Trajectory Tracking Control of a Real Redundant Manipulator of the SCARA Type

Claudio Urrea† and John Kern*

Abstract – Modeling, control and implementation of a real redundant robot with five Degrees Freedom (DOF) of the SCARA (Selective Compliant Assembly Robot Arm) manipulator type is presented. Through geometric methods and structural and functional considerations, the inverse kinematics for redundant robot can be obtained. By means of a modification of the classical sliding mode control law through a hyperbolic function, we get a new algorithm which enables reducing the chattering effect of the real actuators, which together with the learning and adaptive controllers, is applied to the model and to the real robot. A simulation environment including the actuator dynamics is elaborated. A 5 DOF robot, a communication interface and a signal conditioning circuit are designed and implemented for feedback. Three control laws are executed in: a simulation structure (together with the dynamic model of the SCARA type redundant manipulator and the actuator dynamics) and a real redundant manipulator of the SCARA type carried out using MatLab/Simulink programming tools. The results, obtained through simulation and implementation, were represented by comparative curves and RMS indices of the joint errors, and they showed that the redundant manipulator, both in the simulation and the implementation, followed the test trajectory with less pronounced maximum errors using the adaptive controller than the other controllers, with more homogeneous motions of the manipulator.

Keywords: Robots, Redundant manipulators, Dynamic model, Controllers, Simulation

1. Introduction

Since the appearance of the first industrial robot, developed by George Devol and Joseph Engelberger, interest in using robotic manipulators has increased very rapidly. These systems have improved the productivity and quality of manufactured products and their use has extended from the automobile industry (General Motors: Unimate in 1960) [1] to the aerospace industry (National Aeronautics and Space Administration: Curiosity in 2012) [2]. This extensive range of applications has therefore required flexibilizing the work space of the robots, a characteristic that can be achieved by increasing their degrees of freedom, i.e., providing them with redundancy. However, all these activities would not be possible without an adequate design of the robot and of its technical control. Fulfilling this requires the knowledge and study of a mathematical model and of a certain class of “intelligence” that can direct the manipulator to perform the assigned tasks. Using the basic laws of physics that govern the robot's dynamics, it is possible to derive a mathematical model that represents its behavior, and through appropriate programming tools, develop an environmental simulation to subject it to different tests such as, for example, following trajectories [3-7].

This paper takes up the modeling, control, and implementation of a redundant robot with five degrees of freedom of the SCARA manipulator type that is tested by making it follow a test trajectory composed of a spiral in Cartesian space. This work on the redundant robot has provided a complete dynamic model, considering the actuators, inverse kinematics, under certain considerations, along with direct kinematics.

Three controllers were made to test both the model and the real redundant manipulator: a hyperbolic sliding mode, with learning, and adaptive. A simulator is developed by means of MatLab/Simulink software. A 5 DOF robot, a communication interface and a signal conditioning circuit are designed and implemented for feedback. A redundant robot model and real redundant manipulator are executed together with each controller. This analysis also includes the dynamics of the actuators. The results are shown by means of comparative curves and RMS indices of the joint errors, both in simulation and in implementation.

2. Redundant robots

Redundant robots are those that have more degrees of freedom than those required to perform a given task [8-11]. In recent years special attention has been given to the
study of redundant manipulators, and this redundancy has been considered as an important characteristic in the performance of tasks that require dexterity comparable to that of the human arm, such as, for example, in the space mission called Mars Science Laboratory (MSL), better known as Curiosity, shown in Fig. 1.

Although most redundant manipulators do not have a sufficient number of degrees of freedom to carry out their main tasks, e.g., following the position and/or the orientation, it is known that its restricted manipulability results in a reduction of the work space due to the mechanical limitations of the joints and to the presence of obstacles in that space. This has led researchers to study the performance of the manipulators when more degrees of freedom are added (kinematic redundancy), allowing them to fulfill additional tasks defined by the user. Those tasks can be represented as kinematic functions, including not only the functions of kinematics that reflect some desirable characteristics of the joints and the evasion of obstacles, but can also be expanded to include measurements of the dynamic performance through the definition of functions in the robot’s dynamic model, e.g., impact strength, control of inertia, etc. [12].

The proposed robotic manipulator incorporates two additional degrees of freedom, giving it redundancy in its rotational motion, in its motion on the x-y plane, as well as in its prismatic motion along the z axis, as shown in Fig. 2.

3. Redundant Manipulator with 5 DOF

Fig. 3 is a schematic diagram of the SCARA-type redundant manipulator showing its redundancy in its rotational motion, its motion on the x-y plane, as well as its prismatic motion along the z axis, as well as the distribution of the coordinate axes systems and the location of the centroids.

Fig. 3. Scheme of a redundant manipulator of the SCARA type

where $q_1$, $q_2$, $q_3$, $q_4$, $q_5$, $q_6$, $q_6$, and $l_1$, $l_2$, $l_3$, $l_4$, $l_5$, represent the generalized coordinates and the lengths of the links: first, second, third, fourth and fifth, respectively, and $l_{c2}$, $l_{c3}$ and $l_{c4}$ are the lengths from the origins to the centroids of the corresponding second, third and fourth links.

Now we make the corresponding calculations for the design of a kinematic model of the manipulator.

3.1 Kinematics

To get the kinematic model the standard method of Denavit-Hartenberg has been considered, whose parameters are indicated in Table 1.

Then, using the homogeneous transformations given by (1) and (2) we get the direct kinematic model indicated by matrix (3).

$$i-1 T_j = \text{Rot}(z_{j-1}, \theta_j) \cdot \text{Tras}(z_{j-1}, d_{j}) \cdot \text{Tras}(x_{j}, a_j) \cdot \text{Rot}(x_{j}, \alpha_j) \quad (1)$$
Table 1. Assignment of Denavit-Hartenberg parameters

<table>
<thead>
<tr>
<th>Joint</th>
<th>( \theta_i )</th>
<th>( d_i )</th>
<th>( \alpha_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0°</td>
<td>( l_1 + d_1 )</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>( \theta )</td>
<td>0</td>
<td>( l_2 )</td>
</tr>
<tr>
<td>3</td>
<td>( \theta )</td>
<td>0</td>
<td>( l_3 )</td>
</tr>
<tr>
<td>4</td>
<td>( \theta )</td>
<td>0</td>
<td>180°</td>
</tr>
<tr>
<td>5</td>
<td>0°</td>
<td>( l_4 + d_5 )</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
i^{-1}T_i = \begin{bmatrix}
\cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\
\sin \theta_i & \cos \alpha_i \cos \theta_i & -\cos \alpha_i \sin \theta_i & a_i \sin \theta_i \\
0 & \sin \alpha_i & \cos \alpha_i & d_i \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
T = \begin{bmatrix}
\c_234 & s_{234} & 0 & l_2 s_2 + l_3 s_{23} + l_4 s_{234} \\
-s_{234} & c_{234} & 0 & l_2 c_2 + l_3 c_{23} + l_4 c_{234} \\
0 & 0 & -1 & l_1 + d_1 - l_2 - d_2 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

where \( s_2 = \sin \theta_2 \), \( s_{23} = \sin(\theta_2 + \theta_3) \), \( s_{234} = \sin(\theta_2 + \theta_3 + \theta_4) \), \( c_2 = \cos \theta_2 \), \( c_{23} = \cos(\theta_2 + \theta_3) \), and \( c_{234} = \cos(\theta_2 + \theta_3 + \theta_4) \).

Getting the inverse kinematics of a redundant robot requires looking at different methods and selecting the most adequate one according to the considerations of the model. If homogeneous transformation matrices are used, it is necessary to find the \( n \) variables \( q_i \) as a function of the components of the vectors \( n, o, a \) and \( p \) corresponding to the direct kinematic model, as shown in (4):

\[
\begin{bmatrix}
n & o & a & p
\end{bmatrix} = \begin{bmatrix}
t_{ij}
\end{bmatrix}
\]

where the elements \( t_{ij} \) are functions of the joint coordinates \( \{q_1, \ldots, q_n\} \), so it is possible to think that by means of some combinations of the 12 equations stated in (4) the \( n \) variables \( q_i \) of the joints can be found as a function of the components of the vectors \( n, o, a \) and \( p \). In most cases this method is tedious, producing transcendental equations. However, if three degrees of freedom are considered, the procedure can be simplified as shown in (5):

\[
T = 0T_1 \cdot 1T_2 \cdot 2T_3
\]

Then, multiplying (5) by \((0T_1)^{-1}\) and then by \((1T_2)^{-1}\) to get \(2T_3\), we have:

\[
\left(0T_1\right)^{-1} \cdot T = 1T_2 \cdot 2T_3 \Rightarrow \left(1T_2\right)^{-1} = \left(0T_1\right)^{-1} \cdot T = 2T_3
\]

and since \( T \) is known, the members on the left in expressions (6) are functions of the joints’ variables \( \{q_1, \ldots, q_n\} \), while the members on the right are functions of the joints’ variables \( \{q_1, \ldots, q_n\} \), and in this way it is possible to reduce the complexity in getting the joints’ variables.

Fig. 4. Scheme of the three rotary DOF of the redundant robotic manipulator of the SCARA type

Applying this method to the three rotary degrees of freedom that govern the motion of the robot on the x-y plane, as seen in Fig. 4, we would get multiple solutions. That is why, for simplicity, the condition \( \theta_2 = \theta_3 \) is set.

In accordance with this, and after the adequate simplifications, we get the inverse kinematic model expressed by:

\[
\theta_2 = \pi \pm \arccos (\alpha) \quad (7)
\]

\[
\alpha = \frac{\left(l_2 (l_1 + l_3) + 4l_1 l_3 \right)}{4l_1 l_3}
\]

\[
\beta = \frac{x \pm \sqrt{x^2 + y^2 - s_3^2 \left(4l_1 c_3 (l_1 c_3 + l_2) + l_2^2\right)}}{y + l_2 s_3 + l_3 s_2}
\]

\[
d_2 = l_1 + d_1 - l_2 - z
\]

where \( z \) and \( d_1 \) are known.

3.2 Dynamics

Keeping in mind the characteristics of the manipulator presented so far, it is now necessary to get its dynamic model. For that purpose it is possible to make approximations through second order systems [13] or to develop a complete model, as achieved in [14]. In this work the Lagrange-Euler formulation that is based on the principle of the conservation of energy [15-17] is used; for which it is necessary to determine [18] the kinetic and potential energy of the manipulator, the Lagrangian \( ^2 \) (Eq. 12), and then substitute in the Lagrange-Euler Eq. (13):

\[
L(q, \dot{q}) = K(q, \dot{q}) - U(q) \quad (12)
\]

\[
\tau = \frac{d}{dt} \left( \frac{\partial L(q, \dot{q})}{\partial \dot{q}} \right) - \frac{\partial L(q, \dot{q})}{\partial q} \quad (13)
\]

Scalar function that is defined as the difference between the kinetic energy and the potential energy of a mechanical system.
where $L$ represents the Lagrangian function (the Lagrangian), $K$ is the kinetic energy, $U$ is the potential energy, $\mathbf{q}$ is the vector of generalized coordinates (joints), $\dot{\mathbf{q}}$ is the generalized velocity vector (of the joints), and $\mathbf{f}$ is the generalized forces vector (forces and torques). In this way, the dynamic model of a manipulator of $n$ joints can be expressed by means of (14) [19-22],

$$\mathbf{\tau} = \mathbf{M}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{F}(\mathbf{q}) \quad (14)$$

where $\mathbf{\tau}$ represents the generalized forces vector (with dimension $n \times 1$), $\mathbf{M}$ is the inertia matrix (with dimension $n \times n$), $\mathbf{C}$ is the centrifugal and Coriolis forces vector (with dimension $n \times 1$), $\mathbf{q}$ are the components of the position vector of the joints, $\dot{\mathbf{q}}$ are the components of the velocity vector of the joints, $\mathbf{G}$ is the gravitational force vector (with dimension $n \times 1$), $\ddot{\mathbf{q}}$ is the acceleration of the joints vector (with dimension $n \times 1$), and $\mathbf{F}$ is the friction forces vector (with dimension $n \times 1$).

Therefore, using (12), (13) and (14) we get the dynamic model of the redundant robotic manipulator, which can be expressed by means of Eqs. (15) through (32):

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

$$M_{11} = \sum_{i=1}^{n} m_i, \quad M_{12} = M_{21} = M_{31} = M_{32} = M_{41} = M_{42} = 0, \quad M_{15} = \sum_{i=1}^{n} m_i, \quad M_{22} = \sum_{i=1}^{n} I_{2i}, \quad M_{23} = \sum_{i=1}^{n} I_{2i} c_i, \quad M_{24} = \sum_{i=1}^{n} I_{2i} l_{i4}, \quad M_{25} = \sum_{i=1}^{n} I_{2i} l_{i5}, \quad M_{33} = \sum_{i=1}^{n} I_{3i}, \quad M_{34} = \sum_{i=1}^{n} I_{3i} c_i, \quad M_{35} = \sum_{i=1}^{n} I_{3i} l_{i4}, \quad M_{44} = \sum_{i=1}^{n} I_{4i}, \quad M_{55} = \sum_{i=1}^{n} I_{5i} \quad (16)-(32)$$

$$\mathbf{C} = \begin{bmatrix} C_{11} & C_{21} & C_{31} & C_{41} & C_{51} \end{bmatrix}^T \quad (26)$$

$$C_{11} = C_{15} = 0 \quad (27)$$

$$C_{12} = \left( l_2 s_3 \theta_2 \theta_3 + 2 l_2 s_3 \theta_2 \theta_3 \right) (l_3 m_3 + l_3 m_4 + l_3 m_5) + \ldots \quad (28)$$

$$C_{13} = \left( l_2 s_3 \theta_2 \theta_3 - 2 l_2 s_3 \theta_4 \theta_5 \right) (l_4 m_4 + l_4 m_5) + \ldots \quad (29)$$

$$C_{14} = \left( l_2 s_3 \theta_2 \theta_3 - 2 l_2 s_4 \theta_4 \theta_5 \right) (l_4 m_4 + l_4 m_5) + \ldots \quad (30)$$

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & -m_5 g \end{bmatrix}^T \quad (31)$$

$$\mathbf{F} = \begin{bmatrix} F_{11} & F_{21} & F_{31} & F_{41} & F_{51} \end{bmatrix}^T \quad (32)$$

According to this, and using the Kronecker product, matrix $\mathbf{V}_m$ is calculated as expressed by Eqs. (34) through (43):

$$\mathbf{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}_m(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{F}(\dot{\mathbf{q}}) \quad (33)$$
Claudio Urrea and John Kern

\[ V_{m33} = - (l_c + m_4 + l_m + m_3) j l_s + \dot{\theta}_4 \] (40)

\[ V_{m34} = - (l_c + m_4 + l_m + m_3) j l_s (\dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4) \] (41)

\[ V_{m42} = (l_2 s_{34} + l_3 s_{44}) 2 + l_1 s_{44} (l_4 + m_3 + l_4 m_2) \] (42)

\[ V_{m43} = (l_c + m_4 + l_m + m_3) j l_s (\dot{\theta}_2 + \dot{\theta}_3) \] (43)

4. Actuator Dynamics

The actuators considered in this study correspond to an analogic servo motors. Fig. 5 shows a schematic of a servo motor coupled with a robotic manipulator as the load [23]. These systems are constituted by a dc motor, a set of gears to reduce the rotational speed and increase the torque on its drive shaft, a potentiometer connected to that output shaft, which is used to know the position, and a feedback control circuit that converts an input signal of the PWM (Pulse-Width Modulation) type to voltage, comparing it with the fed back position and then amplifying it and activating an H bridge to cause a turn at a given speed [5].

Fig. 6 shows a block diagram of an analogic servo motor connected with a load consisting of a robot.

The dynamic model of the servo motors considered has been developed by the authors [23] and is given by (44) and (45):

\[ \tau = \left( \frac{k_a}{R_a} \right) A k_p q_1 + J_m q_1 n - \cdots \] (44)

\[ f_{e1} (\dot{q}) = F_{e1} \tanh (k_q) (1 + \text{sgn}(\dot{q}))/2 + \cdots \] (45)

\[ f_{e2} \tanh (k_q)(1 - \text{sgn}(\dot{q}))/2 \]

where \( n \) represents the gear ratio \((n_1/n_2)\), \( k_m \) is the motor's torque constant, \( R_a \) is the armature resistance, \( A \) is the current amplifier gain (H bridge), \( k_i \) is the sensitivity of the comparator, \( k_p \) is the total gain of the PWM conversion \((k_{p1} \cdot k_{p2})\), \( v_i \) is the input voltage to the servo motor, \( J_m \) is the moment of inertia of the motor, \( k_b \) is the inverse electromotive force constant, \( B_m \) is the viscous friction of the motor, \( p \) is the gain of the position potentiometer, and \( k \) is the gain of the slope of the tanh function used to increase or reduce the slope of the curve as it crosses zero.

5. Controllers Used

Below we present a summary of the controllers considered for the evaluation of the robot model together with its actuators and their corresponding performance.

5.1 Hyperbolic sliding mode controller

Sliding Mode Control (SMC) systems correspond to a particular type of Variable Structure Control (VSC) systems whose characteristic is to change their structure, by means of some law, in order to satisfy desired characteristics [24, 25]. The SMC consists in defining a control law that, commuting at a high frequency, succeeds in taking the state of a system to a surface called sliding surface, and once there, keeping it away from possible external perturbations [26]. One of the advantages of the sliding mode control is its invariance against parameter uncertainties and external perturbations. However, the high commutation frequencies cannot be implemented [27], and they also introduce the vibration phenomenon of “chattering” in the actuators, which must be avoided in many physical systems like servo control systems, structure vibration control systems, and robotic systems [28, 29]. That is why a modification is introduced in the classical SMC through a hyperbolic tangent function, with the purpose of reducing its abrupt commutation characteristic, as shown in (46),

\[ \tau = - K \cdot \tanh (a \cdot s) \] (46)

where \( K = \text{diag}(K_1, K_2, \ldots, K_n) \) and \( a = \text{diag}(a_1, a_2, \ldots, a_n) \) are positive-definite diagonal matrices (with dimensions \( n \times n \)). The sliding surface is given by:

\[ s = W \cdot (q - q_d) + (q - q_4d) \] (47)

where \( W = \text{diag}(W_1, W_2, \ldots, W_n) \) corresponds to a positive-definite diagonal matrix (with dimension \( n \times n \)), and \( q_d \) and \( q_4d \) represent the desired position and velocity vectors of the joints (with dimensions \( n \times 1 \)), respectively.

5.2 Control with Learning

Control with learning is based on the correction of the
control system through successive repetitions of the operations in order to compensate for the model’s uncertainties. In this way, a first control torque is generated, estimating that one part of the model is known, and a second control torque is generated from a model that is adjusted by means of a learning law, in successive repetitions of the same operation. A control scheme that considers Proportional and Derivative (PD) terms, and terms dependent on the known model, are shown in Eq. (48) [30]:

$$\tau = \hat{M}(q)(\dot{q}_d + K_e \dot{e} + K_p e) + \hat{C}(q, \dot{q}) + \hat{G}(q) + \gamma_k$$  \hspace{1cm} (48)$$

where $\hat{M}$ expresses the estimation of the inertia matrix (with dimension $n \times n$), $\hat{C}$ is the estimation of the centrifugal and Coriolis forces vector (with dimension $n \times 1$), $\hat{G}$ is the estimation of the gravitational force vector (with dimension $n \times 1$), $\dot{q}_d$ is the desired acceleration vector of the joints (with dimension $n \times 1$), $\gamma_k$ is the torque obtained by learning in successive repetitions of the motion ($k = 1, 2, \ldots$), and $K_e = \text{diag}(K_{e1}, K_{e2}, \ldots, K_{en})$ and $K_p = \text{diag}(K_{p1}, K_{p2}, \ldots, K_{pn})$ are derivative and proportional gain matrices, and positive-definite diagonal matrices (with dimensions $n \times n$), respectively.

The position error, $e$, and velocity error, $\dot{e}$, vectors in terms of the manipulator’s joint coordinates are expressed by (49) and (50), respectively:

$$e = q_d - q$$ \hspace{1cm} (49)$$

$$\dot{e} = q_{\dot{d}} - \dot{q}$$ \hspace{1cm} (50)$$

Using the dynamic model indicated in Eq. (14) and substituting in (48) we get

$$\dot{e} + K_e \dot{e} + K_p e = \hat{M}^{-1}(q) (F(\dot{q}) - \gamma_k),$$ \hspace{1cm} (51)$$

where the acceleration error vector, in terms of the manipulator’s joint coordinates, is expressed by (52):

$$\ddot{e} = q_{\ddot{d}} - \ddot{q}$$ \hspace{1cm} (52)$$

Making the substitutions $D = \hat{M}^{-1} F(\dot{q})$ and $\hat{D}_k = \hat{M}^{-1} \gamma_k$, we get (53):

$$\ddot{q}_e + K_e \dot{q}_e + K_p q_e = D - \hat{D}_k$$ \hspace{1cm} (53)$$

Eq. (54) is an expression for the robot’s joints $i$:

$$\ddot{q}_i + K_{ei} \dot{q}_i + K_{pi} q_i = D_i - \hat{D}_{ik}$$ \hspace{1cm} (54)$$

where $D_i$ is a function of the time that it remains constant in the repetitions and $\hat{D}_{ik}$ represents a torque that is adjusted in each repetition $k$. In [30] a learning law is proposed for joint $i$ which considers the convolution between the response impulse $P$ of a filter and the error $q_{eik}$ in iteration $k$, according to (55):

$$\hat{D}_{ik+1} = \hat{D}_{ik} + P \cdot q_{eik}$$ \hspace{1cm} (55)$$

Combining the Laplace transforms of Eqs. (54) and (55) we get:

$$\hat{D}_{ik+1}(s) = G(s) \hat{D}_{ik}(s) + D_i(s)(1 - G(s))$$ \hspace{1cm} (56)$$

where

$$G(s) = 1 - P(s)\left(s^2 + K_{vi}s + K_{pi}\right)$$ \hspace{1cm} (57)$$

After $k$ iterations, starting with $\hat{D}_{i0} = 0$, we have

$$\hat{D}_{ik}(s) = D_i(s) + CG^k(s)$$ \hspace{1cm} (58)$$

where $C$ is a constant. If factor $G^2(s)$ tends to zero in the successive iterations, the convergence of the learning algorithm takes place, making sure that $\hat{D}_{i}(s) \rightarrow D_i(s)$. If $P(s)$ is chosen appropriately, e.g.,

$$P(s) = s^2 + (K_{vi} - \mu)s + (K_{pi} - \mu)$$ \hspace{1cm} (59)$$

where $\mu$ is a constant, it is possible to achieve the convergence of the algorithm, provided that:

$$\lim_{k \rightarrow \infty} \|g_k(t)\| = 0$$ \hspace{1cm} (60)$$

where $g_k(t)$ is equivalent to the inverse Laplace transform of $G^k(s)$.

5.3 Adaptive control

Adaptive control has the purpose of getting the correct performance of the robotic system in spite of the many uncertainties related to different aspects of the manipulator, e.g., the flexibility of the links and joints, external perturbations, the dynamics of the actuators, friction at the joints, the noise of sensors, and in other not modeled dynamic behaviors. In that control the parameters are variables that are estimated online and are adjusted through a mechanism based on the system’s measurements [26].

The adaptive control considered is based on a law of control presented in [15, 31, 32], for which it is necessary to express the dynamic model of the manipulator as indicated in (33) and to define an auxiliary error signal $r = A e + \ddot{e}$ and its derivative $\dot{r} = A \dot{e} + \dddot{e}$ with respect to time, where $A = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ corresponds to a positive-definite diagonal matrix (with dimension $n \times n$). When $r$ and $\dot{r}$ are combined with expression (33) we get:

$$\tau = Y(\cdot)(\varphi - M(q)\dot{r} - V_m(q, \dot{q})r)$$ \hspace{1cm} (61)$$
where

$$Y(q, \dot{q}, q_d, \dot{q}_d, \dot{q}_d) = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1n} \\ Y_{21} & Y_{22} & \cdots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{nn} \end{bmatrix}$$ (62)

represents a parameter regression matrix (with dimension \(n \times n\)) and \(\phi\) is a parameter vector (with dimension \(n \times 1\)). The forms of the control torque and of the updating rule are expressed by (63) and (64), respectively:

$$\tau = Y(\cdot)\dot{\phi} + K_r r$$ (63)
$$\dot{\phi} = \Gamma Y^T (\cdot) r$$ (64)

where \(\dot{\phi}\) indicates a parameter estimation vector (with dimension \(n \times 1\)) and \(\Gamma = \text{diag}(\gamma_1, \gamma_2, \ldots, \gamma_n)\) is a positive-definite diagonal matrix (with dimension \(n \times n\)). The final adaptive control is calculated using \(r\), (63) and (64), as shown in (65):

$$\tau = Y(\cdot)\int \Gamma Y^T (\cdot)(\Lambda e + \dot{e}) + K_v \Lambda e + K_v \dot{e}$$ (65)

6. Simulation and Experimentation

The three control laws mentioned above, are executed in: a simulation structure (together with the dynamic model of the SCARA type redundant manipulator and the actuator dynamics) and a real redundant manipulator of the SCARA type carried out using MatLab/Simulink programming tools like that shown in Fig. 7 and Fig. 8, respectively. The values of the parameters considered in the manipulator are shown in Table 2. Table 3 shows the set of values of the

**Table 2. Parameters considered in the manipulator**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Link 1</th>
<th>Link 2</th>
<th>Link 3</th>
<th>Link 4</th>
<th>Link 5</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>(l)</td>
<td>0.524</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.14</td>
<td>[m]</td>
</tr>
<tr>
<td>(l_c)</td>
<td>-</td>
<td>0.0229</td>
<td>0.0229</td>
<td>0.0229</td>
<td>-</td>
<td>[m]</td>
</tr>
<tr>
<td>(m)</td>
<td>1.228</td>
<td>1.023</td>
<td>1.023</td>
<td>1.023</td>
<td>0.5114</td>
<td>[kg]</td>
</tr>
<tr>
<td>(I_z)</td>
<td>-</td>
<td>0.0058</td>
<td>0.0058</td>
<td>0.0058</td>
<td>0.0058</td>
<td>[kg⋅m²]</td>
</tr>
<tr>
<td>(F_v)</td>
<td>0.03</td>
<td>0.025</td>
<td>0.025</td>
<td>0.025</td>
<td>0.02</td>
<td>[N⋅m/s]</td>
</tr>
<tr>
<td>(F_{v\alpha})</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.03</td>
<td>[N⋅m]</td>
</tr>
<tr>
<td>(F_{v\beta})</td>
<td>-0.05</td>
<td>-0.05</td>
<td>-0.05</td>
<td>-0.05</td>
<td>-0.03</td>
<td>[N⋅m]</td>
</tr>
</tbody>
</table>

**Table 3. Parameters considered in the servo motors**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Servo 1</th>
<th>Servos 2-3-4</th>
<th>Servo 5</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R_r)</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
<td>[Ω]</td>
</tr>
<tr>
<td>(L_r)</td>
<td>0.0048</td>
<td>0.0048</td>
<td>0.0048</td>
<td>[H]</td>
</tr>
<tr>
<td>(J_m)</td>
<td>0.007</td>
<td>0.007</td>
<td>0.007</td>
<td>[kg⋅m²]</td>
</tr>
<tr>
<td>(B_m)</td>
<td>0.01413</td>
<td>0.01313</td>
<td>0.01208</td>
<td>[N⋅m/s]</td>
</tr>
<tr>
<td>(k_c)</td>
<td>0.35</td>
<td>0.35</td>
<td>0.35</td>
<td>[N/m]</td>
</tr>
<tr>
<td>(k_s)</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>[V⋅s/rad]</td>
</tr>
<tr>
<td>(J_{el})</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>[N⋅m]</td>
</tr>
<tr>
<td>(F_{eca})</td>
<td>-0.05</td>
<td>-0.05</td>
<td>-0.03</td>
<td>[N⋅m]</td>
</tr>
<tr>
<td>(n)</td>
<td>1/600</td>
<td>1/561.6</td>
<td>1/561.6</td>
<td>[Times]</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>[Times]</td>
</tr>
<tr>
<td>(k_c)</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>[Times]</td>
</tr>
<tr>
<td>(k_s)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>[Times]</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>[Times]</td>
</tr>
<tr>
<td>(p)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>[Times]</td>
</tr>
</tbody>
</table>

**Table 4. Gains considered in the controllers**

<table>
<thead>
<tr>
<th>Controller</th>
<th>Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyperbolic Sliding Mode</td>
<td>(K_t, \ldots, K_t), (W_t, \ldots, W_t), (\alpha_t, \ldots, \alpha_t)</td>
</tr>
<tr>
<td>Learning</td>
<td>(K_{\alpha\alpha}, \ldots, K_{\alpha\alpha}), (K_{\alpha\beta}, \ldots, K_{\alpha\beta}), (\mu_{\alpha\alpha}, \ldots, \mu_{\alpha\alpha})</td>
</tr>
<tr>
<td>Adaptive Inertia</td>
<td>(K_{t\alpha}, \ldots, K_{t\alpha}), (\lambda_{\alpha\alpha}, \ldots, \lambda_{\alpha\alpha}), (\gamma_{\alpha\alpha}, \ldots, \gamma_{\alpha\alpha})</td>
</tr>
</tbody>
</table>

**Fig. 7.** Block diagram of the simulator used to test the model of the manipulator together with the control laws mentioned

**Fig. 8.** Real redundant manipulator of the SCARA type used to run the control laws mentioned

**Fig. 9.** Communication interface
and establishing the control laws to be used, a test trajectory in space was determined to make the manipulator model follow it, and then study the results as a function of the performance of each controller. That trajectory is shown in Fig. 11.

Fig. 12 and Fig. 13 show the curves corresponding to the desired and real simulated and desired and real experimental joint trajectories, respectively, using the hyperbolic sliding mode controller, where $q_{d1}$ and $q_{d2}$ represent the desired and real joint trajectories ($n$ indicates joints 1 through 5).

Fig. 14 and Fig. 15 show the graphs of the errors obtained from the desired and real simulated, and desired and real experimental joint trajectories, respectively, using the hyperbolic sliding mode controller; $e_n$ express the errors in the joint trajectories ($n$ represents joints 1 through 5).

Fig. 16 and Fig. 17 show the graphs related to the desired and real simulated and desired and real experimental joint trajectories using the hyperbolic sliding mode controller.

7. Results

After developing the manipulator model and the simulation environment, incorporating the actuator dynamics, parameters used for each actuator. Table 4 shows the set of values of the gains used for each type of controller.

Fig. 9 and Fig. 10 show the communication interface, and the signal conditioning circuit, respectively, used to run in real time the control algorithms on the real redundant manipulator of the SCARA type.
trajectories, respectively, using the controller with learning. The errors produced from the difference between the simulated desired and real, and experimental desired and real joint trajectories, applying the controller with learning, are shown in Fig. 18 and Fig. 19, respectively.

Fig. 15. Joint trajectory error using the hyperbolic sliding mode controller (experimental)

Fig. 16. Comparison of the desired and real joint trajectories using the controller with learning

Fig. 17. Comparison of the desired and real joint trajectories using the controller with learning (experimental)

Fig. 18. Joint trajectory error using the controller with learning

Fig. 19. Joint trajectory error using the controller with learning (experimental)

Fig. 20. Comparison of the desired and real joint trajectories using the adaptive controller
The comparisons of the desired and real simulated and desired and real experimental joint trajectories, as a function of the adaptive controller, are shown in Fig. 20 and 21, respectively.

Fig. 22 and Fig. 23 show the error curves obtained for the desired and real joint trajectories: simulated and experimental, respectively, using the adaptive controller.

Fig. 24 and Fig. 25 show the simulated and experimental performance index corresponding to the joint’s trajectory.

Fig. 26. Performance index corresponding to the joint’s trajectory (simulated and experimental)
RMS errors, respectively, according to Eq. (66).

\[
\text{RMS} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} e_i^2}
\]  

(66)

where \( e_i \) represents the joint as well as the Cartesian errors of the trajectory, and \( n \) is the number of data.

Finally, Fig. 26 shows a comparison between simulated and experimental RMS errors.

8. Conclusion

A kinematic and dynamic model of a redundant robot with five degrees of freedom of the SCARA manipulator type using Denavit-Hartenberg and geometric methods, and Lagrange-Euler methods, respectively, was developed. Three controllers were made: hyperbolic sliding mode, with learning, and adaptive. A simulator was made using the MatLab / Simulink software. A 5 DOF robot, a communication interface and a signal conditioning circuit are designed and implemented for feedback. The tests of the manipulator model and the real manipulator were presented, including the dynamics of the actuators and together with each controller, by following a test trajectory composed of a spiral in the Cartesian space.

The results, obtained through a simulation environment and implementation, were represented by comparative curves and RMS indices of the joint, and the different control laws involved, new fault tolerant control strategies for redundant robot manipulators have started.

Future Work

From the performance achieved in this work, through the simulation and implementation tests, the redundant robot and the different control laws involved, new fault tolerant control strategies for redundant robot manipulators have started.

References


Claudio Urrea He received the M.Sc. Eng. and the Dr. degrees from Universidad de Santiago de Chile, Santiago, Chile in 1999, and 2003, respectively; and the Ph.D. degree from Institut National Polytechnique de Grenoble, France in 2003. Ph.D. Urrea is currently Professor at the Department of Electrical Engineering, Universidad de Santiago de Chile, from 1998. He has developed and implemented a Robotics Laboratory, where intelligent robotic systems are development and investigated. He is currently Director of the Doctorate in Engineering Sciences, Major in Automation, at the Universidad de Santiago de Chile.

John Kern He received the M.Sc. Eng. and the Dr. degrees from Universidad de Santiago de Chile, Santiago, Chile in 2010 and 2013, respectively. John Kern is currently Professor of electronic engineering in the areas of automatic control and robotics. He has developed laboratories: theory and control systems, signals and communication system, and electronics.