = \frac{1}{2}\alpha_{11}x_1^2(t) + \frac{1}{2}x_2^2(t) \tag{17}$$

taking the time derivative of the Lyapuniov function, it yielded:

$$\dot{V} = \begin{bmatrix} \frac{\partial V}{\partial x_1} & \frac{\partial V}{\partial x_2} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \tag{18}$$

$$\dot{V} = -\alpha_{11} x_2^2(t) + x_2 f_1(t) \tag{19}$$

The function presented in Equation (19) was not definitely negative because we could not use a static negative sign in the term $x_2 f_1(t)$. When we did not take the disturbance into account, Equation (19) became:

$$\dot{V} = -\alpha_{11} x_2(t)^2 \tag{20}$$

The above derivative was always negative, but only in the state $x_2(t)$. To prove total stability, we used La Salle's theorem with $\dot{V} = 0$:

$$0 = -\alpha_{11} x_2^2(t) \tag{21}$$

This meant that $x_2(t) = 0$. Thus, its derivative also equaled zero: $\dot{x}_2(t) = 0$. Substituting $x_2(t)$ and $\dot{x}_2(t)$ into the state-space Equation (16) yielded the following result:

$$\begin{aligned} \dot{x}_1(t) &= 0 \\ 0 &= -\beta_{11} x_1(t) \end{aligned} \tag{22}$$

where $x_1(t) = 0$. In this manner, we show that both states become zero when following the desired trajectory with no disturbances in the system.

Numerical simulations are addressed in Section 4, which describes the control with and without disturbances.

3.2. SMC

We have seen that CTC reduced the dynamic model of the manipulator to a set of homogeneous second-order differential equations that were not lumped. However, to achieve this, the robot parameters need to be fully known. Sometimes the parameters are not available or the system has disturbances in one or various DOFs, which affects the performance of CTC in trajectory tracking.

To deal with the problems mentioned above, SMC is a highly robust control that has been discussed in the literature. To apply this control scheme, we selected a surface as follows:

$$\mathbf{s} = \dot{\mathbf{e}}(t) + c\mathbf{e}(t) \tag{23}$$

where vector $\mathbf{s}$ contains all of the sliding surfaces in each DOF and c is a diagonal matrix with the constant c_k.

Using the following Lyapunov candidate function:

$$V = \frac{1}{2}\mathbf{s}^T\mathbf{s} \tag{24}$$

we rewrite Equation (9) as follows, including a distrubance $\mathbf{f}(t)$:

$$\ddot{\mathbf{q}} = M(\mathbf{q})^{-1}(\mathbf{u} - h(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{f}(t)) \tag{25}$$

The time derivative of the Lyapunov function was given by:

$$\dot{V} = \mathbf{s}^T\dot{\mathbf{s}} \tag{26}$$

Substituting the time derivative of $\mathbf{s}$ into (26) yielded:

$$\begin{aligned} \dot{V} &= \mathbf{s}^T\dot{\mathbf{s}} \\ &= \mathbf{s}^T[\ddot{\mathbf{q}}_d(t) - M(\mathbf{q})^{-1}(\mathbf{u} - h(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{f}(t)) + c\dot{\mathbf{e}}(t)] \end{aligned} \tag{27}$$

Applying the control $\mathbf{u} = \tau_o sign(\mathbf{s})$, where τ_o is a diagonal matrix of gains τ_{ok}, and substituting the control into (27) produced:

$$\begin{aligned}\dot{V} &= \mathbf{s}^T\dot{\mathbf{s}} \\ &= \mathbf{s}^T[\ddot{\mathbf{q}}_d(t) + c\dot{\mathbf{e}}(t) + M(\mathbf{q})^{-1}(h(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{f}(t))] \\ &\quad - \mathbf{s}^T M(\mathbf{q})^{-1}\tau_o sign(\mathbf{s})\end{aligned} \tag{28}$$

hence:

$$abs(\mathbf{s}^T) = \mathbf{s}^T sign(\mathbf{s}) \tag{29}$$

with:

$$abs(\mathbf{s}^T) = [|s_1||s_2|\dots|s_n|]^T \tag{30}$$

and:

$$abs(\mathbf{f}(t)) < \mathbf{L} \tag{31}$$

Equation (29) stated that each $f_i(t)$ had an upper bound constant of L_i. By combining Equations (29)–(31) into (28), we obtained the following inequality:

$$\begin{aligned}\dot{V} \leq abs(\mathbf{s}^T)[abs(\ddot{\mathbf{q}}_d(t) + c\dot{\mathbf{e}}(t) + M(\mathbf{q})^{-1}(h(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{L}) \\ - M(\mathbf{q})^{-1}\tau_0] < 0\end{aligned} \tag{32}$$

Equation (32) was definitely negative when the following was always true:

$$\tau_0 > abs[M(\mathbf{q})(\ddot{\mathbf{q}}_d(t) + c\dot{\mathbf{e}}(t)) + (h(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{L})] \tag{33}$$

In other words, when all the gains along the diagonal of the matrix τ_0 fulfilled the inequality, the system reached the surface and stayed there for the entire period of time t. As with CTC, the simulations are addressed in Section 4 to validate the robustness of the SMC.

4. Numerical Simulations of CTC and SMC

We showed in the previous section that we considered two types of control scheme (CTC and SMC). We showed via Lyapunov stability analysis that CTC was not robust against disturbances and that the dynamic model had to be completely known in order to achieve good performance. Meanwhile, the SMC could handle trajectory tracking without knowing the dynamic model in the control law. In addition to this, the SMC was robust against disturbances in the system.

This section presents the numerical simulations of the two control laws to better understand their behaviour. The actuator parameters are shown in Table 1.

Table 1. Actuator parameters.

Parameters	Actuators at the Arm	Actuators at the Wrist
$R_a(\Omega)$	5.54	5.3
$L_a(\mu\text{H})$	821	1052
$K_t(\frac{\text{Nm}}{\text{A}})$	0.0208	0.00363
$K_e(\frac{\text{Vs}}{\text{rad}})$	0.02076	0.00363
$J(\text{grcm}^2)$	19.4	0.0621
$B(\text{mNmms})$	0.875	0.7971

The parameters and a diagram of the manipulator are shown in Table 2 and Figure 1, respectively. The procedure that was used to obtain the dynamic model of the manipulator can be seen in [14].

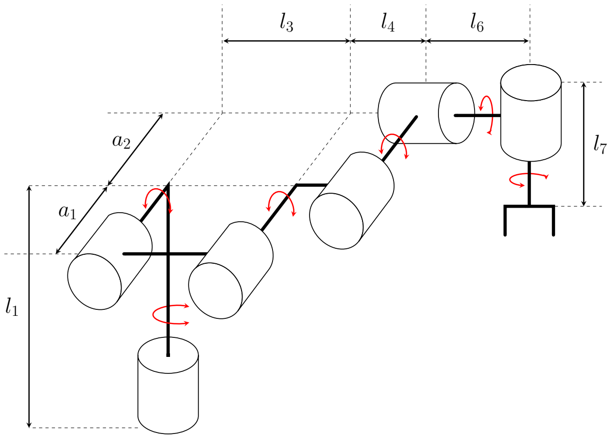

Figure 1. Diagram of the 6DOF manipulator.

Table 2. Manipulator parameters.

First Articulation	**Second Articulation**	**Third Articulation**
$a_1 = 6$ cm $a_{c1} = 3.036$ cm $m_1 = 0.11$ kg	$l_3 = 23$ cm $l_{c3} = 10.513$ cm $m_2 = 0.2125$ kg	$a_1 = 6$ cm $a_{c1} = 3.036$ cm $m_s = 0.5574$ kg
$I_{zz1} = 2.6 \times 10^{-6}\,\text{kgm}^2$	$I_{xx2} = 9.2 \times 10^{-4}\,\text{kgm}^2$ $I_{yy2} = 2.0515 \times 10^{-3}\,\text{kgm}^2$ $I_{zz2} = 2.9224 \times 10^{-3}\,\text{kgm}^2$	$I_{xxs} = 2.922 \times 10^{-6}\,\text{kgm}^2$ $I_{yys} = 2.922 \times 10^{-6}\,\text{kgm}^2$ $I_{zzs} = 5.615 \times 10^{-6}\,\text{kgm}^2$
		$l_4 = 24$ cm $l_{c4} = 4.310$ cm $m_3 = 0.215$ kg
		$I_{xx3} = 1.9229 \times 10^{-5}\,\text{kgm}^2$ $I_{yy3} = 2.1720 \times 10^{-4}\,\text{kgm}^2$ $I_{zz3} = 2.3177 \times 10^{-4}\,\text{kgm}^2$
Fourth Articulation	**Fifth Articulation**	**Sixth Articulation**
$a_2 = 10$ cm $a_{c2} = 5$ cm $m_4 = 0.0803$ kg	$l_6 = 4$ cm $l_{c6} = 2$ cm $m_5 = 0.1412$ kg	$l_7 = 10$ cm $lc_7 = 5$ cm $m_5 = 0.0712$ kg
$I_{xx4} = 4.7373 \times 10^{-5}\,\text{kgm}^2$ $I_{yy4} = 1.8376 \times 10^{-5}\,\text{kgm}^2$ $I_{zz4} = 4.7373 \times 10^{-5}\,\text{kgm}^2$	$I_{xx5} = 2.6153 \times 10^{-5}\,\text{kgm}^2$ $I_{yy5} = 3.9774 \times 10^{-5}\,\text{kgm}^2$ $I_{zz5} = 4.1205 \times 10^{-5}\,\text{kgm}^2$	$I_{xx6} = 7.3043 \times 10^{-5}\,\text{kgm}^2$ $I_{yy6} = 7.4434 \times 10^{-5}\,\text{kgm}^2$ $I_{zz6} = 6.86 \times 10^{-6}\,\text{kgm}^2$

For trajectory tracking, a Linear Segment Parabola Blending (LSPB) was used. This function was defined as follows:

$$qd = \begin{cases} q_0 + \frac{a}{2}t^2 & 0 < t \le t_b \\ \frac{q_f + q_0 + vt_f}{2} + vt & t_b < t \le t_f - tb \\ q_f - \frac{at_f^2 t}{2} + at_f t - \frac{a}{2}t^2 & t_f - t_b < t \le t_f \end{cases} \tag{34}$$

where q_d is the trajectory LSPB, q_0 and q_f are the initial and final values of the trajectory, t_b and t_f are the mixing and final times, and v and a are the velocity and acceleration of the trajectory, respectively. We could compute v and a using the values of q_0, q_f, t_f, and $t_b = \frac{1}{3}t_f$. The LSPB trajectory with $t_f = 4$, $q_0 = 0$, and $q_f = \pi$ is shown in Figure 2. In both the CTC and STM control simulations, a step perturbation and an LSPB trajectory were used. The t_f of each DOF was 4 s and the final degree of each DOF was:

$$\mathbf{q}_f = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{3} \\ -\frac{\pi}{3} \\ \frac{\pi}{4} \\ \frac{\pi}{3} \\ -\frac{\pi}{6} \end{bmatrix}$$

For CTC, a simulation without perturbations is shown in Figure 3 and a simulation with perturbations is shown in Figure 4.

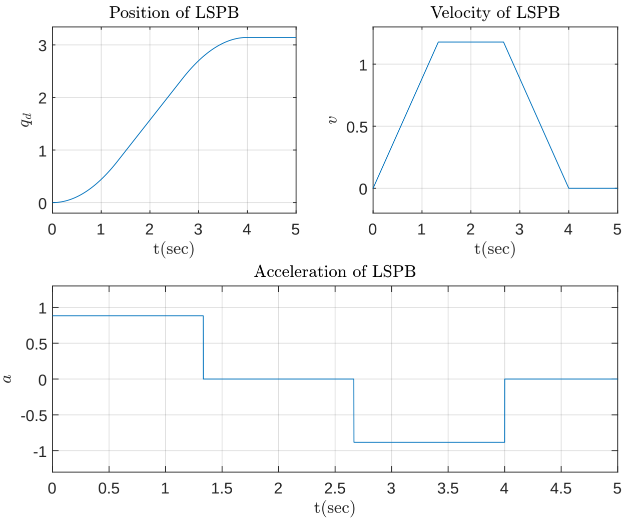

Figure 2. LSPB trajectory.

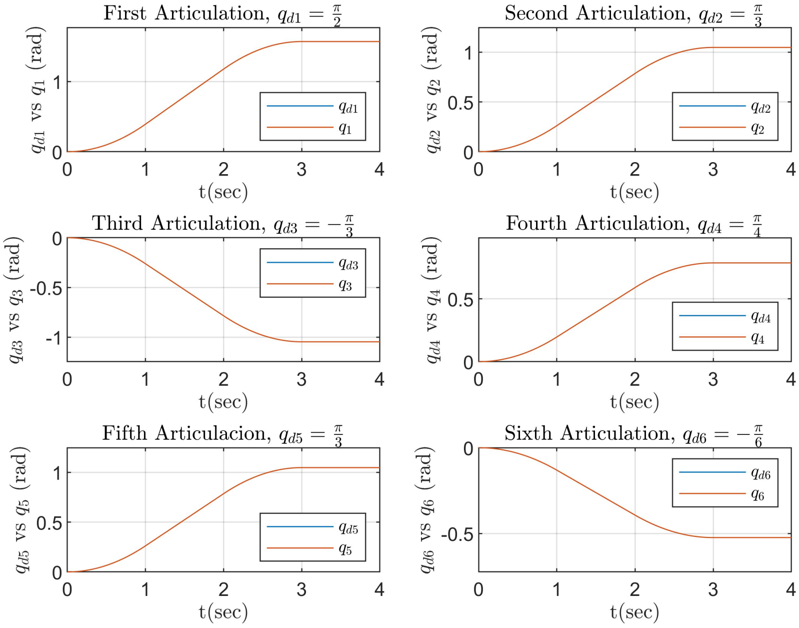

Figure 3. CTC numerical simulation with no disturbances.

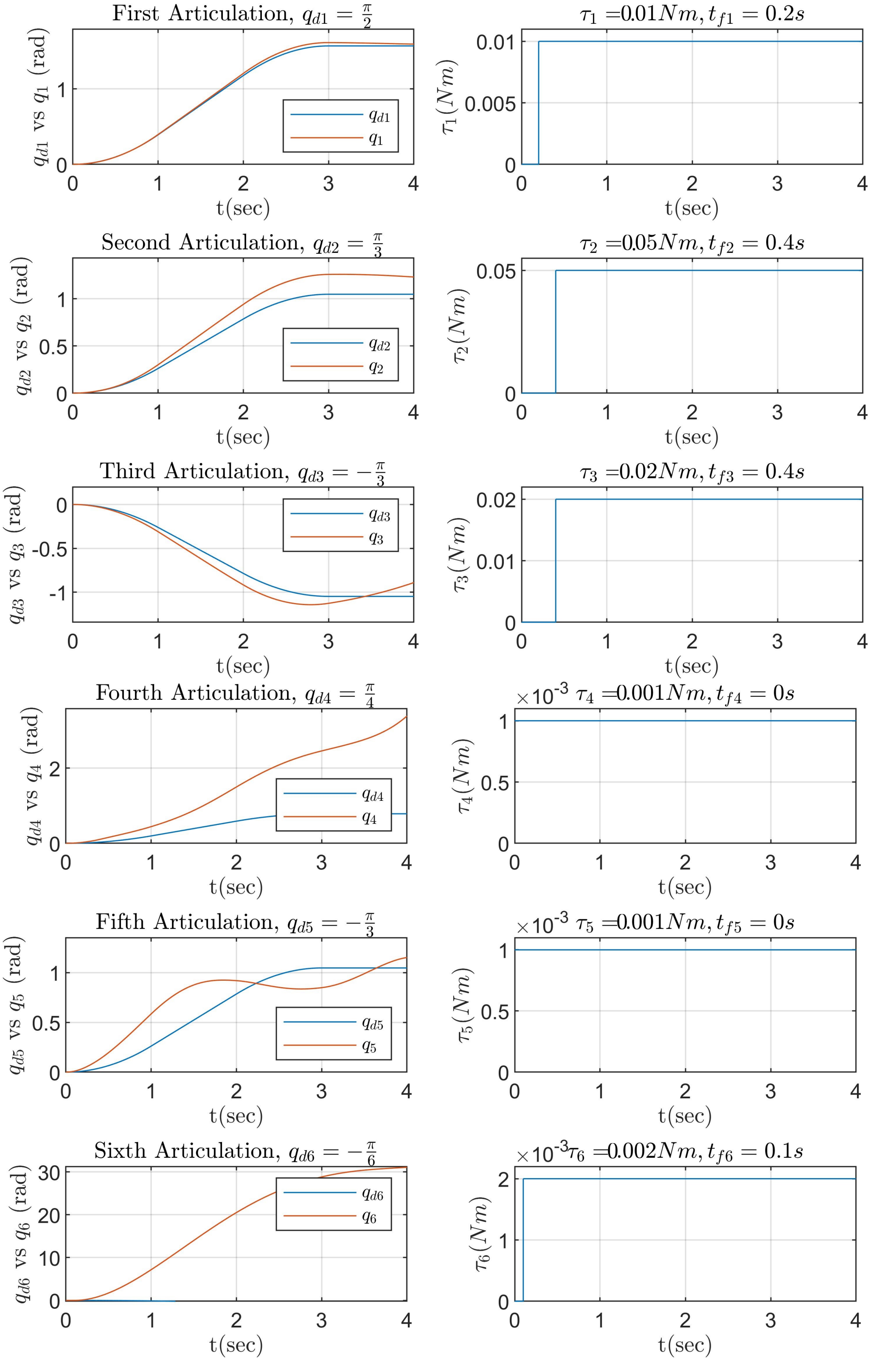

Figure 4. Numerical simulation of CTC with perturbations.

Each DOF on the manipulator followed the desired trajectory when there were no perturbations, as shown in Figure 3, but the robot lost the trajectory when we applied a perturbation in each DOF. As we showed in Section 3, the Lyapunov function could not have a definite sign when a perturbation was applied to the CTC, so there was no guarantee that stability existed. This can be seen in Figure 4. The SMC and its control are shown in Figures 5 and 6, respectively.

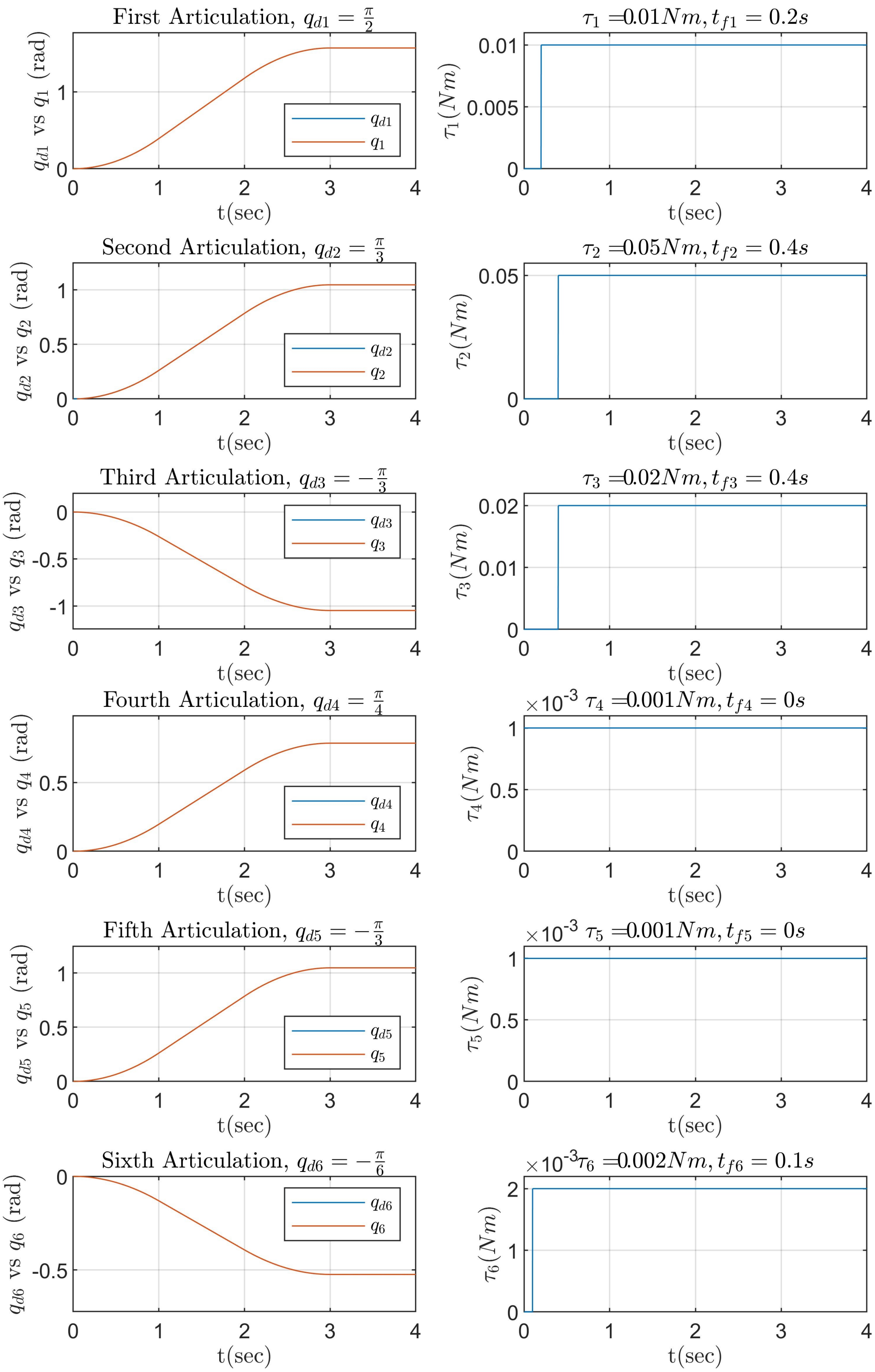

Figure 5. Simulation of the SMC with perturbations using numerical methods.

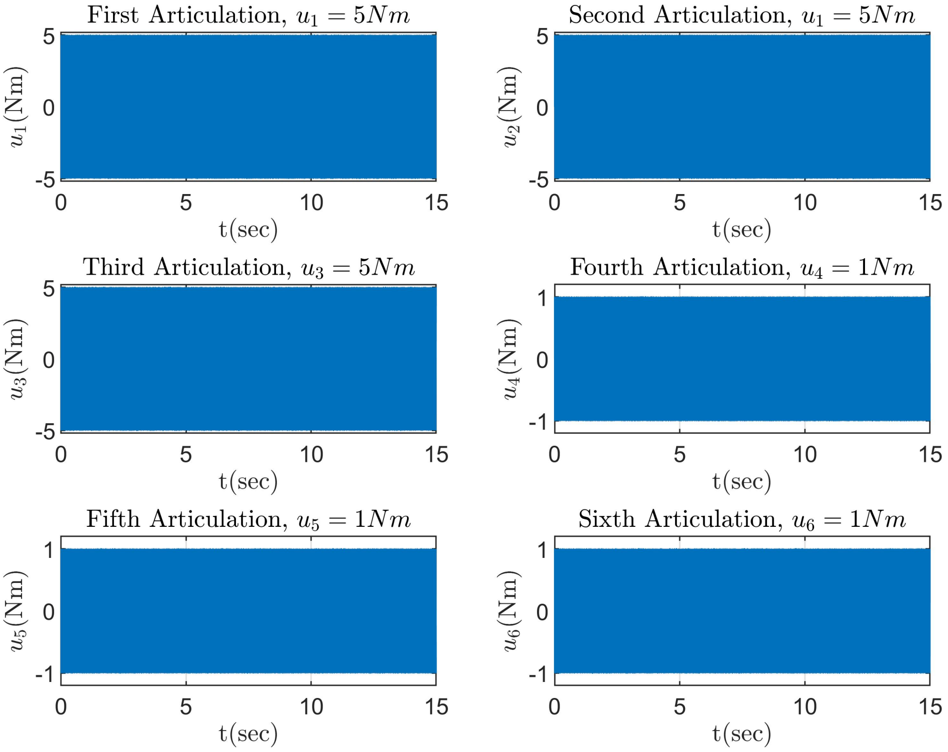

Figure 6. Simulation of the signal regulation of the SMC with perturbations using numerical methods.

Three different techniques to measure error (ITAE, IAE and ISE) in trajectory tracking were used to compare the two controllers with perturbations, which can be seen in Figure 7.

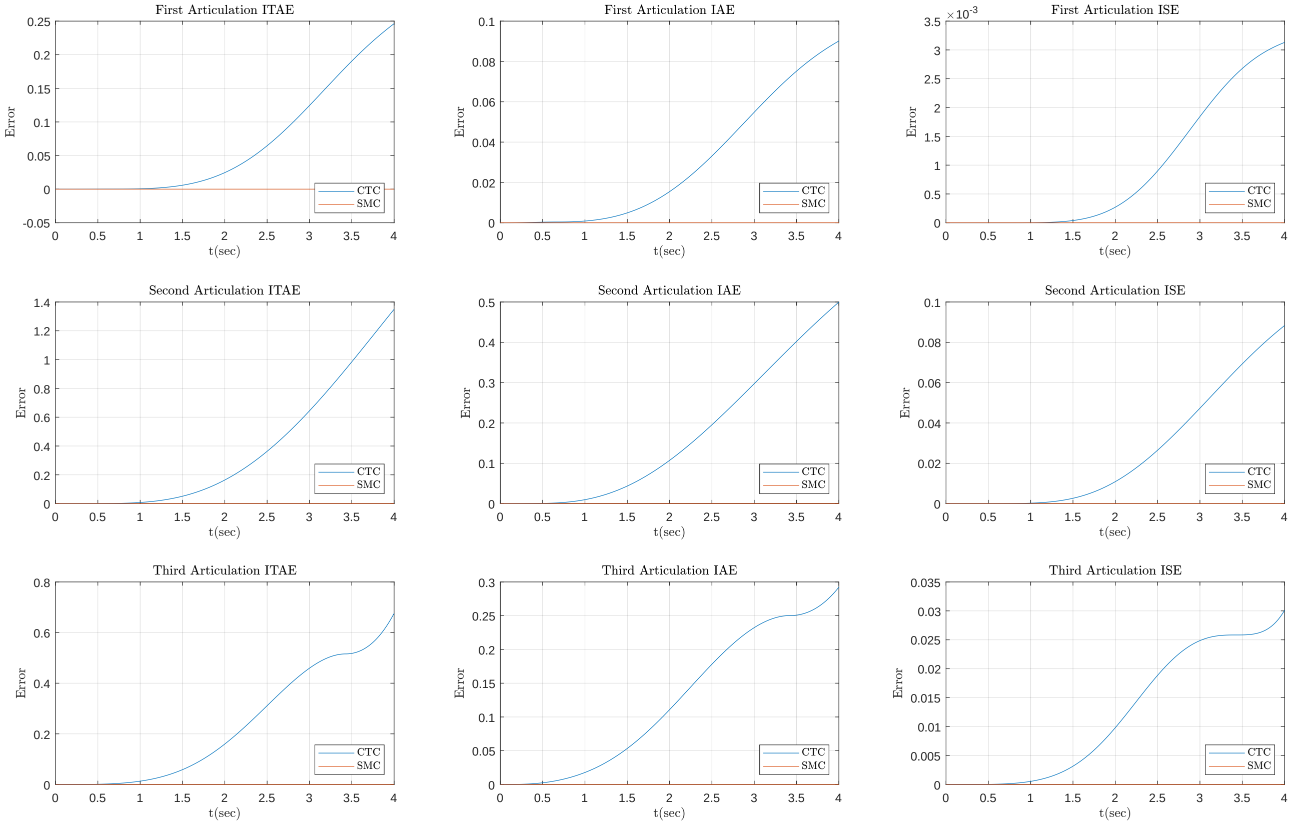

Figure 7. *Cont.*

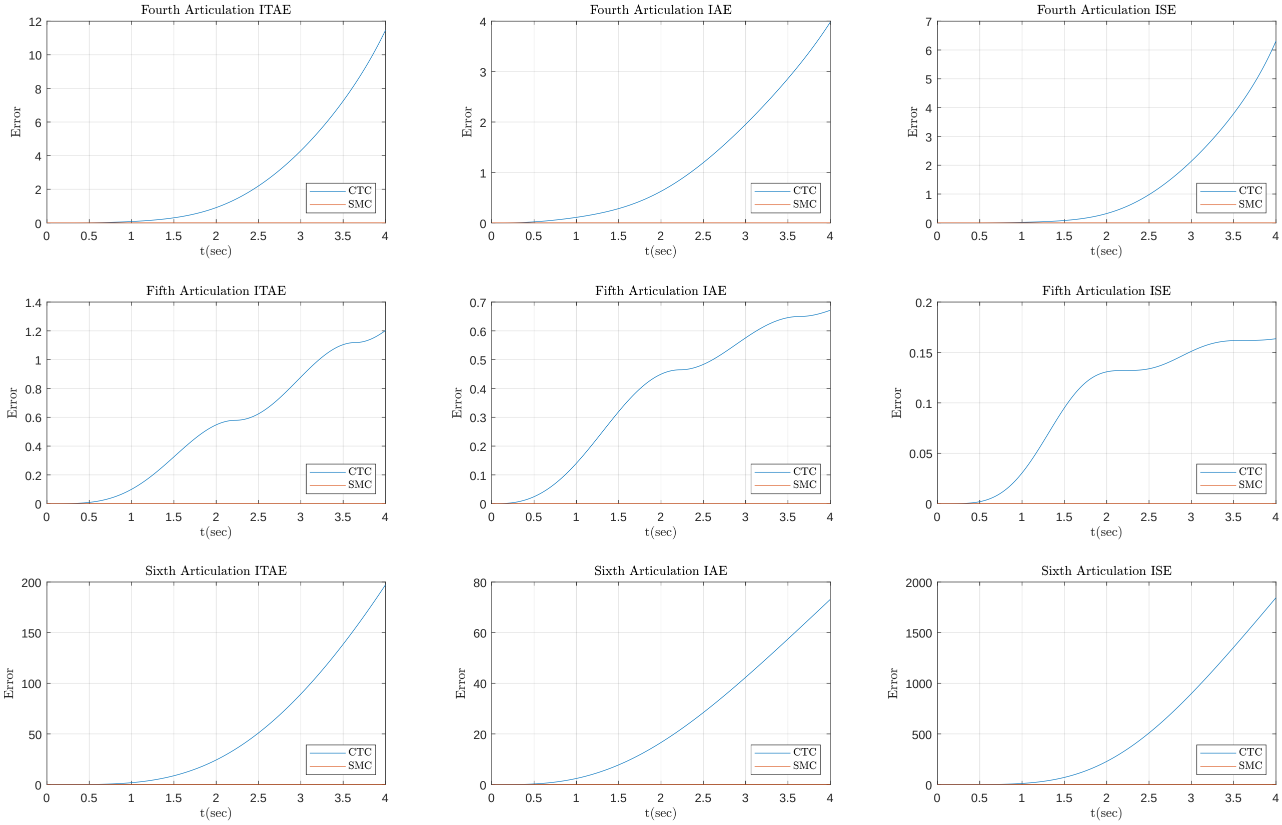

Figure 7. Error performance indexes: ITAE, IAE, and ISE.

It can be seen in Figure 5 that the manipulator followed the trajectory when we had enough gain in the SMC control, as demonstrated in Section 3, and that the time derivative of the Lyapunov function was always negative. However, it can be seen in Figure 6 that the effect of using a discontinuous controller caused the applied torque to oscillate within the manipulator.

The comparison between the SMC and CTC with perturbation can be seen in Figure 7. This comparison demonstrated the robustness of the SMC against perturbations. The CTC had a visible and high magnitude of error, but the SMC error was not visible. In the following section, an SMC was applied to a 6DOF manipulator using a self-developed electronic interface (SDEI) in MATLAB. The advantages and disadvantages of the proposed control scheme are also discussed.

5. Implementation and Experimental Results

As we showed in Section 4, SMC offered a simpler control law compared to CTC, which only needs to know the gain of the sign function. The numerical simulations showed that the SMC achieved asymptotic stability in the presence of perturbations. For these reasons, SMC was selected for the trajectory tracking of a 6DOF manipulator driven by DC motors. The manipulator was located at the "Laboratorio de Control de Electromecanismos" at the "Tecnológico de Chihuahua". The SDEI manipulator is shown in Figure 8.

To apply the SMC, we had to understand the control law in terms of the software and hardware that were being used. As the dynamic model of the manipulator with actuators had a voltage input, the sign function changed the polarity of the voltage in the DC motor. A H-bridge was in charge of this. Additionally, the voltage in the H bridge (power converter) was the gain in sliding mode. The H bridge that was selected for this study was an L298 motor drive, which could handle two DC motors simultaneously. Two systems were used for the experiment: the control system and the power system. The control system had the following main elements, each with its own communication protocols:

- SDEI (UART-SPI);
- 4 PIC18F26K20;
- FTDI FT232RL (USB-UART).

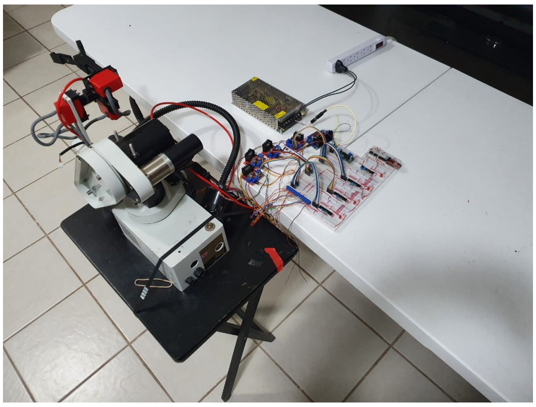

Figure 8. The 6DOF manipulator with SDEI.

The control system started in the computer, which contained the control algorithm within the MATLAB environment. MATLAB sent the information via a virtual serial port to the FTDI (USB), then the SDEI sent the information to the SEDI via the UART protocol. SEDI had four PIC18F26K20s, one of which was the master that received the data from FTDI and sent them to slaves via SPI. The slaves were in charge of generating the PWM that was needed for the power system and reading the position sensors of the serial manipulator. The slaves sent the data positions to the master and, using the same sequence as before, the data reached MATLAB. MATLAB computed the control law with the information obtained and sent it to the FTDI; then, the process was repeated again. It took 5 mili-seconds for the SDEI to complete the cycle, which was the sample time of the system. Diagrams of the experimental setup and the SDEI configuration are shown in Figure 9.

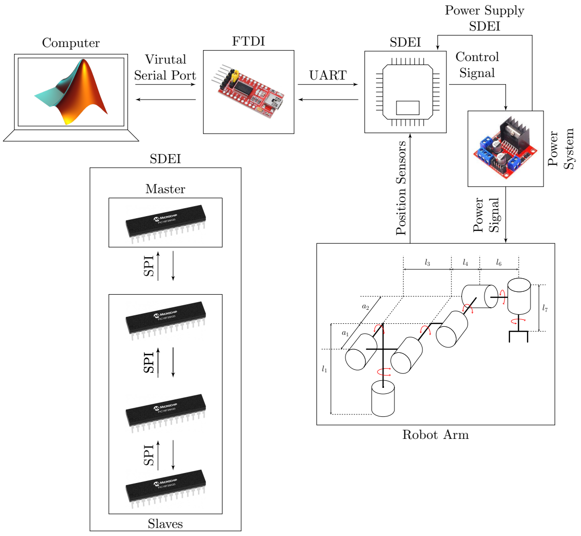

Figure 9. SDEI diagram.

The power system was responsible for supplying the voltage to the SDEI, the H bridge, and the power converters. The elements of the power system were the following:

- Power supply (12 Volts);
- Four H-bridges;
- DC–DC converter (12 Volts − 5 Volts);
- DC–DC converter (12 Volts − 3.3 Volts).

The DC–DC converters were connected to the power supply of 12 Volts. One of them reduces the voltage to 3.3 V to power the SDEI. The other reduces the voltage to 5 V to power the 2 H-bridges. The 2 H bridges with 5 V are connected to wrist motors, while the 2 other H bridges are connected to the power supply (12 V). These H-bridges are in charge of the power of the arm motors. A diagram of the power system is presented in Figure 10.

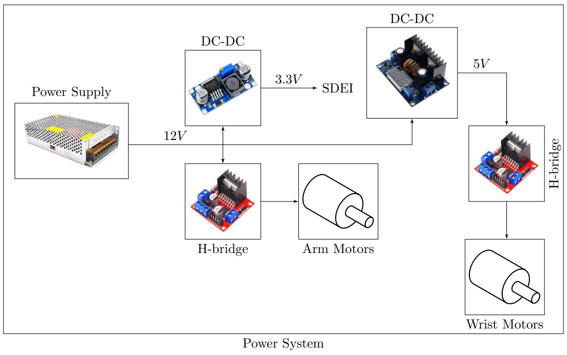

Figure 10. The power system.

With SDEI, communication was achieved between MATLAB and the manipulator. The control algorithm (SMC) and its reference, the LSPB trajectory in this case, were programmed in MATLAB. Figure 11 shows the control diagram.

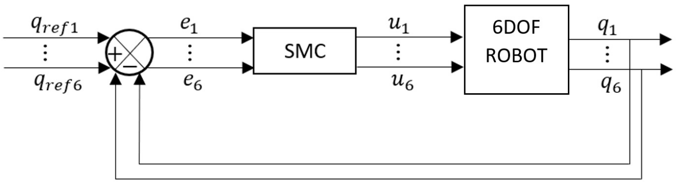

Figure 11. The control diagram.

The physical arm only communicated with the position sensor, which meant that we could only measure the position error, but we needed both the position and velocity errors for the sliding surface. Due to this, a backward Euler method was used as a differentiator to estimate the velocity error. To accomplish this, we use the following.

$$\dot{\mathbf{e}} = \frac{\mathrm{d}e(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{e(t) - e(t - \Delta t)}{\Delta t} \tag{35}$$

where Δt is the sample time and $e(t - \Delta t)$ is the previous error value. As the control was implemented in a digital environment, the limit of (35) could not reach zero and (35) was approximated in the following way:

$$\dot{\mathbf{e}} \approx \frac{e(t) - e(t - \Delta t)}{\Delta t} \tag{36}$$

In other words, when the sample time was shorter, Equation (36) obtained a better approximation of Equation (35). We needed to know whether the voltage in each DC motor could follow the desired trajectory. As mentioned above, these voltages were the sliding mode gains in voltage terms. Using a numerical simulation, we graphed Equation (33) and the maximum of this graph was the torque required to reach the surface and stay there for the entire period of time. Since we did not know how much torque was needed, the simulation started with a gain of 5 nm for the arm motors and 1 nm for the wrist motors. For the physical experiment, each DOF followed an LSPB. The final times of the LSPB were $\mathbf{t}_f = \begin{bmatrix} 10 & 12 & 9 & 6 & 8 & 7 \end{bmatrix}$ and the final degrees of the LSPB are $\mathbf{q}_f = \begin{bmatrix} \frac{\pi}{3} & \frac{\pi}{2} & \frac{\pi}{4} & \frac{\pi}{3} & \pi & \frac{\pi}{2} \end{bmatrix}$. Figures 12–14 present these results.

It can be seen in Figure 12 that the manipulator followed the desired trajectory, which meant that the correct gains were used. To obtain the minimum required gain, we calculated the maximum of each torque (L_1 to L_6) using the Lyapunov analysis shown in Figure 14 and converted it into voltage using Equation (8). With these taken into account, the maximum torques τ_0, the minimum required voltages V_0, and the voltage in each motor V_m are shown in Table 3.

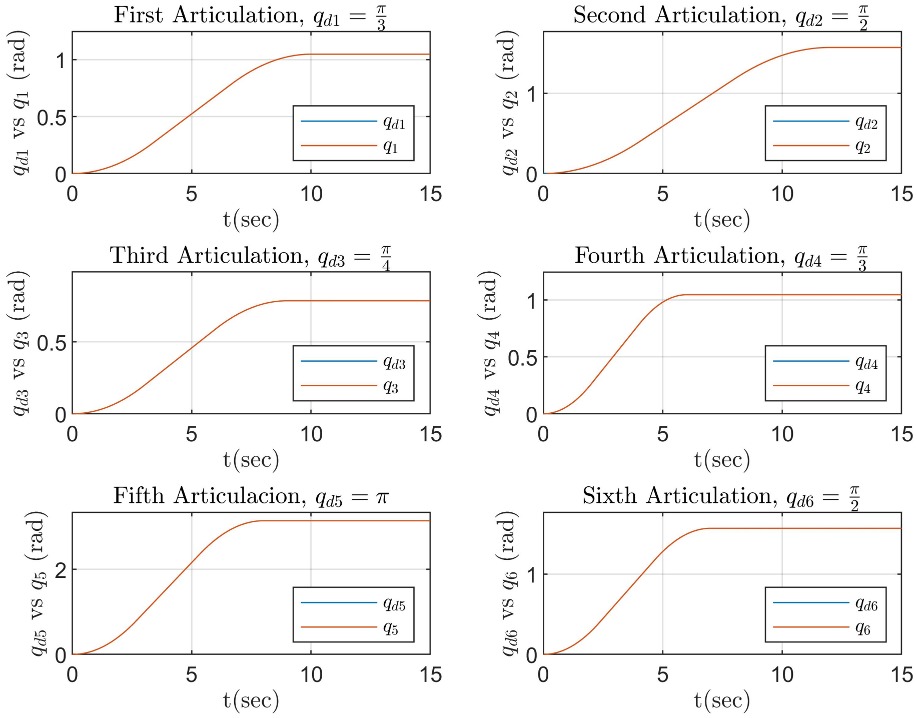

Figure 12. Trajectory tracking simulation of the applied SMC.

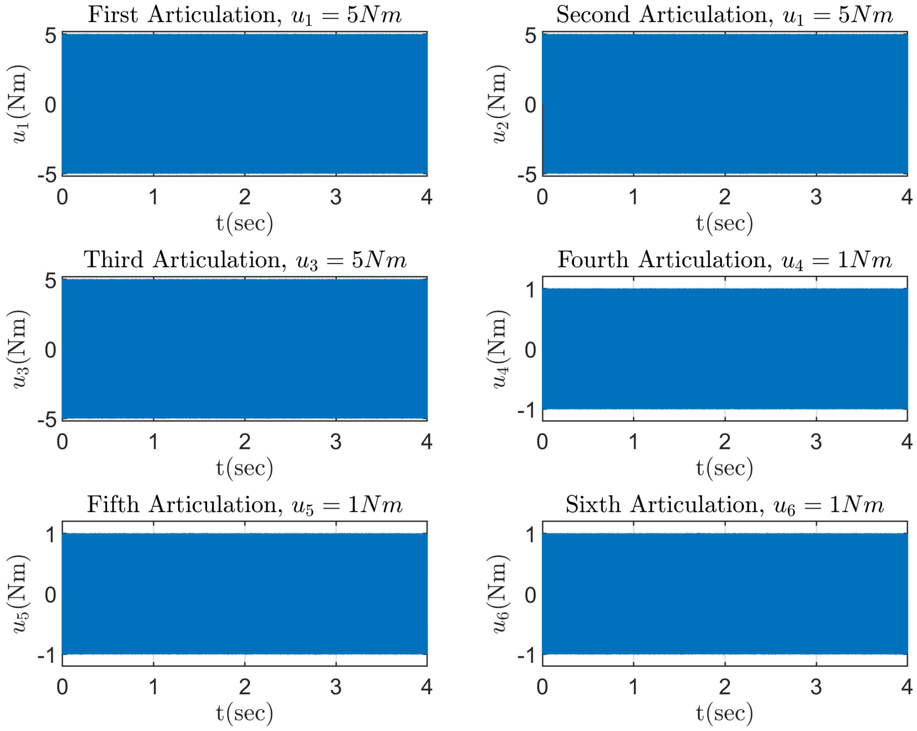

Figure 13. Input signal control.

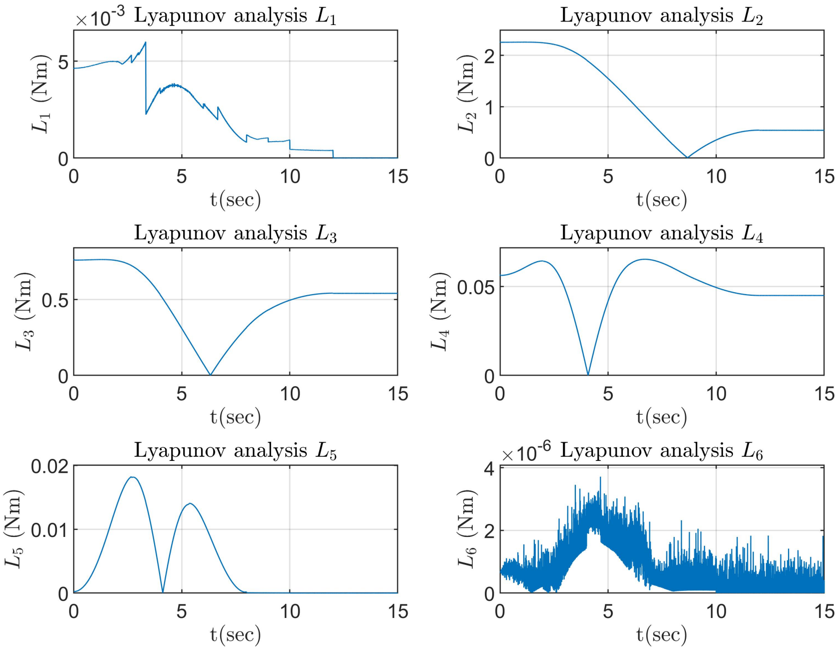

Figure 14. The signal of the Lyapunov analysis.

Table 3. Comparison between the voltage that was needed to follow the trajectory and the voltage of the electronic interface.

DOF	τ_0 (nm)	V_0 (Volts)	V_m (Volts)
1	0.0060	0.2665	12
2	2.2668	11.6858	12
3	0.7644	10.1807	12
4	0.0652	4.1253	5
5	0.0182	1.1529	5
6	3.71×10^{-6}	2.35×10^{-3}	5

When we compared the voltage in each motor V_m to the minimum required voltage (sliding-mode gain in voltage terms) V_0, $V_0 < V_m$. Therefore, the voltage in each motor was sufficient to track the LSPB trajectory. In this manner, the verification of the sliding mode was completed, and a physical application was performed. Figures 15 and 16 show the results of the physical implementation. As expected, the gain of the SMC was responsible for all tracking of the trajectories because the dynamic model of the manipulator (the known part of the ECM) was not in the control law. The consequence was a high magnitude discontinuous control law, as shown in Figure 16, which caused each DOF of the robot arm to vibrate (chatter) (see Figure 15). As can be seen in Figure 15, the physical arm followed the proposed trajectory of each DOF, as in the simulation. Thus, the voltage in each motor was sufficient. According to this result, we could validate the sliding mode gain using the Lyapunov stability analysis. This solved the problem of using iterative methods in the design of the sliding gain. A verification of this gain before this application has not been shown before in the literature. In this sense, the gain could be selected without increasing the gain in the experiments or simulations. The following advantages of the control strategy proposed in this paper were:

- The desired trajectory was perfectly followed without knowing the parameters of the manipulator;
- The SMC was a simple control strategy compared to CTC. The CTC had the dynamic model of the manipulator in its control law, which made it a complex law due to the 6DOF of the robot. For the SMC, we only had to know the sign of the surface, which did not depend on any parameter or perturbation of the manipulator. In

this way, the SMC programming was less complicated and performed better in the software environment;

- The gain validation of the SMC developed in this paper could be used as a straightforward tuning procedure. As mentioned above, the main drawback of the well-known PID controller is the tuning process. This made the SMC a better option in terms of selecting the gains of the controllers;
- There was no need to obtain the gain of the sliding mode using iterative methods in simulations or experiments.

However, even though the classic SMC had some great advantages, the following disadvantages were seen in the controller strategy when it was implemented:

- As mentioned in the literature, the chattering problem was present in the movement of the manipulator due to the discontinuous law. This could create fatigue in the actuators and the mechanical parts of the robot;
- A discontinuous law was implemented, which used all the gain in power of the electronic interface and developed an inefficient use of energy in the control law.

The most significant result of this paper was the gain validation of the SMC for its application in the experimental results. Furthermore, Lyapunov analysis was used as a simple tuning method for the gain of the SMC, which was a great advantage for this type of controller. However, the chattering phenomenon was present in robotic trajectory tracking. One of the reasons for the chattering in the manipulator was the numerical differentiator. In all of the simulations, little chattering was present because the velocity was taken directly from the model. However, in the implementation, the sample time of the SDEI induced numerical errors in the sliding surface. For better performance and a reduction in chattering, enhanced sliding mode controllers or hybrid sliding mode controllers need to be used. Similarly, when the classical SMC was applied with the numerical differentiator used in this paper, electronic interfaces with lower sample times had to be used. The approach for advanced differentiators could also reduce the chattering phenomenon.

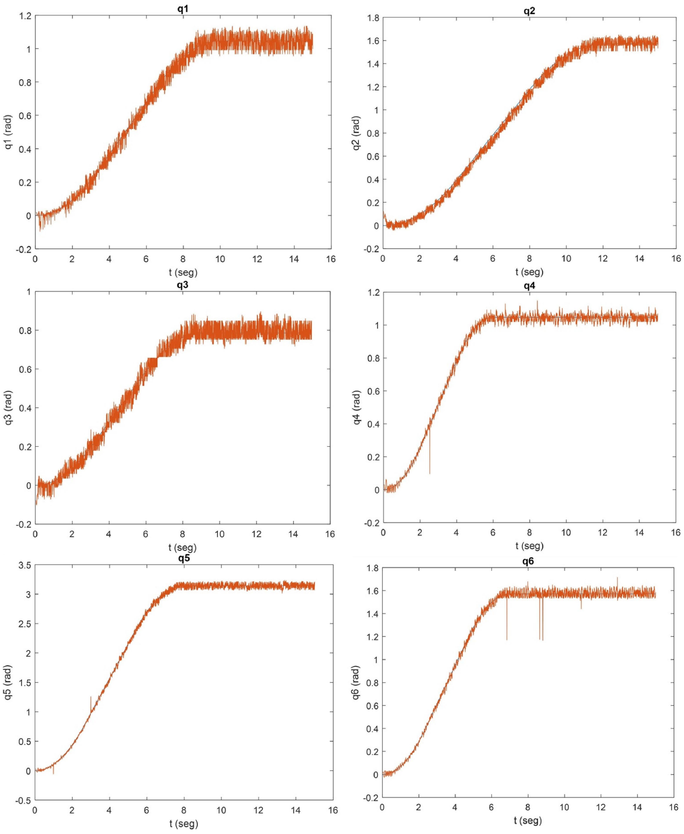

Figure 15. Tracking of the physical robot.

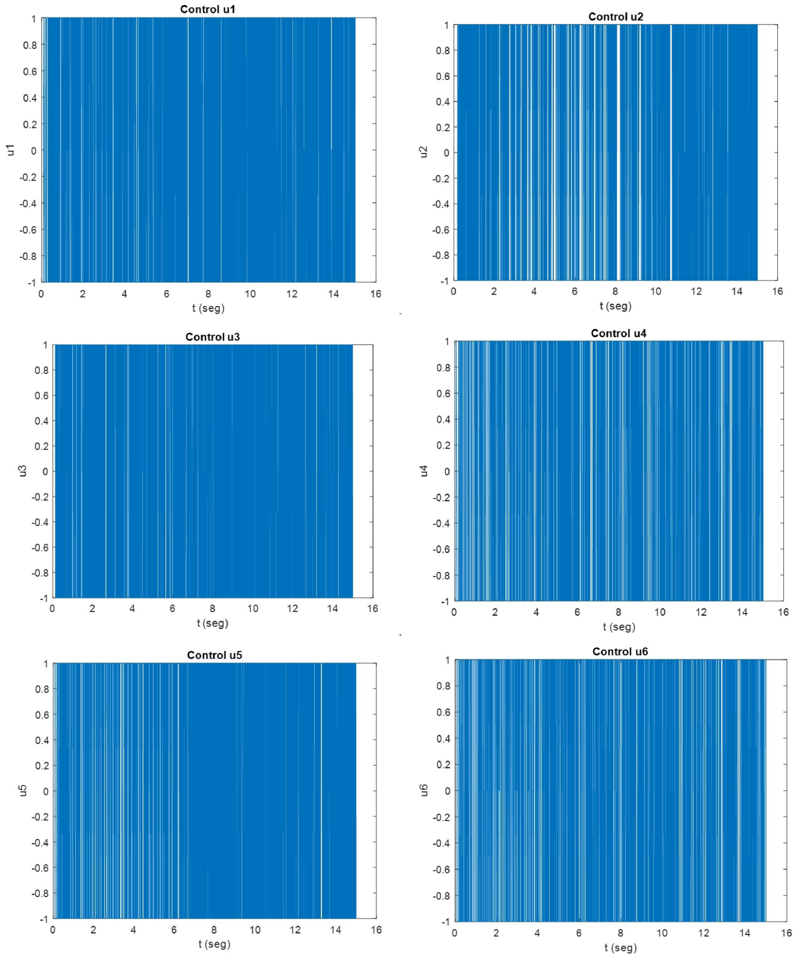

Figure 16. Signal of control applied to the physical manipulator.

6. Conclusions

In this article, the calculation of the SMC gain was performed using Lyapunov analysis and validated in a real application using a DC motor driven serial manipulator. The experimental results showed great performance in trajectory tracking with the selected gain. As we showed in Section 3, the SMC was a simple strategy compared to CTC, for which the dynamic model needed to be known a priori, and a 6DOF robot was too complex. Furthermore, in the numerical simulations, the CTC was not robust against perturbations and lost the trajectory. However, the discontinuous control law in the SMC generated chattering in the manipulator, which could cause damage to the actuators and mechanical parts. For this reason, in the future advanced SMC strategies that deal with the chattering problem must be taken into account.

Author Contributions: Conceptualization, I.R.-V., A.G.-R. and R.E.B.-A.; methodology, A.G.-R. and R.E.B.-A.; software, A.G.-R. and R.E.B.-A.; validation, A.G.-R. and R.E.B.-A.; formal analysis, A.G.-R., A.E.R.-M. and R.E.B.-A.; investigation, A.G.-R., A.E.R.-M. and R.E.B.-A.; writing—original draft preparation, A.G.-R., P.R.A.C.d.l.R., I.R.-V. and R.E.B.-A.; writing—review and editing, A.G.-R., P.R.A.C.d.l.R., I.R.-V. and R.E.B.-A.; supervision, A.G.-R., P.R.A.C.d.l.R., I.R.-V. and R.E.B.-A.; project administration, R.E.B.-A.; funding acquisition, R.E.B.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by the project thanks to the "Brazo manipulador con morfología adaptada a una plataforma móvil robotizada tipo industrial." TENM key: 6050.17-P.

Data Availability Statement: Not applicable.

Acknowledgments: The Tecnológico Nacional de México campus Instituto Tecnológico de Chihuahua is thanked for the economic funding "Brazo manipulador con morfología adaptada a una plataforma móvil robotizada tipo industrial." key: 6050.17-P. This work is grateful to CONACYT for supporting the first author with a scholarship.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CTC	Computed torque control
SMC	Sliding mode control
DOF	Degrees of freedom
ECM	Equivalent control method
NSMC	Novel sliding control method
EGWO	Extended gray wolf optimizer
SOSMC	Second-order sliding mode control
RBFN	Radial basis function network
DAM	Dual arm
MRAC	Model reference adaptive control
SPO	Sliding perturbation observer
STA	Super-twisting algorithm
ASMNN	Adaptive sliding mode neural network control
RBFNN	Radial basis function neural network
TDE	Time delay estimation
AISMC	Adaptive integral sliding mode control
6DOF	6 degrees of freedom
UAV	Unmanned aerial vehicle
ETC	Electrical time constant
MTC	Mechanical time constant
LSPB	Linear segment parabola blending
SDEI	Self-developed electronic interface

References

1. Spong, M.W.; Vidyasagar, M. *Robot Dynamics and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
2. Kelly, R. A tuning procedure for stable PID control of robot manipulators. *Robotica* **1995**, *13*, 141–148. [CrossRef]
3. Kelly, R.; Davila, V.S.; Perez, J.A.L. *Control of Robot Manipulators in Joint Space*; Springer Science & Business Media: Berlin, Germany, 2006.
4. Liu, M. Computed torque scheme based adaptive tracking for robot manipulators. In Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, 21–27 May 1995; Volume 1, pp. 587–590.
5. Llama, M.; Kelly, R.; Santibanez, V. Stable computed-torque control of robot manipulators via fuzzy self-tuning. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2000**, *30*, 143–150. [CrossRef] [PubMed]
6. Jiang, J.; Cao, S.; Dai, Y. Research on RBF neural network model compensation and adaptive control of robot manipulators. In Proceedings of the 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, China, 28–30 May 2016; pp. 516–520.
7. Shtessel, Y.; Edwards, C.; Fridman, L.; Levant, A. *Sliding Mode Control and Observation*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 10.
8. Hu, Q.; Zhang, D.; Wu, Z. Trajectory planning and tracking control for 6-DOF Stanford manipulator based on adaptive sliding mode multi-stage switching control. *Int. J. Robust Nonlinear Control.* **2021**, *31*, 6602–6625. [CrossRef]
9. Utkin, V. Sliding mode control design principles and applications to electric drives. *IEEE Trans. Ind. Electron.* **1993**, *40*, 23–36. [CrossRef]
10. Haddad, S.Q.G.; Akkar, H.A. Intelligent swarm algorithms for optimizing nonlinear sliding mode controller for robot manipulator. *Int. J. Electr. Comput. Eng.* **2021**, *11*, 3943–3955. [CrossRef]
11. Pérez-San Lázaro, R.; Salgado, I.; Chairez, I. Adaptive sliding-mode controller of a lower limb mobile exoskeleton for active rehabilitation. *ISA Trans.* **2021**, *109*, 218–228. [CrossRef] [PubMed]
12. Lei, Y.l.; Wen, G.; Fu, Y.; Li, X.; Hou, B.; Geng, X. Trajectory-following of a 4WID-4WIS vehicle via feedforward–backstepping sliding-mode control. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2022**, *236*, 322–333. [CrossRef]
13. Utkin, V.; Guldner, J.; Shi, J. *Sliding Mode Control in Electro-Mechanical Systems*; CRC Press: Boca Raton, FL, USA, 2017.
14. González, A.; Baray, R. Desarrollo del modelo dinámico de un brazo articulado de seis grados de libertad con comprobación. *Congr. Int. en Ing. Electrónica. Mem. ELECTRO* **2020**, *42*, 130–139.

15. Bailey, E.; Arapostathis, A. Simple sliding mode control scheme applied to robot manipulators. *Int. J. Control.* **1987**, *45*, 1197–1209. [CrossRef]
16. Chaturvedi, N.K.; Prasad, L. A comparison of computed torque control and sliding mode control for a three link robot manipulator. In Proceedings of the 2018 International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 28–29 September 2018; pp. 1019–1024.
17. Chu, X. Two degrees of freedom Cooperative Suspension Control for Maglev Wind Yaw System. *IEEE/ASME Trans. Mechatron.* **2021**. [CrossRef]
18. Gutiérrez-Oribio, D.; Mercado-Uribe, J.A.; Moreno, J.A.; Fridman, L. Robust global stabilization of a class of underactuated mechanical systems of two degrees of freedom. *Int. J. Robust Nonlinear Control.* **2021**, *31*, 3908–3928. [CrossRef]
19. Rahmani, M.; Komijani, H.; Rahman, M.H. New sliding mode control of 2-DOF robot manipulator based on extended grey wolf optimizer. *Int. J. Control. Autom. Syst.* **2020**, *18*, 1572–1580. [CrossRef]
20. Tuan, L.A.; Joo, Y.H.; Tien, L.Q.; Duong, P.X. Adaptive neural network second-order sliding mode control of dual arm robots. *Int. J. Control. Autom. Syst.* **2017**, *15*, 2883–2891. [CrossRef]
21. Tuan, L.A.; Joo, Y.H.; Duong, P.X.; Tien, L.Q. Parameter estimator integrated-sliding mode control of dual arm robots. *Int. J. Control. Autom. Syst.* **2017**, *15*, 2754–2763. [CrossRef]
22. Chen, K.Y. Robust optimal adaptive sliding mode control with the disturbance observer for a manipulator robot system. *Int. J. Control. Autom. Syst.* **2018**, *16*, 1701–1715. [CrossRef]
23. Utkin, V.I. *Sliding Modes in Control and Optimization*; Springer Science & Business Media: Berlin, Germany, 2013.
24. Kallu, K.D.; Jie, W.; Lee, M.C. Sensorless reaction force estimation of the end effector of a dual-arm robot manipulator using sliding mode control with a sliding perturbation observer. *Int. J. Control. Autom. Syst.* **2018**, *16*, 1367–1378. [CrossRef]
25. Jeong, C.S.; Kim, J.S.; Han, S.I. Tracking error constrained super-twisting sliding mode control for robotic systems. *Int. J. Control. Autom. Syst.* **2018**, *16*, 804–814. [CrossRef]
26. Yen, V.T.; Nan, W.Y.; Van Cuong, P. Robust adaptive sliding mode neural networks control for industrial robot manipulators. *Int. J. Control. Autom. Syst.* **2019**, *17*, 783–792. [CrossRef]
27. Truong, L.V.; Huang, S.D.; Yen, V.T.; Van Cuong, P. Adaptive trajectory neural network tracking control for industrial robot manipulators with deadzone robust compensator. *Int. J. Control. Autom. Syst.* **2020**, *18*, 2423–2434. [CrossRef]
28. Fei, J.; Wang, H.; Fang, Y. Novel Neural Network Fractional-Order Sliding-Mode Control With Application to Active Power Filter. *IEEE Trans. Syst. Man, Cybern. Syst.* **2021**. [CrossRef]
29. Baek, J.; Jin, M.; Han, S. A new adaptive sliding-mode control scheme for application to robot manipulators. *IEEE Trans. Ind. Electron.* **2016**, *63*, 3628–3637. [CrossRef]
30. Lee, J.; Chang, P.H.; Jin, M. Adaptive integral sliding mode control with time-delay estimation for robot manipulators. *IEEE Trans. Ind. Electron.* **2017**, *64*, 6796–6804. [CrossRef]
31. Kim, H.H.; Lee, M.C.; Kyung, J.H.; Do, H.M. Evaluation of Force Estimation Method Based on Sliding Perturbation Observer for Dual-arm Robot System. *Int. J. Control. Autom. Syst.* **2021**, *19*, 1–10. [CrossRef]
32. Dumlu, A. Design of a fractional-order adaptive integral sliding mode controller for the trajectory tracking control of robot manipulators. *Proc. Inst. Mech. Eng. Part I J. Syst. Control. Eng.* **2018**, *232*, 1212–1229. [CrossRef]
33. Mofid, O.; Mobayen, S.; Wong, W.K. Adaptive terminal sliding mode control for attitude and position tracking control of quadrotor UAVs in the existence of external disturbance. *IEEE Access* **2020**, *9*, 3428–3440. [CrossRef]
34. Liu, K.; Wang, Y.; Ji, H.; Wang, S. Adaptive saturated tracking control for spacecraft proximity operations via integral terminal sliding mode technique. *Int. J. Robust Nonlinear Control.* **2021**, *31*, 9372–9396. [CrossRef]