

Modeling, Simulation and Control of a Redundant SCARA-Type Manipulator Robot

Regular Paper

Claudio Urrea* and John Kern

Departamento de Ingeniería Eléctrica, DIE
Universidad de Santiago de Chile, USACH* Corresponding author E-mail: claudio.urrea@usach.cl

Received 11 Apr 2012; Accepted 15 May 2012

DOI: 10.5772/51701

© 2012 Urrea and Kern; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract In this paper, the modeling of a redundant SCARA-type manipulator robot with five degrees of freedom is presented. We propose three controllers - hyperbolic sine-cosine, sliding mode, and calculated torque - which are applied to the discussed model. A simulation environment is developed using MatLab/Simulink programming tools. This simulation environment is employed to perform several tests (including actuators' dynamics) on the model of the redundant manipulator, with each different controller, under path tracking requirements. Results were obtained from comparative curves and *rms* index for joints and Cartesian errors.

Keywords Redundant manipulators, dynamic model, controllers, simulation.

1. Introduction

The widespread utilization of manipulator robots during the current stage of industrial development has led to productivity enhancements and quality improvements in

manufactured products - mainly due to the better repeatability of robot movements, which has consequently resulted in a higher accuracy in their performance. Although the first applications of such manipulators appeared in painting and welding processes, the automotive industry soon started to use them, expanding their field of application to the manipulation of bodyworks, engines, chassis and other components [1-5]. This required an increase in the flexibility of their workspace, a feature that can be achieved by increasing the number of degrees of freedom in the robots, i.e., making them redundant. Nevertheless, none of these activities would be possible without the adequate design of robots along with proper control techniques. To fulfil this purpose, knowledge and the study of a mathematical model and a certain kind of "intelligence" is required to govern the manipulator in order to perform the assigned tasks. By employing the basic physical laws ruling robot dynamics, it is possible to derive a model representing its behaviour and, by means of proper programming tools, to develop a simulation environment for subjecting the manipulator to different tests, such as path tracking.

In this paper, we present the modelling of a SCARA-type redundant manipulator robot with five degrees of freedom and, furthermore, we subject the resulting model to a path tracking test consisting of a spiral in Cartesian space. Three controllers are elaborated for testing the model: hyperbolic sine-cosine, sliding mode and calculated torque. A simulator is developed by means of the MatLab/Simulink *software* to run the redundant robot model with each controller. Actuator dynamics are also included in this analysis. Results are obtained from comparative curves and the *rms* index of joints and Cartesian errors.

2. Redundant Robots

Redundant robots are a kind of manipulator which has more degrees of freedom than the number required to perform some specific task [6, 7]. During recent years, special attention has been paid to the study of these manipulators, since redundancy has been considered a major feature for performing tasks requiring a level of dexterity comparable to the human arm, as in space technology applications such as the Special Purpose Dexterous Manipulator (SPDM), an essential component of the *Canadarm-2*: Robotic Arm designed by Canada for the International Space Station, shown in Fig. 1.



Figure 1. Special Purpose Dexterous Manipulator.

Even if the majority of non-redundant manipulators have enough degrees of freedom to perform their main tasks, for example, position and/or orientation tracking, it is known that their restricted manipulability leads to a reduction of the work space, due to mechanical limitations of the joints and the presence of obstacles in this space. This problem has motivated researchers to study manipulators' behavior when adding more degrees of freedom (kinematic redundancy), permitting these systems to manage additional tasks defined by the user. Such tasks can be represented as kinematic functions, including not only the kinematic functions reflecting some desirable properties of the manipulator's behavior, like joints characteristics and obstacle avoidance, but also can be expanded to include dynamic yield measurements through the definition of functions in the robot's dynamic model, like impact force, control of inertia, etc. [8].

The studied robotic manipulator has two additional degrees of freedom, giving it redundancy for its rotational movement in plane x - y , as well as in its prismatic movement in z axis, as seen in Fig. 2.

It is possible to generalize kinematic redundancy in the plane x - y as is shown in Fig. 3, where the internal variable θ establishes the relative angle between two adjacent links, and to determine the position of the end effector in terms of Eq. (1) [9].

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n l_i \cos \psi_i & \sum_{i=1}^n l_i \sin \psi_i \end{bmatrix}^T, \quad (1)$$

where ψ_i corresponds to the link's absolute angle, given by Eq. (2):

$$\psi_i = \sum_{j=1}^i \theta_j. \quad (2)$$



Figure 2. Scheme of a robotic manipulator with rotational and prismatic redundancy.

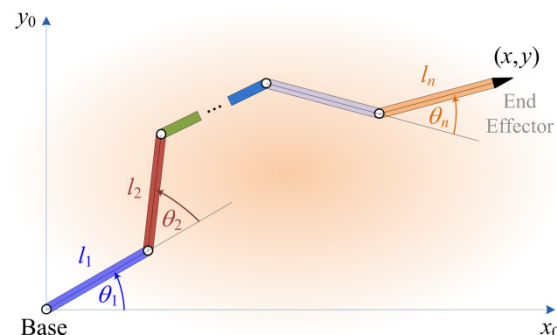


Figure 3. Scheme of a manipulator with generalized rotational redundancy in the plane x - y .

3. Scara-Type Redundant Manipulator

Fig. 4 shows a scheme of a SCARA-type redundant manipulator where we can see the redundancy in its rotational movement in the plane x - y as well as that in its

prismatic movement along the z axis and the distribution of the coordinate axes' systems and the position of centroids.

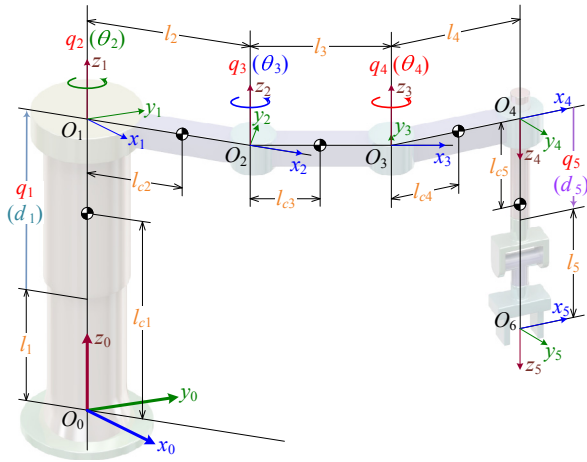


Figure 4. Scheme of a SCARA-type redundant manipulator.

Next, we proceed to perform the calculations corresponding to the manipulator's kinematic model.

3.1 Kinematics

In order to obtain the kinematic model, we have considered the standard Denavit-Hartenberg method, whose parameters are listed in Table 1.

Articulación i	θ_i	d_i	a_i	α_i
1	0°	$l_1 + d_1$	0	0°
2	θ_2	0	l_2	0°
3	θ_3	0	l_3	0°
4	θ_4	0	l_4	180°
5	0°	$l_5 + d_5$	0	0°

Table 1. Denavit-Hartenberg parameter assignment.

Then, applying the homogeneous transformations given by Eqs. (3) and (4), we obtain the direct kinematic model given by the matrix (5):

$${}^{i-1}\mathbf{T}_i = \text{Rot}(z_{i-1}, \theta_i) \cdot \text{Tras}(z_{i-1}, d_i) \cdot \text{Tras}(x_i, a_i) \cdot \text{Rot}(x_i, \alpha_i), \quad (3)$$

$${}^{i-1}\mathbf{T}_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

$$\mathbf{T} = \begin{bmatrix} c_{234} & s_{234} & 0 & l_2 c_2 + l_3 c_{23} + l_4 c_{234} \\ s_{234} & -c_{234} & 0 & l_2 s_2 + l_3 s_{23} + l_4 s_{234} \\ 0 & 0 & -1 & l_1 + d_1 - l_5 - d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5)$$

where:

$$s_2 = \sin\theta_2, \quad s_{23} = \sin(\theta_2 + \theta_3), \quad s_{234} = \sin(\theta_2 + \theta_3 + \theta_4),$$

$$c_2 = \cos\theta_2, \quad c_{23} = \cos(\theta_2 + \theta_3), \quad c_{234} = \cos(\theta_2 + \theta_3 + \theta_4).$$

In order to obtain inverse kinematics in a redundant robot we must consider different methods, selecting the most adequate one in accordance with model considerations. If we use homogeneous transformation matrices it is necessary to clear the n " q variables", in terms of vector components $\mathbf{n}$, $\mathbf{o}$, $\mathbf{a}$ and $\mathbf{p}$ corresponding to the direct kinematic model, as stated in expression (6):

$$\begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = [t_{ij}], \quad (6)$$

where the t_{ij} elements are functions of the joints coordinates $[q_1, \dots, q_n]^T$, and in this way it is possible to think that by means of some combinations of the 12 equations presented in (6) we can clear the n " q joints variables" in terms of the vector components $\mathbf{n}$, $\mathbf{o}$, $\mathbf{a}$ and $\mathbf{p}$. In most cases, this method can be quite tedious, appearing as transcendental equations. Nevertheless, if we consider three degrees of freedom, we can simplify the procedure in the following way:

$$\mathbf{T} = {}^0\mathbf{T}_1 \cdot {}^1\mathbf{T}_2 \cdot {}^2\mathbf{T}_3, \quad (7)$$

rearranging, we obtain:

$$\begin{aligned} ({}^0\mathbf{T}_1)^{-1} \mathbf{T} &= {}^1\mathbf{T}_2 \cdot {}^2\mathbf{T}_3 \\ ({}^1\mathbf{T}_2)^{-1} ({}^0\mathbf{T}_1)^{-1} \mathbf{T} &= {}^2\mathbf{T}_3, \end{aligned} \quad (8)$$

since $\mathbf{T}$ is known, the members to the left in expressions (8) are functions of the joints variables ($q_1, \dots, q_k$), while the members to the right are functions of the joints variables ($q_{k+1}, \dots, q_n$). In this way, it is possible to reduce the complexity when obtaining the joints variables.

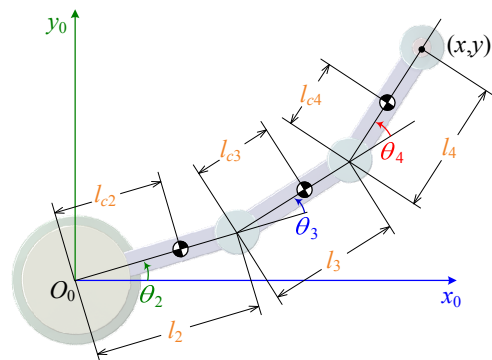


Figure 5. Scheme of the three rotational DOF in the SCARA-type redundant robot manipulator.

When applying this method to the three rotational degrees of freedom that rules the robot's movement in the plane x - y , as shown in Fig. 5, multiple solutions will be obtained. In order to overcome this limitation, we establish the condition stated by Eq. (9):

$$\theta_4 = \theta_3. \quad (9)$$

Accordingly with these considerations - and after adequate simplifications - the inverse kinematic model is obtained, expressed by:

$$\theta_3 = \pi \pm \arccos(\alpha), \quad (10)$$

$$\alpha = l_2(l_1 + l_3)/4l_1l_3 \pm \dots \quad (11)$$

$$\left((4l_1l_3 - l_2^2)(2l_1l_3 - l_1^2 - l_3^2) + 4l_1l_3(x^2 + y^2) \right)^{\frac{1}{2}} / 4l_1l_3,$$

$$\theta_2 = -2\arctan(\beta), \quad (12)$$

$$\beta = \frac{x \pm \left(x^2 + y^2 - s_3^2 \left(4l_3c_3(l_3c_3 + l_2) + l_2^2 \right) \right)^{\frac{1}{2}}}{y + l_2s_3 + l_3s_{2,3}}, \quad (13)$$

$$d_5 = l_1 + d_1 - l_5 - z, \quad (14)$$

where z and d_1 are known.

3.2 Dynamics

Having in mind the above presented manipulator, it is now necessary to obtain its dynamic model. For this purpose, we will employ the Lagrange-Euler formulation that is based on the principle of energy conservation [10]. Next, we need to obtain the manipulator's kinetic energy and potential energy. Therefore, to obtain the dynamic equation of the robotic manipulator, we must determine [11]:

- manipulator's kinetic and potential energy,
- the Lagrangian (Eq. (15)), and
- to replace in Lagrange-Euler equation (16),

$$L(\mathbf{q}, \dot{\mathbf{q}}) = K(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q}), \quad (15)$$

$$\boldsymbol{\tau} = \frac{d}{dt} \left(\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}}, \quad (16)$$

where:

L : Lagrangian function (Lagrangian).

K : Kinetic energy.

U : Potential energy.

$\mathbf{q}$: Vector of generalized (joint) coordinates.

$\dot{\mathbf{q}}$: Vector of generalized (joint) speeds.

$\boldsymbol{\tau}$: Vector of generalized forces (forces and torques).

In this way, the dynamic model of a manipulator with n joints can be expressed through Eq. (17) [9, 12, 13]:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{F}(\dot{\mathbf{q}}), \quad (17)$$

where:

$\boldsymbol{\tau}$: Vector of generalized forces ($n \times 1$ dimension).

$\mathbf{M}$: Inertia matrix ($n \times n$ dimension).

$\mathbf{C}$: Vector of centrifugal and Coriolis forces ($n \times 1$ dimension).

$\mathbf{q}$: Components of the joint position vector.

$\dot{\mathbf{q}}$: Components of the joint speed vector.

$\mathbf{G}$: Vector of gravitational force ($n \times 1$ dimension).

$\ddot{\mathbf{q}}$: Vector of joint acceleration ($n \times 1$ dimension).

$\mathbf{F}$: Vector of friction forces ($n \times 1$ dimension).

Therefore, from Eqs. (15), (16) and (17) the dynamic model of the redundant robotic manipulator can be expressed through Eqs. (18) to (35):

$$\mathbf{M} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} & M_{15} \\ M_{21} & M_{22} & M_{23} & M_{24} & M_{25} \\ M_{31} & M_{32} & M_{33} & M_{34} & M_{35} \\ M_{41} & M_{42} & M_{43} & M_{44} & M_{45} \\ M_{51} & M_{52} & M_{53} & M_{54} & M_{55} \end{bmatrix}, \quad (18)$$

$$M_{11} = m_1 + m_2 + m_3 + m_4 + m_5, \quad (19)$$

$$M_{12} = M_{21} = M_{13} = M_{31} = M_{14} = M_{41} = \dots \quad (20)$$

$$M_{25} = M_{52} = M_{35} = M_{53} = M_{45} = M_{54} = 0$$

$$M_{15} = M_{51} = -m_5 \quad (21)$$

$$M_{22} = l_{c2}^2 m_2 + (l_2^2 + l_{c3}^2 + 2l_2 l_{c3} c_3) m_3 + \dots \quad (22)$$

$$(l_2^2 + l_3^2 + l_{c4}^2 + 2l_2 l_3 c_3) m_4 + \dots$$

$$2(l_3 l_{c4} c_4 + l_2 l_{c4} c_{34}) m_4 + \dots$$

$$(l_2^2 + l_3^2 + l_4^2 + 2l_2 l_3 c_3) m_5 + \dots$$

$$2(l_3 l_4 c_4 + l_2 l_4 c_{34}) m_5 + \dots$$

$$I_{2zz} + I_{3zz} + I_{4zz}$$

$$M_{23} = M_{32} = (l_{c3}^2 + l_2 l_{c3} c_3) m_3 + I_{3zz} + I_{4zz} + \dots \quad (23)$$

$$(l_3^2 + l_{c4}^2 + l_2 l_3 c_3 + 2l_3 l_{c4} c_4) m_4 + \dots$$

$$(l_2 l_{c4} c_{34}) m_4 + (l_2 l_4 \cos_{34}) m_5 \dots +$$

$$(l_3^2 + l_4^2 + l_2 l_3 c_3 + 2l_3 l_4 c_4) m_5$$

$$M_{24} = M_{42} = (l_{c4}^2 + l_3 l_{c4} c_4 + l_2 l_{c4} c_{34}) m_4 + \dots \quad (24)$$

$$(l_4^2 + l_3 l_4 c_4 + l_2 l_4 c_{34}) m_5 + I_{4zz}$$

$$M_{33} = l_{c3}^2 m_3 + (l_3^2 + l_{c4}^2 + 2l_3 l_{c4} c_4) m_4 + \dots \\ (l_3^2 + l_4^2 + 2l_3 l_4 c_4) m_5 + I_{3zz} + I_{4zz} \quad (25)$$

$$M_{34} = M_{43} = (l_{c4}^2 + l_3 l_{c4} c_4) m_4 + \dots \\ (l_4^2 + l_3 l_4 c_4) m_5 + I_{4zz} \quad (26)$$

$$M_{44} = l_{c4}^2 m_4 + l_4^2 m_5 + I_{4zz} \quad (27)$$

$$M_{55} = m_5 \quad (28)$$

$$\mathbf{C} = [C_{11} \quad C_{21} \quad C_{31} \quad C_{41} \quad C_{51}]^T \quad (29)$$

$$C_{11} = C_{15} = 0 \quad (30)$$

$$C_{12} = -(l_{c3} m_3 + l_3 m_4 + l_3 m_5) l_2 s_3 \cdot \dot{\theta}_3^2 + \dots \\ (l_{c4} m_4 + l_4 m_5) l_2 s_{34} \cdot \dot{\theta}_3^2 + \dots \\ -(l_{c4} m_4 + l_4 m_5) (l_2 s_{34} + l_3 s_4) \dot{\theta}_4^2 + \dots \\ -2(l_{c3} m_3 + l_3 m_4 + l_3 m_5) l_2 s_3 \cdot \dot{\theta}_2 \dot{\theta}_3 + \dots \\ (l_{c4} m_4 + l_4 m_5) l_2 s_{34} \cdot \dot{\theta}_2 \dot{\theta}_3 + \dots \\ -2(l_{c4} m_4 + l_4 m_5) (l_2 s_{34} + l_3 s_4) \dot{\theta}_2 \dot{\theta}_4 + \dots \\ -2(l_{c4} m_4 + l_4 m_5) (l_3 s_4 + l_2 s_{34}) \dot{\theta}_3 \dot{\theta}_4 \quad (31)$$

$$C_{13} = (l_{c3} m_3 + l_3 m_4 + l_3 m_5) l_2 s_3 \cdot \dot{\theta}_2^2 + \dots \\ (l_{c4} m_4 + l_4 m_5) l_2 s_{34} \cdot \dot{\theta}_2^2 + \dots \\ -(l_{c4} m_4 + l_4 m_5) l_3 s_4 \cdot \dot{\theta}_4^2 + \dots \\ -2(l_{c4} m_4 + l_4 m_5) l_3 s_4 \cdot \dot{\theta}_2 \dot{\theta}_4 + \dots \\ -2(l_{c4} m_4 + l_4 m_5) l_3 s_4 \cdot \dot{\theta}_3 \dot{\theta}_4 \quad (32)$$

$$C_{14} = (l_{c4} m_4 + l_4 m_5) (l_3 s_4 + l_2 s_{34}) \dot{\theta}_2^2 + \dots \\ (l_{c4} m_4 + l_4 m_5) l_3 s_4 \cdot \dot{\theta}_3^2 + \dots \\ 2(l_4 m_5 + l_{c4} m_4) l_3 s_4 \cdot \dot{\theta}_2 \dot{\theta}_3 \quad (33)$$

$$\mathbf{G} = [(m_1 + m_2 + m_3 + m_4 + m_5) g_z \quad 0 \quad 0 \quad 0 \quad -m_5 g_z]^T \quad (34)$$

$$\mathbf{F} = [F_{11} \quad F_{21} \quad F_{31} \quad F_{41} \quad F_{51}]^T \quad (35)$$

where:

$$s_3 = \sin \theta_3, \quad s_4 = \sin \theta_4, \quad c_3 = \cos \theta_3, \\ c_4 = \cos \theta_4, \quad s_{34} = \sin(\theta_3 + \theta_4), \quad c_{34} = \cos(\theta_3 + \theta_4),$$

m_1 : Mass of the first link.
 m_2 : Mass of the second link
 m_3 : Mass of the third link.

m_4 : Mass of the fourth link.

m_5 : Mass of the fifth link.

l_1 : Length of the first link.

l_2 : Length of the second link.

l_3 : Length of the third link.

l_4 : Length of the fourth link.

l_5 : Length of the fifth link.

l_{c2} : Length from the origin of the 2nd link to its centroid.

l_{c3} : Length from the origin of the 3rd link to its centroid.

l_{c4} : Length from the origin of the 4th link to its centroid.

l_{2zz} : Inertial momentum of the 2nd link with respect to the 1st z axis of its joint.

l_{3zz} : Inertial momentum of the 3rd link with respect to the 1st z axis of its joint.

l_{4zz} : Inertial momentum of the 4th link with respect to the 1st z axis of its joint.

In the manipulator's dynamic model, as pointed in Eq. (17), the term corresponding to centrifugal and Coriolis forces is frequently expressed through a $\mathbf{V}_m$ matrix, as in (36):

$$\boldsymbol{\tau} = \mathbf{M}(q) \ddot{\mathbf{q}} + \mathbf{V}_m(q, \dot{q}) \dot{\mathbf{q}} + \mathbf{G}(q) + \mathbf{F}(\dot{q}) \quad (36)$$

According to this, the matrix $\mathbf{V}_m$ can be expressed by means of Eqs. (37) to (46):

$$\mathbf{V}_m = \begin{bmatrix} V_{m11} & V_{m12} & V_{m13} & V_{m14} & V_{m15} \\ V_{m21} & V_{m22} & V_{m23} & V_{m24} & V_{m25} \\ V_{m31} & V_{m32} & V_{m33} & V_{m34} & V_{m35} \\ V_{m41} & V_{m42} & V_{m43} & V_{m44} & V_{m45} \\ V_{m51} & V_{m52} & V_{m53} & V_{m54} & V_{m55} \end{bmatrix} \quad (37)$$

$$V_{m11} = V_{m12} = V_{m13} = V_{m14} = V_{m15} = V_{m21} = \dots \\ V_{m25} = V_{m31} = V_{m35} = V_{m41} = V_{m44} = V_{m45} = \dots \\ V_{m51} = V_{m52} = V_{m53} = V_{m54} = V_{m55} = 0 \quad (38)$$

$$V_{m22} = -(l_{c3} m_3 + l_3 m_4 + l_3 m_5) l_2 s_3 \cdot \dot{\theta}_3 + \dots \\ (l_{c4} m_4 + l_4 m_5) l_2 s_{34} \cdot \dot{\theta}_3 + \dots \\ -(l_{c4} m_4 + l_4 m_5) (l_3 s_4 + l_2 s_{34}) \dot{\theta}_4 \quad (39)$$

$$V_{m23} = -(l_{c3} m_3 + l_3 m_4 + l_3 m_5) l_2 s_3 \cdot \dot{\theta}_2 + \dots \\ -(l_{c4} m_4 + l_4 m_5) l_2 s_{34} \cdot \dot{\theta}_2 + \dots \\ -(l_{c3} m_3 + l_3 m_4 + l_3 m_5) l_2 s_3 \cdot \dot{\theta}_3 + \dots \\ -(l_{c4} m_4 + l_4 m_5) l_2 s_{34} \cdot \dot{\theta}_3 + \dots \\ -((l_{c4} m_4 + l_4 m_5) (l_3 s_4 + l_2 s_{34})) \dot{\theta}_4 \quad (40)$$

$$V_{m24} = -(l_{c4} m_4 + l_4 m_5) (l_3 s_4 + l_2 s_{34}) (\dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4) \quad (41)$$

$$V_{m32} = (l_{c3} m_3 + l_3 m_4 + l_3 m_5) l_2 s_3 \cdot \dot{\theta}_2 + \dots \\ (l_{c4} m_4 + l_4 m_5) l_2 s_{34} \cdot \dot{\theta}_2 + \dots \\ -(l_{c4} m_4 + l_4 m_5) l_3 s_4 \cdot \dot{\theta}_4 \quad (42)$$

$$V_{m33} = -(l_{c4}m_4 + l_4m_5)l_3s_4 \cdot \dot{\theta}_4 \quad (43)$$

$$V_{m34} = -(l_{c4}m_4 + l_4m_5)l_3s_4(\dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4) \quad (44)$$

$$V_{m42} = (l_{c4}m_4 + l_4m_5)(l_2s_{34} + l_3s_4)\dot{\theta}_2 + \dots \\ (l_{c4}m_4 + l_4m_5)l_3s_4 \cdot \dot{\theta}_3 \quad (45)$$

$$V_{m43} = (l_{c4}m_4 + l_4m_5)l_3s_4(\dot{\theta}_2 + \dot{\theta}_3) \quad (46)$$

4. Actuator Dynamics

The actuators considered in this study are servo motors, which are composed by a direct current motor, a gear train for reducing spin speed and increasing torque in the drive shaft, a potentiometer connected to the output axis which is used to know a position, and a feedback control circuit which converts a PWM (Pulse-Width Modulation) input signal into voltage, compares it with the feedback position, and then amplifies it to drive an H-bridge so as to produce the spin at a specified speed [3].

In Figs. 6 and 7 we can see, respectively, a schematic diagram and a block diagram of a servomotor coupled to a robotic manipulator as a load [14].

The dynamic model of the considered servo motors has been developed by authors in [14], and is given by expressions (47) and (48):

$$\tau_L = \frac{1}{n} \left(\frac{k_a}{R_a} A k_s k_p v_i - J_m \frac{1}{n} \ddot{q} \right) - \dots \\ \frac{1}{n} \left(\left(\frac{k_a k_b}{R_a} + B_m \right) \frac{1}{n} \dot{q} + \frac{k_a}{R_a} A k_s p q + f_{ec} \left(\frac{1}{n} \dot{q} \right) \right) \quad (47)$$

$$f_{ec}(\dot{q}) = F_{ec1} \tanh(k\dot{q})(1 + \text{sgn}(\dot{q}))/2 + \dots \\ F_{ec2} \tanh(k\dot{q})(1 - \text{sgn}(\dot{q}))/2 \quad (48)$$

where:

- n : Gear ratio (n_1/n_2).
- k_a : Motor torque constant.
- R_a : Armature resistance.
- A : Gain of the power amplifier (H bridge).
- k_s : Comparator sensibility.
- k_p : Total PWM ($k_{p1} \cdot k_{p2}$) conversion gain.
- v_i : Servomotor input voltage.
- J_m : Motor inertial momentum.
- k_b : Inverse electromotive force constant.
- B_m : Motor viscous friction.
- p : Position potentiometer gain.
- k : $\tanh$ function slope gain.

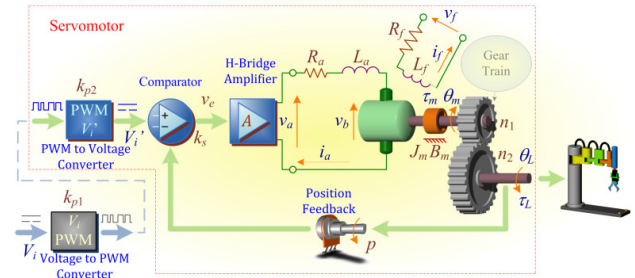


Figure 6. Schematic diagram of a servomotor coupled to a robotic manipulator as a load.

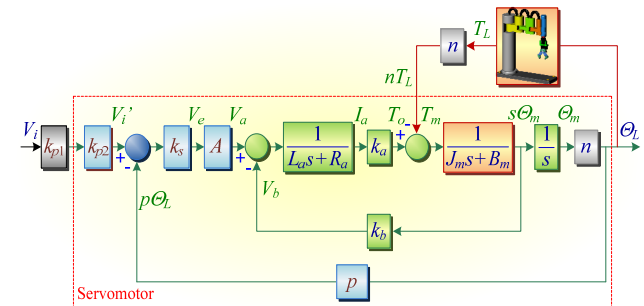


Figure 7. Block diagram of a servomotor coupled to a robotic manipulator as a load.

k is a constant used to increase or reduce the curve slope in the zero-crossing.

5. Considered Controllers

Next we present a summary of the controllers considered for model evaluation of the robot along with its actuators, with their respective performance.

6. Hyperbolic Sine-Cosine Controller

This controller, already presented in [15], is composed by a proportional part based on hyperbolic sine-cosine functions, a derivative part based on a hyperbolic sine, and gravity compensation, as pointed in Eqs. (49) and (50).

$$\tau = K_p \sinh(q_e) \cosh(q_e) + \dots \\ - K_v \sinh(\dot{q}) + G(q) \quad (49)$$

$$q_e = q_d - q \quad (50)$$

where:

$$K_p = \text{diag}(K_{p1}, K_{p2}, \dots, K_{pn}) \quad (51)$$

K_p : Proportional gain matrix, diagonal and definite positive ($n \times n$ dimension).

$$K_v = \text{diag}(K_{v1}, K_{v2}, \dots, K_{vn}) \quad (52)$$

K_v : Derivative gain matrix, diagonal and definite positive ($n \times n$ dimension).

$\mathbf{q}_d$: Joint desired position vector ($n \times 1$ dimension).

$$\lim_{t \rightarrow \infty} \mathbf{q}_e \rightarrow 0 \quad (53)$$
$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} \mathbf{q}_e \\ \dot{\mathbf{q}} \end{bmatrix} &= \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \\ a_1 &= -\dot{\mathbf{q}} \\ a_2 &= \mathbf{M}(\mathbf{q})^{-1} \left(\mathbf{K}_p \sinh(\mathbf{q}_e) \cosh(\mathbf{q}_e) + \dots \right. \\ &\quad \left. - \mathbf{K}_v \sinh(\dot{\mathbf{q}}) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} \right) \end{aligned} \quad (54)$$

The control law corresponds to:

$$\boldsymbol{\tau} = -\mathbf{K} \cdot \text{sgn}(\mathbf{s}) \quad (55)$$

$$\mathbf{K} = \text{diag}(K_1, K_2, \dots, K_n) \quad (56)$$
$$\mathbf{s} = \mathbf{W} \cdot (\mathbf{q} - \mathbf{q}_d) + (\dot{\mathbf{q}} - \dot{\mathbf{q}}_d) \quad (57)$$
$$\mathbf{W} = \text{diag}(W_1, W_2, \dots, W_n) \quad (58)$$

Another developed controller employs a control law by calculated torque, consisting in the application of a torque in order to compensate the centrifugal, Coriolis, gravitatory and friction effects, as stated in Eq. (59) [20]:

$$\begin{aligned} \boldsymbol{\tau} = & \hat{\mathbf{M}}(q)(\ddot{\mathbf{q}}_d + \mathbf{K}_v \dot{\mathbf{q}}_e + \mathbf{K}_p \mathbf{q}_e) + \dots \\ & \hat{\mathbf{C}}(q, \dot{q}) + \hat{\mathbf{G}}(q) + \hat{\mathbf{F}}(\dot{q}) \end{aligned} \quad (59)$$

$\hat{\mathbf{M}}$: Estimation of inertia matrix ($n \times n$ dimension).

$\hat{\mathbf{G}}$: Estimation of gravity force vector ($n \times 1$ dimension).

$\hat{\mathbf{F}}$: Estimation of friction forces vector ($n \times 1$ dimension).

$\mathbf{K}_v$: Definite positive diagonal matrix ($n \times n$ dimension) defined in (52).

$\mathbf{K}_p$: Definite positive diagonal matrix ($n \times n$ dimension) defined in (51).

$\mathbf{q}_e$: Joint position error vector ($n \times 1$ dimension) defined in (50).

$$\dot{\mathbf{q}}_e = \dot{\mathbf{q}}_d - \dot{\mathbf{q}} \quad (60)$$

$\dot{\mathbf{q}}_e$: Joint speed error vector ($n \times 1$ dimension).

$\ddot{\mathbf{q}}_d$: Joint desired acceleration vector ($n \times 1$ dimension).

If the estimation errors are little, joints errors approach to a linear equation, as pointed in Eq. (61).

$$\ddot{\mathbf{q}}_e + \mathbf{K}_v \dot{\mathbf{q}}_e + \mathbf{K}_p \mathbf{q}_e \approx 0 \quad (61)$$

[illegible]Claudio Urrea and John Kern: Modeling, Simulation and Control
of a Redundant SCARA-Type Manipulator Robot

The three control laws above mentioned, along with the SCARA-type redundant manipulator dynamic model and the actuator dynamics, were run in a simulation structure, created by means of MatLab/Simulink programming tools, as shown in Fig. 8.

Parameter values considered for the manipulator are displayed in Table 2.

	Link1	Link2	Link3	Link4	Link5	Units
l	0.524	0.2	0.2	0.2	0.14	[m]
l_c	-	0.0229	0.0229	0.0229	-	[m]
m	1.228	1.023	1.023	1.023	0.5114	[kg]
I_{zz}	-	0.0058	0.0058	0.0058	-	[kg·m ²]
F_v	0.03	0.025	0.025	0.025	0.02	[N·m·s/rad]
F_{eca}	0.05	0.05	0.05	0.05	0.03	[N·m]
F_{ecb}	-0.05	-0.05	-0.05	-0.05	-0.03	[N·m]

Table 2. Parameters considered for the manipulator.

Table 3. shows the set of parameter values used for each actuator.

Table 4. shows the set of gain values employed for each kind of controller.

7. Results

After developing the manipulator model and the simulation environment - including the actuator dynamics - and establishing the control laws to be used, a test trajectory was determined in the space to subject the manipulator's model to path tracking, and we then studied the results in terms of the performance of each controller. This trajectory is shown in Fig. 9.

	Servo 1	Servo 2-3-4	Servo 5	Units
R_a	1.6	1.6	1.6	[Ω]
L_a	0.0048	0.0048	0.0048	[H]
J_m	0.007	0.007	0.007	[kg·m ²]
B_m	0.01413	0.01313	0.01208	[N·m·s/rad]
k_a	0.35	0.35	0.35	[N·m/A]
k_b	0.04	0.04	0.04	[V·s/rad]
F_{eca}	0.05	0.05	0.03	[N·m]
F_{ecb}	-0.05	-0.05	-0.03	[N·m]
n	1/600	1/561.6	1/561.6	[Times]
A	15	15	15	[Times]
k_s	10	10	10	[Times]
k_p	1	1	1	[Times]
p	1	1	1	[Times]

Table 3. Parameters considered for servo motors.

Constants	Kind of Controller		
	Sinh-Cosh	Sliding Mode	Calculated Torque
$K_{p1}, K_{p2}, K_{p3}, K_{p4}, K_{p5}$	400, 300, 200, 100, 100	—	400, 600, 700, 800, 100
$K_{v1}, K_{v2}, K_{v3}, K_{v4}, K_{v5}$	5, 4, 3, 2, 2	—	120, 100, 60, 50, 40
K_1, K_2, K_3, K_4, K_5	—	0.74, 1.45, 1.4, 1.35, 1.54	—
W_1, W_2, W_3, W_4, W_5	—	10, 10, 10, 10, 10	—

Table 4. Gains considered for the controllers.

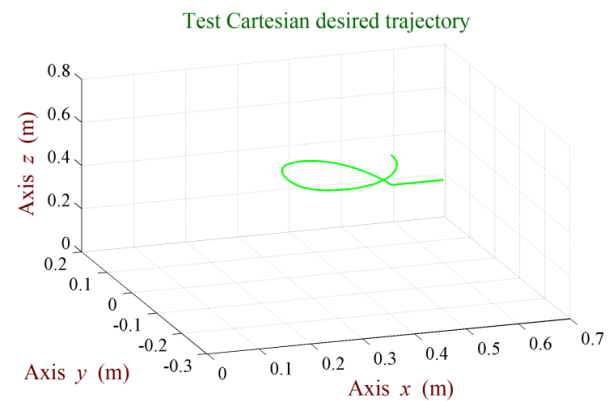


Figure 9. Test Cartesian desired trajectory.

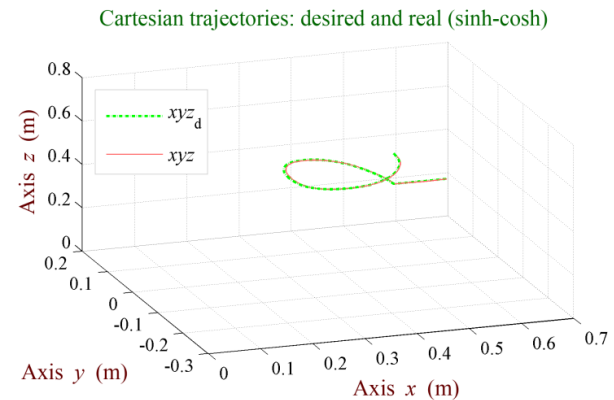


Figure 10. Comparison in the space of the desired and real Cartesian trajectories using the hyperbolic sine-cosine controller.

Figs. 10, 11 and 12 show the curves corresponding to comparisons between the desired Cartesian trajectory and the real Cartesian trajectory, displaying the views in space, plane x - y and plane y - z , respectively. All of this is under the action of the hyperbolic sine-cosine controller, where:

- xyz_d : Desired Cartesian trajectory, in space.
- xyz : Real Cartesian trajectory, in space.
- xy_d : Desired Cartesian trajectory, in plane x - y .
- xy : Real Cartesian trajectory, in plane x - y .
- yz_d : Desired Cartesian trajectory, in plane y - z .
- yz : Real Cartesian trajectory, in plane y - z .

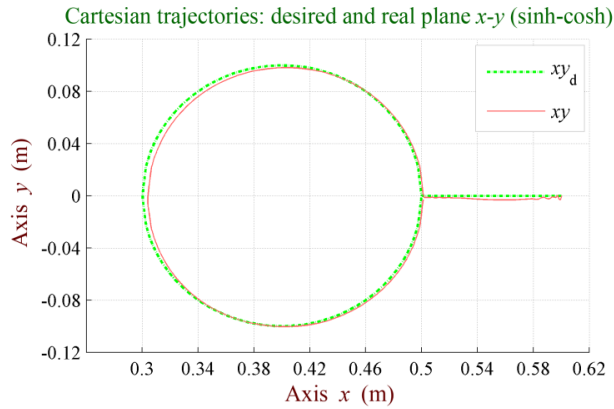


Figure 11. Comparison in plane x - y of the desired and real Cartesian trajectories using the hyperbolic sine-cosine controller.

In Fig. 13 are seen the curves corresponding to desired and real joint trajectories, when using the hyperbolic sine-cosine controller, where:

q_{dn} : Desired joint trajectory, where n represents joints 1 to 5.

q_n : Real joint trajectory, where n represents joints 1 to 5.

Forces and torques supplied to the robot by actuators, under the action of hyperbolic sine-cosine controller, are shown in Fig. 14, where:

f_1 : Force applied in joint 1.

τ_2 : Torque applied in joint 2.

τ_3 : Torque applied in joint 3.

τ_4 : Torque applied in joint 4.

f_5 : Force applied in joint 5.

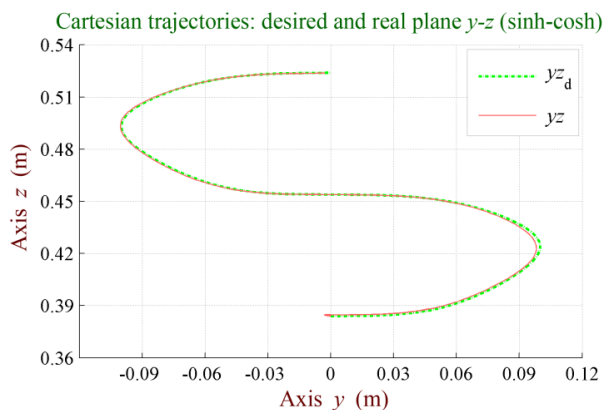


Figure 12. Comparison in plane y - z of the desired and real Cartesian trajectories using the hyperbolic sine-cosine controller.

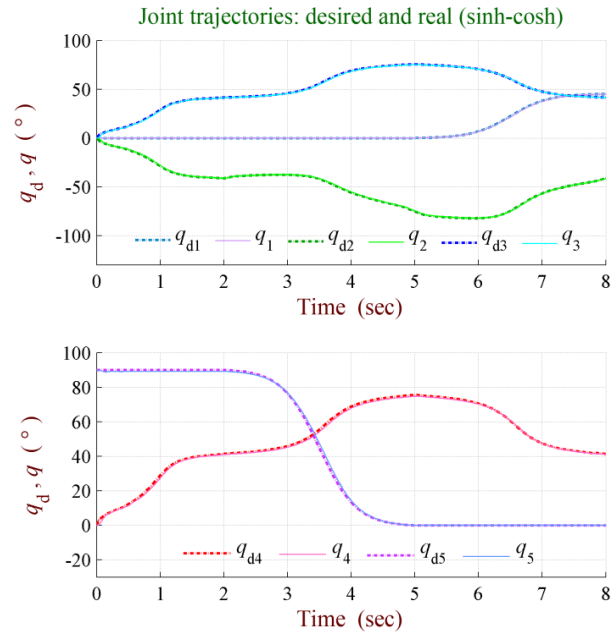


Figure 13. Comparison of the real and desired joint trajectories using the hyperbolic sine-cosine controller.

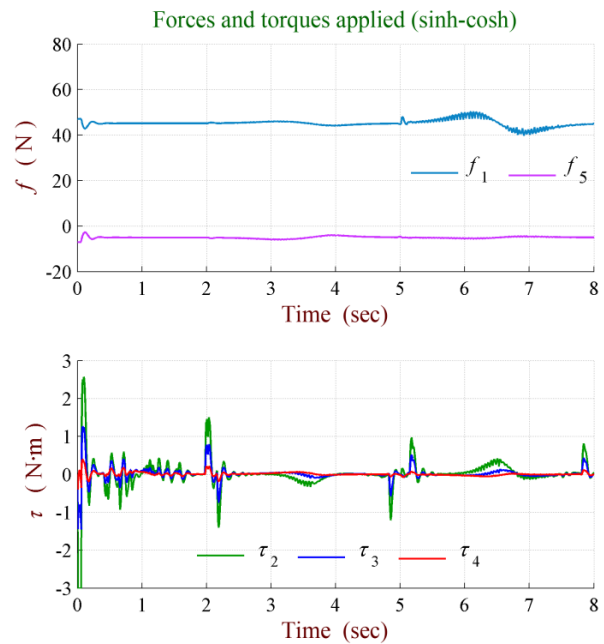


Figure 14. Forces and torques applied to the robot when using the hyperbolic sine-cosine controller.

Figs. 15 and 16 display the errors obtained from desired and real Cartesian trajectories, and desired and real joint trajectories, respectively, when using the hyperbolic sine-cosine controller, where:

e_x : Error in Cartesian trajectory, x axis.

e_y : Error in Cartesian trajectory, y axis.

e_z : Error in Cartesian trajectory, z axis.

e_n : Error in joint trajectory, where n represents joints 1 to 5.

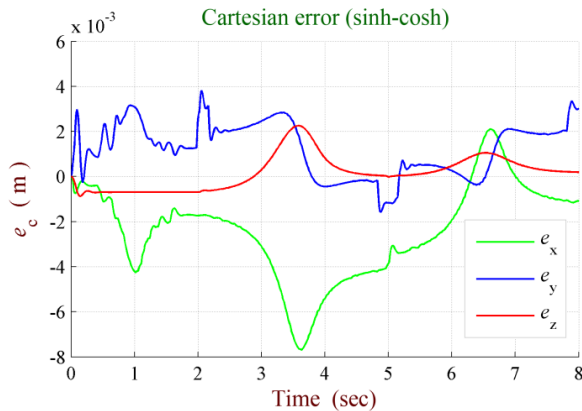


Figure 15. Cartesian trajectory error, when using the hyperbolic sine-cosine controller.

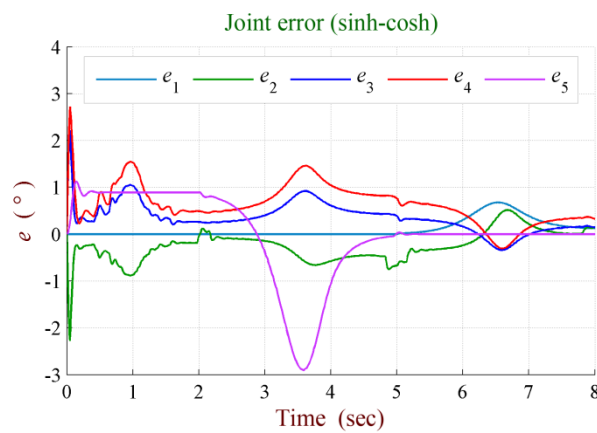


Figure 16. Joint trajectory error, when using the hyperbolic sine-cosine controller.

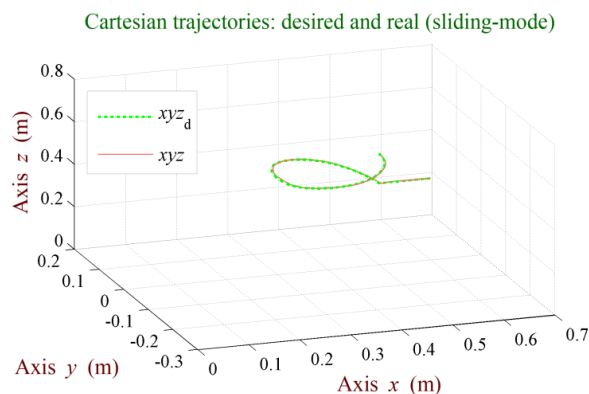


Figure 17. Comparison in the space of the desired and real Cartesian trajectories using the sliding mode controller.

Comparisons between the desired and real Cartesian trajectories, under the sliding mode controller, are shown in Figs. 17, 18 and 19, displaying the charts corresponding to the space, the plane x - y and the plane y - z , respectively.

In Fig. 20 we display the charts related to joint desired and real trajectories when using the sliding mode controller.

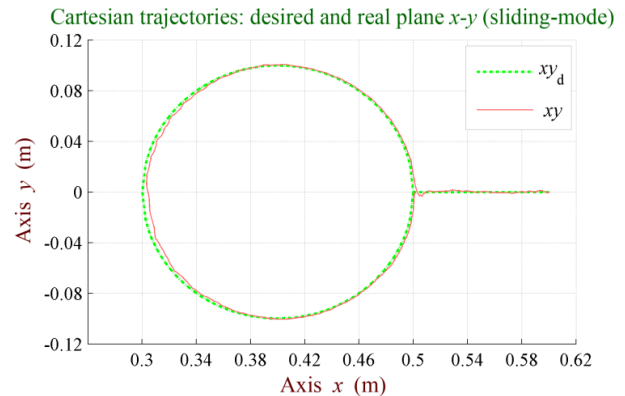


Figure 18. Comparison in plane x - y of the desired and real Cartesian trajectories using the sliding mode controller.

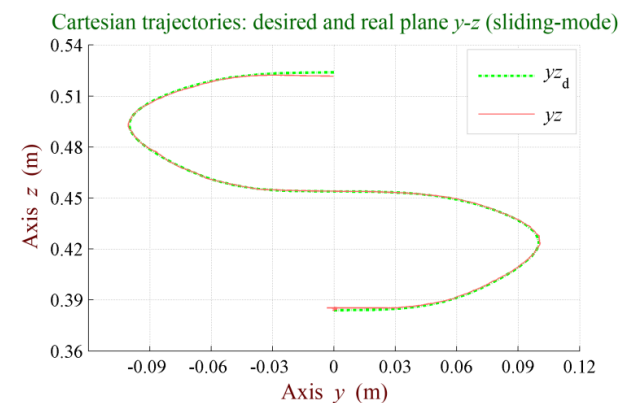


Figure 19. Comparison in the plane y - z of Cartesian desired and real trajectories using the sliding mode controller.

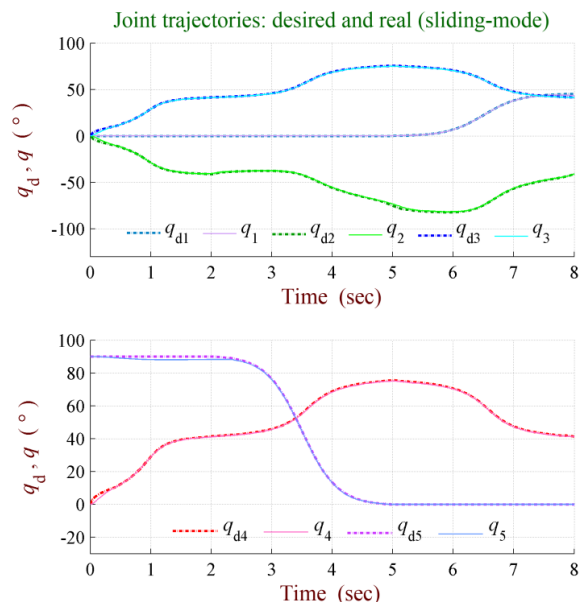


Figure 20. Comparison of the joint desired and real trajectories when using the sliding mode controller.

Fig. 21 shows the response of the actuators, indicating forces and torques developed during trajectory tracking, when using the sliding mode controller.

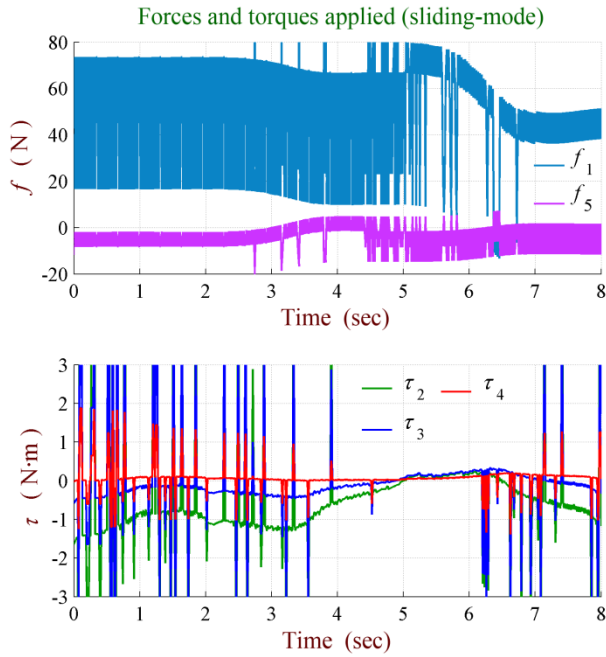


Figure 21. Forces and torques applied to the robot when using the sliding mode controller.

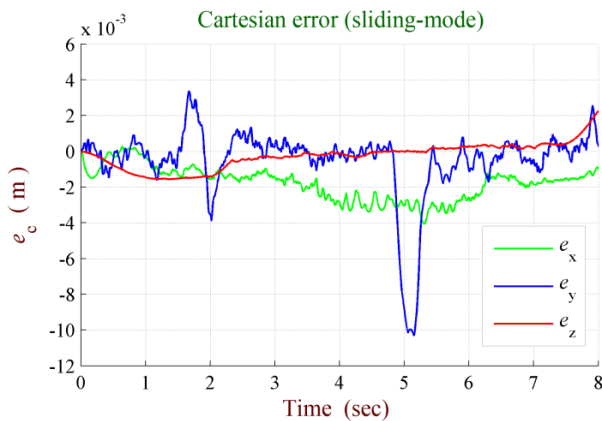


Figure 22. Error in Cartesian trajectory when using the sliding mode controller.

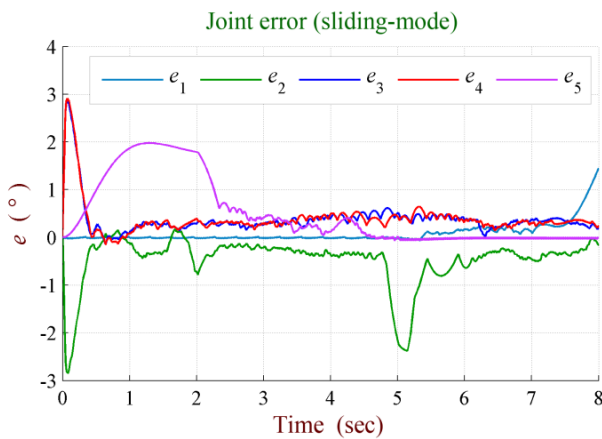


Figure 23. Error in joint trajectory when using the sliding mode controller.

Errors produced by the difference between the desired and real Cartesian trajectories and the joint desired and real trajectories when applying the sliding mode controller, are shown in Figs. 22 and 23, respectively.

The tracking response of robot model in the joint space, when using the calculated torque controller, is shown in Figs. 24, 25 and 26, displaying the desired and real curves in the space, the plane x - y and the plane y - z , respectively.

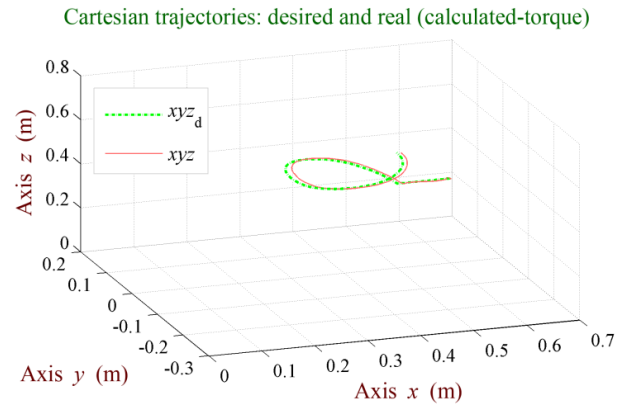


Figure 24. Comparison in the space of the desired and real Cartesian trajectories when using the calculated torque controller.

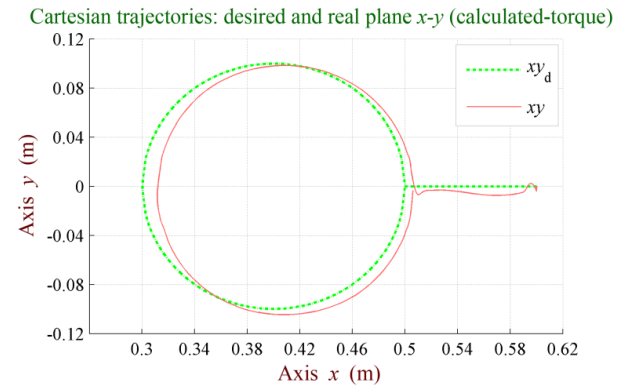


Figure 25. Comparison in plane x - y of the desired and real Cartesian trajectories when using the calculated torque controller.

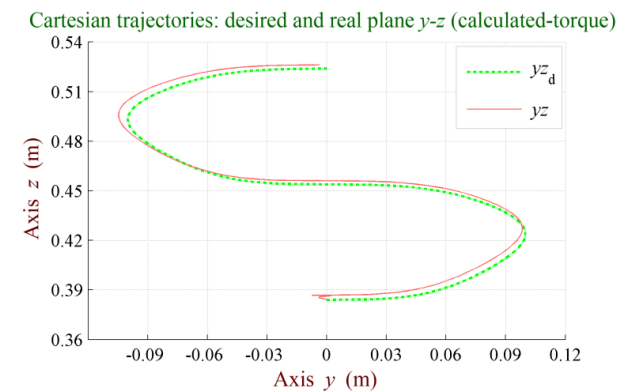


Figure 26. Comparison in plane y - z of the desired and real Cartesian trajectories when using the calculated torque controller.

Comparisons between joint desired and real trajectories under the calculated torque controller are shown in Fig. 27.

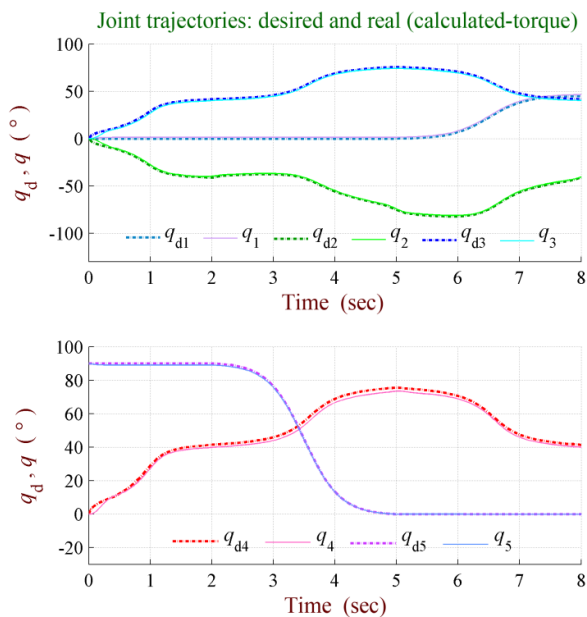


Figure 27. Comparison of the joint desired and real trajectories when using the calculated torque controller.

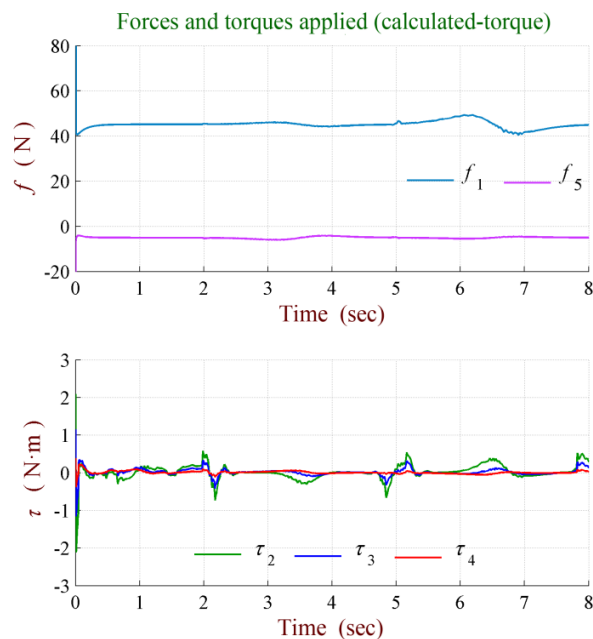


Figure 28. Forces and torques applied to the robot when using the calculated torque controller.

In Fig. 28 are seen the curves corresponding to the forces and torques applied by servo motors under the application of the calculated torque controller.

In Figs. 29 and 30, we can see the error curves obtained from Cartesian and joint desired and real trajectories when using the calculated torque controller.

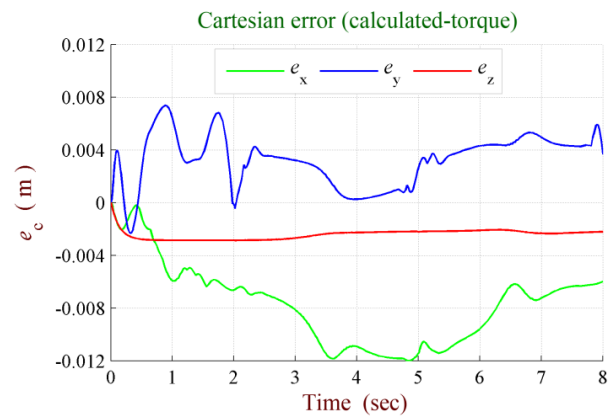


Figure 29. Error in Cartesian trajectory when using the calculated torque controller.

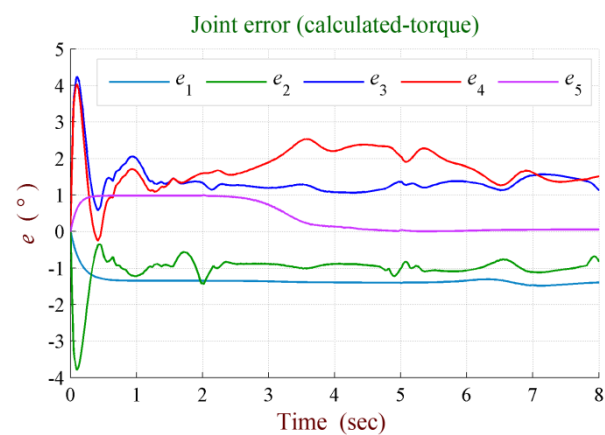


Figure 30. Error in joint trajectory when using the calculated torque controller.

Finally, figs. 31 and 32 show joint and Cartesian *rms* errors, respectively, in accordance with Eq. (62).

$$rms = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2} \quad (62)$$

Where e_i represents both the joint and Cartesian trajectory errors and n is the number of data.

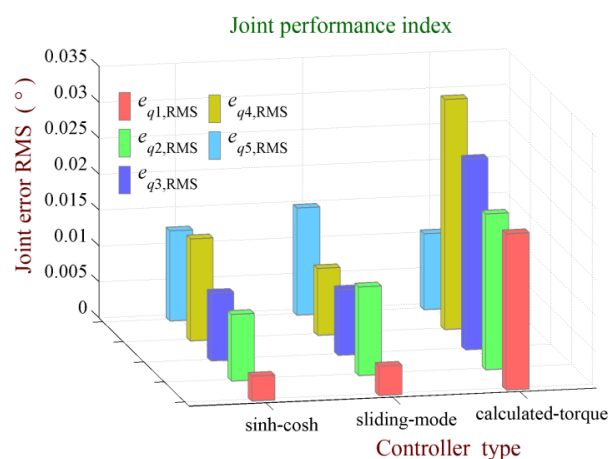


Figure 31. Performance index corresponding to joint trajectory.

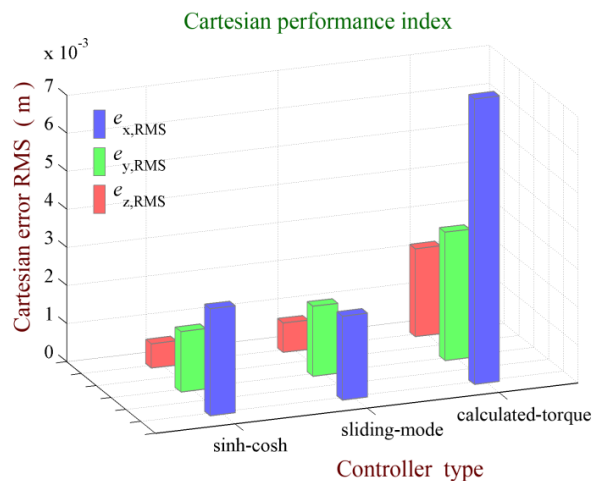


Figure 32. Performance index corresponding to the Cartesian trajectory.

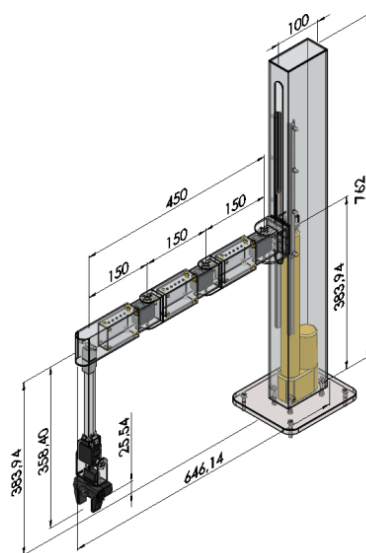


Figure 33. Scheme of a SCARA-type redundant robot (view A).

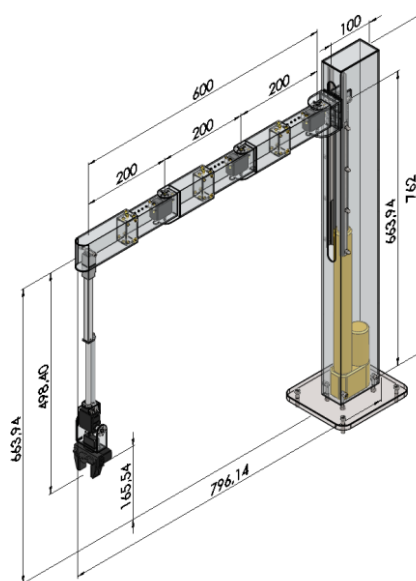


Figure 34. Scheme of a SCARA-type redundant robot (view B).

8. Conclusions

In this paper, we developed the kinematic and dynamic model of a redundant SCARA-type manipulator robot with five degrees of freedom using the Denavit-Hartenberg and Lagrange-Euler methods, respectively. Three controllers were developed: hyperbolic sine-cosine, sliding mode and calculated torque. A simulator was also created, using the MatLab/Simulink *software*. Several tests on this model were presented, including actuator dynamics under each controller through a path tracking test consisting of a spiral in the Cartesian space. The results obtained through the simulation environment were displayed by means of comparative curves and an *rms* index for joint and Cartesian errors.

From the results obtained, we can see that the redundant manipulator model presented a path tracking response whose maximum errors were less severe when using the hyperbolic sine-cosine controller than in the other two cases, leading to more homogeneous manipulator movements. We also appreciate that the greatest joint and Cartesian errors produced when testing the robot model, both for maximum and *rms* values, occurred when using the calculated torque controller. Consequently, the best results for the robotic manipulator's model performance were obtained when applying the hyperbolic sine-cosine controller, as shown in Figs. 31 and 32. It is important to notice that both the hyperbolic sine-cosine and the sliding mode controllers present a lesser simulation complexity, since they do not require the second derivative of joint position. Such a condition can be decisive in the case of not having high-performance processors. The practical implementation of the sliding mode controller, however, has a considerable disadvantage: the controller's high switching frequency, leading to a remarkable deterioration of the actuators.

9. Further Research

From the behavior obtained through the simulation tests for the redundant robot model with its actuators and the different control laws that were analyzed, we can begin a new stage in the study and analysis of redundant robotic manipulators, consisting in the practical implementation of real industrial-type robots, its actuators and controllers, by means of the development of the proper hardware, as is schematized in Figs. 33 and 34.

10. Acknowledgements

This work had the support of the Department of Scientific and Technologic Research of the Universidad de Santiago de Chile, through Project 060713UO.

11. References

- [1] Angulo, J. and Avilés, R., *Curso de Robótica*, Paraninfo S.A., Madrid, 1989.
- [2] Selig, J. M., *Introductory Robotics*, Prentice Hall, London, 1992.
- [3] Iñigo, R. and Vidal, E., *Robot Industriales Manipuladores*, Alfaomega Grupo Editor, México, 2004.
- [4] Torres, F. and et al, *Robots y Sistemas Sensoriales*, (Pearson Educación S.A., Madrid, 2002.
- [5] Craig, J., *Robótica Tercera Edición*, Pearson Educación, México, 2006.
- [6] Rubí, J., *Cinemática, Dinámica y Control de Robots Redundantes y Robots Subactuados*, Tesis Doctoral, Universidad de Navarra, San Sebastián, España, 2002.
- [7] Ollero, A., *Robótica Manipuladores y Robots Móviles*, Marcombo, S.A., Barcelona, España, 2001.
- [8] Patel, R. and Shadpey, F., *Control of Redundant Robot Manipulators*, Springer Verlag, Berlin Heidelberg, 2005.
- [9] Spong, M., Hutchinson, S. and Vidyasagar, M., *Robot Modeling and Control, First Edition*, John Wiley & Sons, Inc., New York, 2005.
- [10] Lewis, F., Dawson, D. and Abdallah, Ch., *Robot Manipulator Control Theory and Practice*, Marcel Dekker Inc., New York, 2004.
- [11] Paul, R., *Robot Manipulators: Mathematics, Programming, and Control*, The MIT Press., Cambridge, Massachusetts, 1981.
- [12] Angeles, J., *Fundamentals of Robotic Mechanical System: Theory, Methods, and Algorithms, Third Edition*, Springer, New York, 2006.
- [13] Craig, J., *Introduction to Robotics, Mechanics and Control*, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1986.
- [14] Urrea, C. and Kern, J., *A New Model for Analog Servo motors. Practical Implementations*, Canadian Journal on Automation, Control and Intelligent Systems 2, 29-38, 2011.
- [15] Barahona, J., Espinosa, L. and Reyes, F., *Evaluación Experimental de Controladores de Posición tipo Saturados para Robot Manipuladores*, Puebla, México: Congreso Nacional de Electrónica, Centro de Convenciones William o Jenkins, 2002.
- [16] DeCarlo, R., Zak, S. and Mathews, G., *Variable Structure Control of Nonlinear Multivariable Systems: A Tutorial*, Proc., IEEE 76, 212-232, 1988.
- [17] Slotine, J. and Li, W., *Applied Nonlinear Control*, Prentice-Hall Inc., New Jersey, 1991.
- [18] Xu, J., Lee, T. H., Wang, M. and Yu, X., *Design of Variable Structure Controllers with Continuous Switching Control*, Int. J. of Control 65, 409-431, 1996.
- [19] Chen, C. L. and Lin, C. J., *A Sliding Mode Control Approach to Robotic Tracking Problem*, Barcelona, Spain IFAC: 15th Triennial World Congress, 2002.
- [20] Siciliano, B. and Khatib, O., *Handbook of Robotics*, Springer-Verlag, Berlin, Heidelberg, 2008.