



Original

Parameter identification methods for real redundant manipulators

Claudio Urrea*, José Pascal

Grupo de Automática, Department of Electrical Engineering, Universidad de Santiago de Chile, Chile

Received 9 April 2016; accepted 16 February 2017

Available online 31 July 2017

Abstract

This work presents the development, assessment and comparison of four techniques for identifying dynamic parameters in an industrial redundant manipulator robot with 5 degrees of freedom. Based on the Lagrange–Euler formulation, a linear model of the robot with unknown parameters is obtained. Then, these parameters are identified using the following techniques: least squares, artificial Adaline neural networks, artificial Hopfield neural networks and extended Kalman filter. The parameters identified are validated by using them for computationally simulating the performance of the redundant manipulator robot, to which are imposed reference trajectories different from the ones used in the estimation. To relate the trajectories performed by the redundant manipulator robot with the estimated parameters, the following error indexes are calculated: Residual Mean Square, Residual Standard Deviation and Agreement Index. Finally, to determine the sensitivity of the model identified – due to the variations of the estimated parameters – a new simulation is conducted on the robot, considering that its parameters vary in a restricted range. In addition, the RMS error index of the trajectories performed is determined. After this step, the parameters of the redundant manipulator robot were successfully identified and, thus, its mathematical model was known.

© 2017 Universidad Nacional Autónoma de México, Centro de Ciencias Aplicadas y Desarrollo Tecnológico. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Keywords: Robot dynamics; System identification; Mathematical model; Error indices

1. Introduction

Four closely related types of problem may be distinguished in the systems theory, namely modeling, analysis, estimation and control (Kessman, 2011). Modeling and estimation are extremely important to system identification. Modeling is a critical step when applying the systems theory to real processes aimed at obtaining a mathematical model that adequately describes a physical situation to be represented. To achieve this goal, several specifications should be considered, such as the limits and variables of the system. Then, the relationships between these variables need to be specified based on previous knowledge as well as on the assumptions about the uncertainties of the model, which characterize its structure. On the other hand, after obtaining an observable and identifiable model, the estimation of its unknown variables is addressed

using an input/output dataset. Basically, there is a distinction between states and parameter estimation. Therefore, systems identification consists in experimentally determining a temporary behavior of a process or system, based on a mathematical model and using measured signals to determine such a behavior. The error (deviation) between the real process and its mathematical system has to be minimized to represent the real process or system in the best possible way (Isermann & Münchhof, 2011).

In the identification process, data characterizing the dynamics of the real process or system is first obtained. Then, a set of models is chosen and, finally, the best mathematical model of the set is selected. It should be noted that a mathematical model may be deficient due to reasons such as (Ljung, 1987):

- The failure in the numerical procedure conducted to find the best mathematical model, given a specific criterion.
- Ill-chosen criterion.
- No mathematical model satisfactorily describes the system.
- The information (data) is not sufficient for the selection of an appropriate mathematical model.

* Corresponding author.

E-mail address: claudio.urrea@usach.cl (C. Urrea).

Peer Review under the responsibility of Universidad Nacional Autónoma de México.

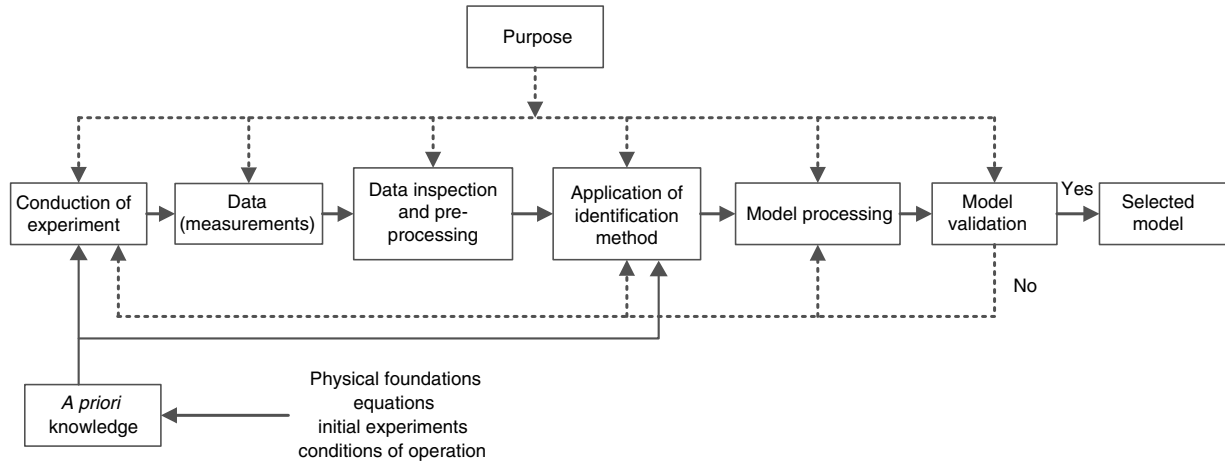


Fig. 1. System identification scheme (Isermann & Münchhof, 2011).

In summary, the general problem of systems identification may be expressed as: once known the structure and the value of some of the input and output variables of the system under study, it is necessary to determine the set of parameters that better adjusts to these measures (Anríquez, 2011). Figure 1 shows the general scheme of the identification of one system.

2. Methods for parameter identification

The methods for parameter identification applied to the redundant manipulator robot are explained below:

2.1. Least squares

Among the parameter estimation methods, least squares is widely used in the identification of manipulator robots (Bona & Curatella, 2005), since it is based on the use of a linear model inverse with respect to the parameters. This method allows estimating base inertial parameters from the measurement or estimation of torques and joint positions, optimizing the root mean square error of the model under the assumption that measurement errors are negligible. However, one problem in the application of this method is the noise sensitivity present in the measured data (Kessman, 2011), because noise will limit the accuracy of the obtained parameters as well as the convergence rate of the algorithm. To overcome this problem, it is suggested to generate excitation trajectories or to apply a filter to data (Ha, Ko, & Kwon, 1989).

The dynamic model may be written as a set of unknown parameters linear equations, as shown in Eq. (1):

$$\tau = \Phi(q, \dot{q}, \ddot{q}) \cdot \lambda \quad (1)$$

where τ corresponds to the generalized forces vector, q to the generalized coordinates vector, λ to the vector or unknown dynamic parameters and Φ to the observation matrix or regressor. By applying the identification model to the model shown in (1), the linear equation system in λ is obtained as:

$$\tau = \Phi(q, \dot{q}, \ddot{q}) \cdot \lambda + \rho \quad (2)$$

where ρ is the residual error vector. The estimated $\hat{\lambda}$ value may be obtained as:

$$\hat{\lambda} = \min \|\rho\|^2 \quad (3)$$

which is the same as calculating:

$$\hat{\lambda} = (\Phi^T \Phi)^{-1} \Phi^T \tau \quad (4)$$

2.2. Adaline neural networks

A neural network consists of a set of simple processing units, which communicate by sending signals to one another through weighted connection. Adaline (adaptive linear element) is one type of neural network. For a one-layer network with one output and one linear activation function, the output is given by Eq. (5):

$$y = \sum_{i=1}^n w_i x_i + \theta \quad (5)$$

This network is able to represent a linear relation between the values of the output and input units. The purpose of this network is to produce a given value $y = d^p$ in the output when the set of values x_i^p , $i = 0, 1, \dots, n$, is applied to the inputs. It is intended to determine the coefficients (weights) w_i , $i = 0, 1, \dots, n$, so the input–output response is correct for a large number of arbitrarily chosen signal sets. In this case, the “delta rule” to adjust weights is used for Adaline, training the network in such a way that the best adjustment for each set of data composed of the input values x_i^p and desired outputs d^p is obtained. For each input datum, the network output differs from the desired value by $d^p - y^p$, where y^p corresponds to the current output for this pattern. The “delta rule” uses a cost or error function based on this difference to adjust the weights. The error function is given by the mean squared error. Therefore, the total error E is defined by:

$$E = \sum_p E^p = \frac{1}{2} \sum_p (d^p - y^p)^2 \quad (6)$$

where the p index extends over the set of input patterns and E^p represents the error over the p pattern. This procedure allows for

finding the value of all weights that minimize the error function by means of the gradient descent method:

$$\Delta_p w_i = -\gamma \frac{\partial E^p}{\partial w_i} \quad (7)$$

where γ is a constant of proportionality and, also:

$$\frac{\partial E^p}{\partial w_i} = \frac{\partial E^p}{\partial y^p} \frac{\partial y^p}{\partial w_i} \quad (8)$$

Due to the linear relationship showed in (5):

$$\frac{\partial y^p}{\partial w_i} = x_i \quad (9)$$

and

$$\frac{\partial E^p}{\partial y^p} = -(d^p - y^p) \quad (10)$$

Finally, we obtain:

$$\Delta_p w_i = \gamma \delta^p x_i \quad (11)$$

where $\delta^p = (d^p - y^p)$ is the difference between the desired output and the current output for the p pattern.

2.3. Hopfield neural networks

Hopfield neural networks or HNNs can be used for estimating parameters on-line. In Hopfield's formulation, the dynamic of neuron i is described by the ordinary differential equation (12) (Alonso, Mendonça, & Rocha, 2009):

$$\frac{du_i}{dt} = -\frac{1}{C_i} \left(\frac{1}{R_i} u_i(t) + \sum_j W_{ij} f_i(u_j(t)) + I_i \right) \quad (12)$$

where u_i is the neuron input, f_i is a strictly increasing, bounded, nonlinear continuous function, and C_i , R_i , W_{ij} and I_i are parameters that correspond to a capacitance, a resistance, the weight associated to the connection of the j neuron with the i neuron, and the i neuron external input or bias. In Abe's formulation, it is assumed that $C_i = 1$ and $R_i = \infty$.

The considered system is linear with respect to the parameters, i.e. it has the form shown in Eq. (1), which (1) works under the following assumptions:

- (a) $\tau, \Phi \in C^1$.
- (b) τ, Φ are bounded.
- (c) $\tau(t), \Phi(t)$ are known in each t instant.
- (d) $\lambda \in]-c, c[$ for some $c > 0$ unknown.

Given Abe's formulation, considering a network with N neurons, u_i denotes the total input of the i neuron, its derivative with respect to time is:

$$\frac{du_i(t)}{dt} = - \left(\sum_{j=1}^N W_{ij}(t) S_j(t) + I_i(t) \right) \quad (13)$$

where S_j represents the state (or output) of the j neuron. The input-state relation for the i neuron is given by:

$$S_i(t) = \alpha \tanh \left(\frac{u_i(t)}{\beta} \right) \quad (14)$$

where $\alpha, \beta > 0$ and, thus, $\forall i \in N, t \geq t_0 \quad S_i(t) \in]-\alpha, \alpha[$.

Using matrix notation, the network is as follows:

$$\frac{du}{dt} = -(W(t)S(t) + I(t)) \quad (15)$$

$$S(t) = \alpha \tanh \left(\frac{u(t)}{\beta} \right) \quad (16)$$

where $S(t), I(t) \in \mathbb{R}^{N \times 1}$ and $W \in \mathbb{R}^{N \times N}$, under the state-space representation is:

$$\frac{dS}{dt} = -\frac{1}{\alpha\beta} D_\alpha(S(t)) (W(t)S(t) + I(t)) \quad (17)$$

where

$$D_\alpha(S(t)) = \text{diag} \left(\left(\alpha^2 - S_i(t)^2 \right)_{i \in N} \right) \quad (18)$$

This a nonlinear and non-autonomous dynamic system whose architecture is completely determined by the N number of neurons, with characteristics given by α, β, W and I .

This work considers a HNN whose state $S(t)$ over time t is taking as the estimation $\hat{\lambda}(t)$ of the λ parameter, in other words, $\hat{\lambda}(t) = S(t)$; implicitly, there is a network with as many neurons as parameters to estimate. Then, taking $\alpha = c$, Eq. (14) guarantees that the trajectory generated for the network is in a feasible region of the estimation problem. Therefore, from the representation of states in Eq. (17), it is considered that: $W = \Phi(t)^T \Phi(t)$ and $I = -\Phi(t)^T \tau$. In this way, the dynamic of the HNN to be used as an on-line estimation algorithm is defined by:

$$\frac{d\hat{\lambda}}{dt}(t) = \frac{1}{c\beta} D_c(\hat{\lambda}(t)) \Phi(t)^T (\tau - \Phi(t)\hat{\lambda}(t)) \quad (19)$$

The advantage of using HNN to resolve the estimation problem is that inverting $\Phi(t)^T \Phi(t)$, is not necessary, since the latter might be ill-conditioned. In addition, given its structure, prior knowledge is not required for selecting $\hat{\lambda}(t_0)$, i.e. of the initial estimation of λ .

Finally, the λ parameter is a globally, uniformly and asymptotically stable balance point of this HNN if the following sustains:

$$I \subset [t_0, +\infty[, \bigcap_{t \in I} \ker(\Phi(t)) = \{0\} \quad (20)$$

2.4. Extended Kalman filter

The extended Kalman filter is based on the direct dynamic model, which is nonlinear with respect to state and parameters. In this algorithm, physical parameters are considered under the

use of an extended state. Kalman filter intends to estimate the $x \in \mathbb{R}^n$ state of a process controlled in discrete time (Welch & Bishop, 2006). The algorithm has two steps that are executed in an iterative way, namely prediction, and correction or update of the state. However, when a system has non-linear characteristics, the formulation should be modified. This change in the algorithm is known as extended Kalman filter. In this case, the process will be represented by the following non-linear equations:

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} \quad (21)$$

$$z_k = h(x_k) + v_k \quad (22)$$

where f corresponds to a non-linear function that allows for knowing the relation between the previous step state vector $k-1$ and the u input with the current state vector k , according to the following procedure:

1. Prediction stage

1.1 Prediction of the state:

$$\hat{x}_k^- = f(x_{k-1}, u_{k-1}) \quad (23)$$

1.2 Prediction of error covariance matrix:

$$P_k^- = F_{k-1} P_{k-1} F_{k-1}^T + Q \quad (24)$$

2. Correction or updating stage

2.1 Kalman gain calculation

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R]^{-1} \quad (25)$$

2.2 State update with measures Z_k :

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-)) \quad (26)$$

2.3 Covariance matrix update:

$$P_k = (I - K_k H_k) P_k^- \quad (27)$$

where F_{k-1} and H_k corresponds the Jacobian matrices described in Eqs. (28) and (29), respectively:

$$F_{k-1} = \left. \frac{\partial f}{\partial x} \right|_{x_{k-1}, u_{k-1}} \quad (28)$$

$$H_k = \left. \frac{\partial h}{\partial x} \right|_{\hat{x}_k^-} \quad (29)$$

where Q and R are the process and observation noise covariance matrices, respectively. In addition, they are diagonal and symmetric. On the other hand, $\hat{x}_k^- \in \mathbb{R}^n$ are the states estimated *a priori* at k instant, given the process behavior and $\hat{x}_k \in \mathbb{R}^n$ the *a posteriori* estimated state at k instant, given the measure z_k .

3. Model of the industrial redundant manipulator robot

The manipulator robot used in this study is redundant, since it has five degrees of freedom (DOF), and a PRRRP configuration. Figure 2 shows a scheme of manipulator robot and its coordinate assignment. In addition, its Denavit–Hartenberg parameters are presented in Table 1.

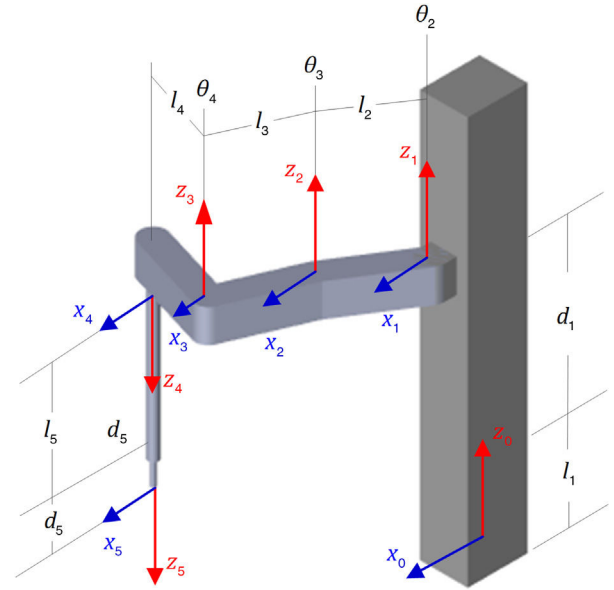


Fig. 2. Assignment of coordinate systems of the PRRRP manipulator robot.

Table 1
Devanavit–Hartenberg parameters.

Articulation	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°

The Lagrange–Euler formulation describes the behavior of a dynamic system in terms of work and energy stored in the system. The Lagrangian L is defined as:

$$L(q_i, \dot{q}_i) = T - U \quad (30)$$

From this Lagrangian, the dynamic system equations of motion are given by:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i \quad (31)$$

where L is the Lagrangian function, T the kinetic energy, U the potential energy, q_i the generalized coordinates, and Q_i the generalized force applied to q_i . The general form of a manipulator robot dynamic model is shown in Annex A.

4. Simulation of the methods for parameter identification

Table 2 shows the dynamic parameters of each link of the redundant manipulator robot (mass, length, center of mass and inertia) (Urrea & Kern, 2016).

In this work, we consider the following methods for identifying the parameters of the redundant manipulator robot: least squares, Adaline neural networks, Hopfield neural networks and extended Kalman filter. A brief description of these methods is presented below:

Table 2
Parameters of the redundant manipulator robot parameters with 5 DOL.

Parameter	Link 1	Link 2	Link 3	Link 4	Link 5	Unit
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]

4.1. Least squares

The least squares procedure to estimate the parameters of each link of the redundant manipulator robot might be summarized in the following steps:

- Using some formulation, generate a robot model that is linear with respect to the inertial parameters.
- Reducing the inertial parameters to a set of base parameters.
- Determining the optimal trajectory of the parameters and optimize the excitation of the trajectories.

There is a number of methodologies to determine both the parameters influencing the systems dynamic and the set of minimum parameters required to identify the models of such systems (Choi, Yoon, Park, & Kim, 2011; Mayeda, Yoshida, & Ohashi, 1989; Mayeda, Yoshida, & Osuka, 1990; Radkhah, Kulic, & Croft, 2007). This work considers a numerical method that allows for identifying the set of independent parameters of the redundant manipulator robot, based on the analysis of the kernel of the regressor.

For a redundant manipulator robot with j links, ten inertial parameters are considered per link:

$$\xi_j = (m_j \quad p_{g,j}^T \quad v_j^T)^T \quad (32)$$

where $m_j, p_{g,j} = (x_{g,j} \quad y_{g,j} \quad z_{g,j})^T$ and $v_j = (I_{xx,j} \quad I_{yy,j} \quad I_{zz,j} \quad I_{xy,j} \quad I_{xz,j} \quad I_{yz,j})^T$ are the mass, the position of the center of gravity, and the vector with the elements of the inertia tensor of the j link, respectively. A linear form may be composed by grouping the parameters as follows:

$$\tau = \Phi(q, \dot{q}, \ddot{q}) \cdot \lambda(\bar{\xi})$$

$$\begin{pmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_n \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1n} \\ 0 & \phi_{22} & \dots & \phi_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \phi_{nn} \end{pmatrix} \begin{pmatrix} \lambda_1(\bar{\xi}) \\ \lambda_2(\bar{\xi}) \\ \vdots \\ \lambda_n(\bar{\xi}) \end{pmatrix} \quad (33)$$

where

$$\lambda_j(\bar{\xi}) = (m_j \quad m_j p_{g,j}^T \quad \pi_j^T \quad v_j^T)^T \in \mathbb{R}^{16}$$

$$\pi_j = m_j (x_{g,j}^2 \quad y_{g,j}^2 \quad z_{g,j}^2 \quad x_{g,j}y_{g,j} \quad y_{g,j}z_{g,j} \quad z_{g,j}x_{g,j})^T \quad (34)$$

The regressor Φ is not always linearly independent or has a full rank, thus there might not be a unique solution.

Table 3
Grouped parameters of the redundant manipulator robot.

Grouped parameters	Inertial parameters
λ_1^*	$m_1 + m_2 + m_3 + m_4$
λ_2^*	$m_2 l_{c2}^2 + m_3 l_{c2}^2 + m_4 l_{c2}^2 + m_5 l_{c2}^2 + I_{2zz}$
λ_3^*	$m_3 l_{c3}^2 + m_4 l_{c3}^2 + m_5 l_{c3}^2 + I_{3zz}$
λ_4^*	$m_3 l_{c3} l_2 + m_4 l_{c3} l_2 + m_5 l_{c3} l_2$
λ_5^*	$m_4 l_{c4}^2 + m_5 l_{c4}^2 + I_{3zz}$
λ_6^*	$m_4 l_{c4} l_2 + m_5 l_{c4} l_2$
λ_7^*	$m_4 l_{c4} l_3 + m_5 l_{c4} l_3$
λ_8^*	m_5

Nevertheless, it is possible to reduce the number of unknown parameters and transform the system into: $\tau = \Phi^* \cdot \lambda^*$, where λ^* is the vector for regrouped parameters. It is hard to find the linear relation between dependent and independent columns of the regressor, since dynamic relationships are complex. To find this relation, the kernel of the matrix is used, which denotes the relation between columns dependent and independent from the regressor in linear combinations. In this way, the new grouped parameters λ^* and the full rank matrix Φ^* may be found. The calculation is conducted in a recursive way, starting from the n joint up to the articulation 1.

In this case, the vector of regrouped parameters λ^* is composed of eight elements to be identified, which are presented in Table 3. The structure of the regressor $\Phi^* \in \mathbb{R}^{5 \times 8}$ is described in Annex B.

4.1.1. Excitation trajectories

When a system identification experiment is designed, it is necessary to consider a trajectory that sufficiently excites the redundant manipulator robot parameters during the movement. Otherwise, some parameters will be impossible to identify, or will become too sensitive to noise (Swevers, Ganseman, De Schutter, & Van Brussel, 1996). Meanwhile, since the regressor Φ^* depends on joint coordinates and its time derivatives, when the manipulator robot executes a particular trajectory $GDL \cdot n_{pts}$ equations will be obtained, generating an over-determined system. Thus, considering the points traveled in the trajectory, the system is given by the expression:

$$\tau = W \cdot \lambda^* \quad (35)$$

where τ is the torque/force vector $\in \mathbb{R}^{GDL \cdot n_{pts}}$, W is the $\in \mathbb{R}^{GDL \cdot n_{pts} \times n_p}$ system observation matrix, λ^* is the vector of the grouped parameters $\in \mathbb{R}^{n_p}$, n_{pts} is the number of samples and n_p corresponds to the number of parameters grouped. Then, to identify the parameters it is necessary to calculate:

$$\lambda^* = (W^T W)^{-1} W^T \tau \quad (36)$$

Parameters accuracy is closely related to the observation matrix W , so the purpose is to obtain a well-conditioned matrix that allows for enhancing the estimation results (Wu, Wang, & You, 2010). This implies finding the denominated “excitation trajectories”, which correspond to an optimization problem that

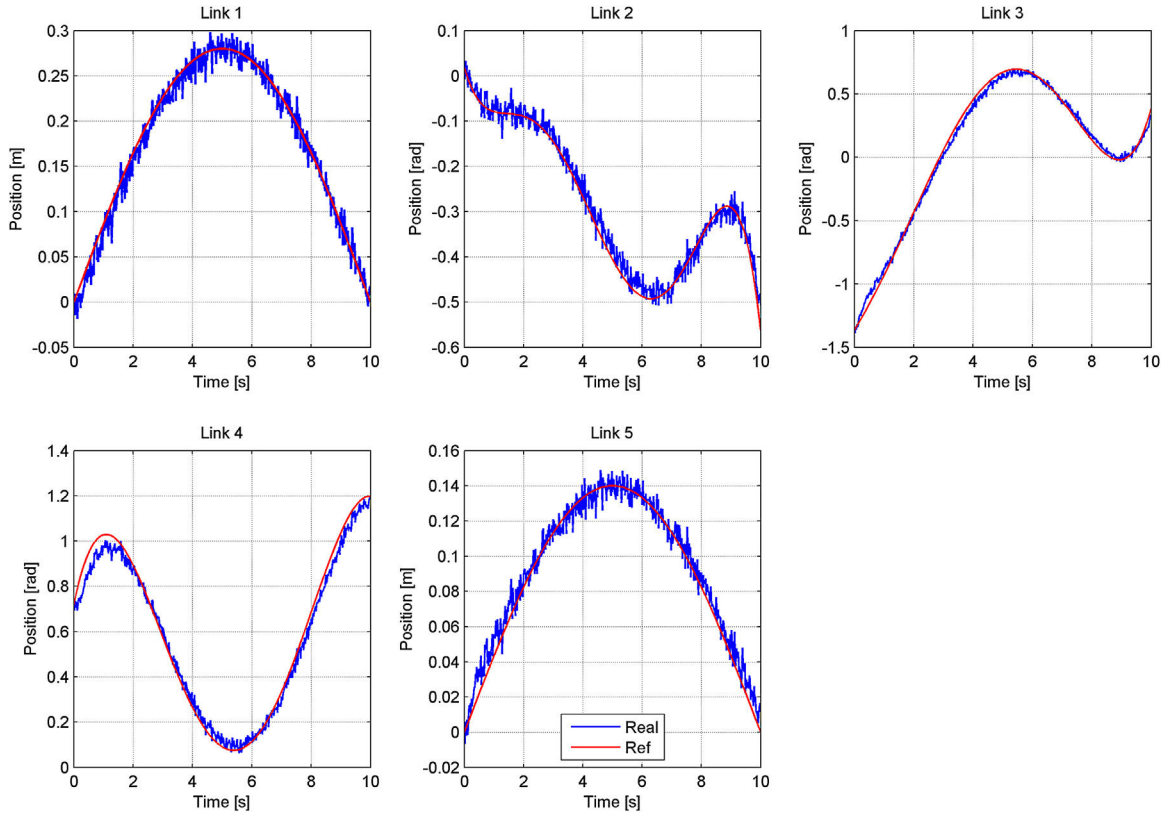


Fig. 3. Reference and output trajectories of the redundant manipulator robot with 5 DOL.

consists in minimizing a function like the one presented in (37) subjected to the constraints of Eq. (38):

$$\min f(q, \dot{q}, \ddot{q}) \quad (37)$$

$$q_{\min} \leq q \leq q_{\max}$$

$$\dot{q}_{\min} \leq \dot{q} \leq \dot{q}_{\max} \quad (38)$$

$$\ddot{q}_{\min} \leq \ddot{q} \leq \ddot{q}_{\max}$$

where function f satisfies the criterion $k(W) = (\sigma_{\max}/\sigma_{\min})$, q, \dot{q}, \ddot{q} are the generalized coordinates of position, velocity and acceleration, respectively, σ_{\max} is the highest singular value of W and σ_{\min} the lowest singular value of W .

4.1.2. Trajectory parameterization

The prismatic joints motion does not participate in the rotary joint dynamics. Thus, without loss of generality, trajectory optimization is only applied to rotary joints, since they are in conflict with the parameter estimation of least squares. The reverse is true for the prismatic link with linear characteristics. The excitation trajectories are planned using a polynomial of degree five (Benimeli, 2005) in the form of Eq. (39) and the elements to be optimized are their coefficients: $a_{0i}, a_{1i}, \dots, a_{5i}$:

$$q_i = a_{0i} + a_{1i}t + a_{2i}t^2 + a_{3i}t^3 + a_{4i}t^4 + a_{5i}t^5 \quad (39)$$

Reference sinusoidal trajectories are applied to the first and fifth link (prismatic joints). All the reference and output trajectories of all the joints of the redundant manipulator robot are

shown in Figure 3. An electronic noise is observed due to the sensing (measuring) of the output trajectories.

4.2. Adaline neural networks

Applying the linear model formulation to the parameters $\tau = \Phi^* \cdot \lambda^*$ (see Annex B), we have that:

$$\begin{aligned} F_1 &= \Phi_{11}^* \lambda_1^* + \Phi_{18}^* \lambda_8^* \\ \tau_2 &= \Phi_{22}^* \lambda_2^* + \Phi_{23}^* \lambda_3^* + \Phi_{24}^* \lambda_4^* + \Phi_{25}^* \lambda_5^* + \Phi_{26}^* \lambda_6^* + \Phi_{27}^* \lambda_7^* \\ \tau_3 &= \Phi_{33}^* \lambda_3^* + \Phi_{34}^* \lambda_4^* + \Phi_{35}^* \lambda_5^* + \Phi_{36}^* \lambda_6^* + \Phi_{37}^* \lambda_7^* \\ \tau_4 &= \Phi_{45}^* \lambda_5^* + \Phi_{46}^* \lambda_6^* + \Phi_{47}^* \lambda_7^* \\ F_5 &= \Phi_{58}^* \lambda_8^* \end{aligned} \quad (40)$$

Therefore, when using an Adaline network, the parameters to be identified coincide with the network weights. Below, one-layer neural networks are designed with their inputs corresponding to the elements of the regressor and their outputs to the elements of torque/forces. Figure 4 shows the scheme of the networks used for the identification of the parameters of the redundant manipulator robot.

4.3. Hopfield neural networks

For satisfactorily identifying the parameters of the redundant manipulator robot, all intervals of $t \in [0, +\infty]$ should satisfy the condition given by Eq. (20). Figure 5 shows the response

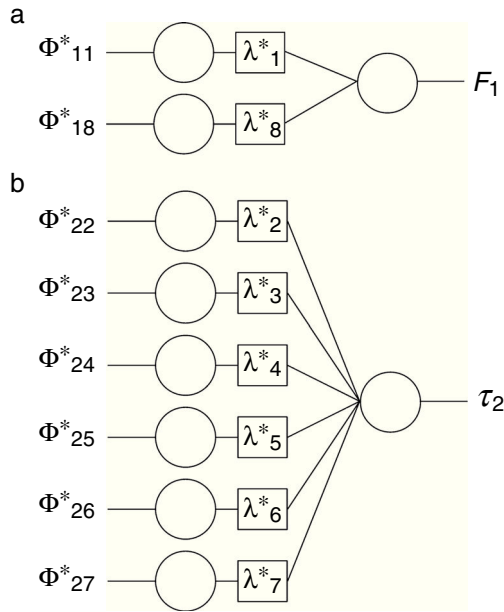


Fig. 4. Adaline neural network for identification. (a) Network 1. (b) Network 2.

produced by the HNN. The last estimated valued is used for the comparison.

4.4. Extended Kalman filter

In this method, use is made of the dynamic model of the redundant manipulator robot, i.e. the joint coordinates that mathematically depended on the force/torque applied to them are used. The direct dynamic model is symbolically calculated from the Lagrange–Euler formulation and the grouped parameters described in Table 3, and then is expressed in the state variables presented in Eq. (41). The parameters to be identified are included in an extended state of the algorithm with a time derivative of null values, as shown in Eq. (42):

$$x = \begin{bmatrix} d_1 & \dot{d}_1 & \theta_2 & \dot{\theta}_2 & \theta_3 & \dot{\theta}_3 & \theta_4 & \dot{\theta}_4 & d_5 & \dot{d}_5 & \lambda_1^* & \lambda_2^* & \lambda_3^* & \lambda_4^* & \lambda_5^* & \lambda_6^* & \lambda_7^* & \lambda_8^* \end{bmatrix}^T \quad (41)$$

$$\dot{x} = \begin{bmatrix} \dot{d}_1 & \ddot{d}_1 & \dot{\theta}_2 & \ddot{\theta}_2 & \dot{\theta}_3 & \ddot{\theta}_3 & \dot{\theta}_4 & \ddot{\theta}_4 & \dot{d}_5 & \ddot{d}_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad (42)$$

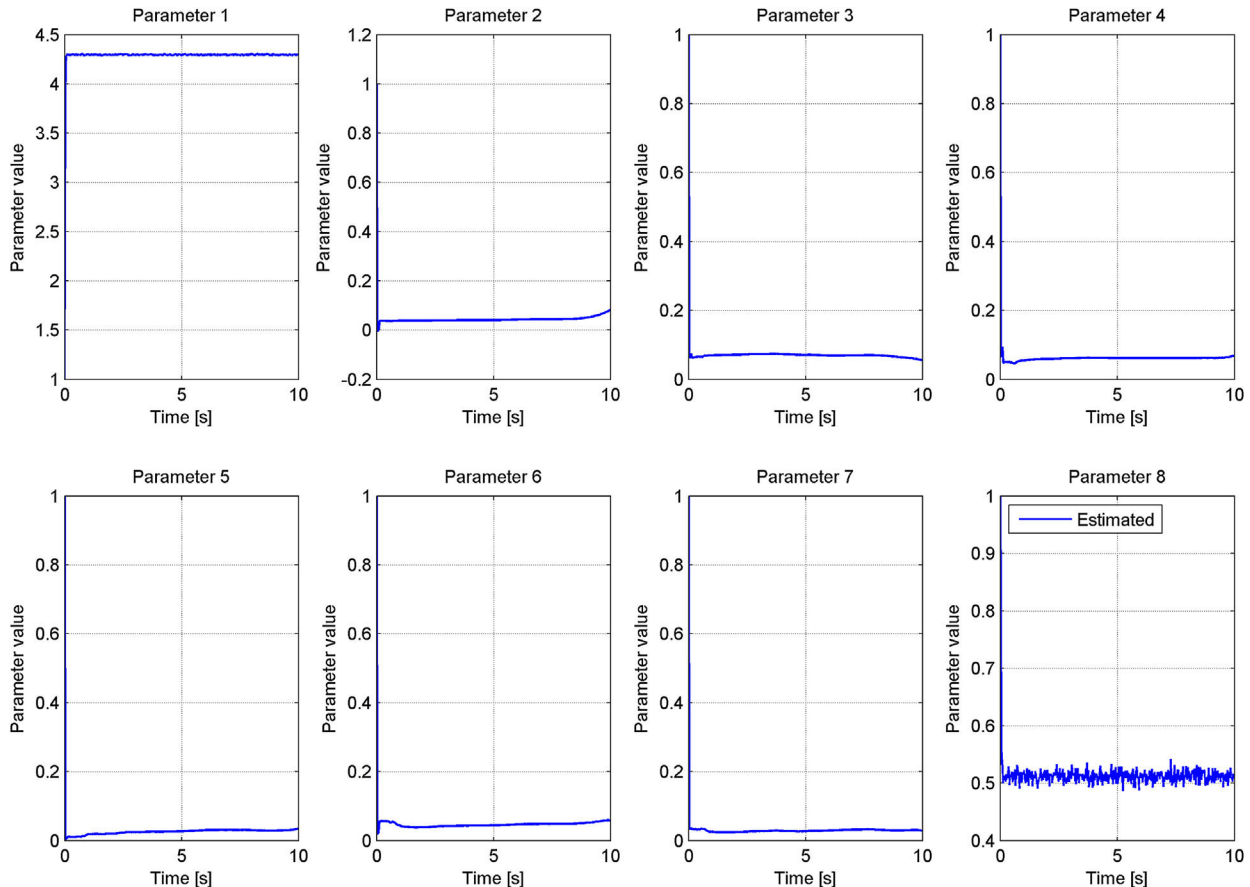


Fig. 5. Online estimation of Hopfield neural networks.

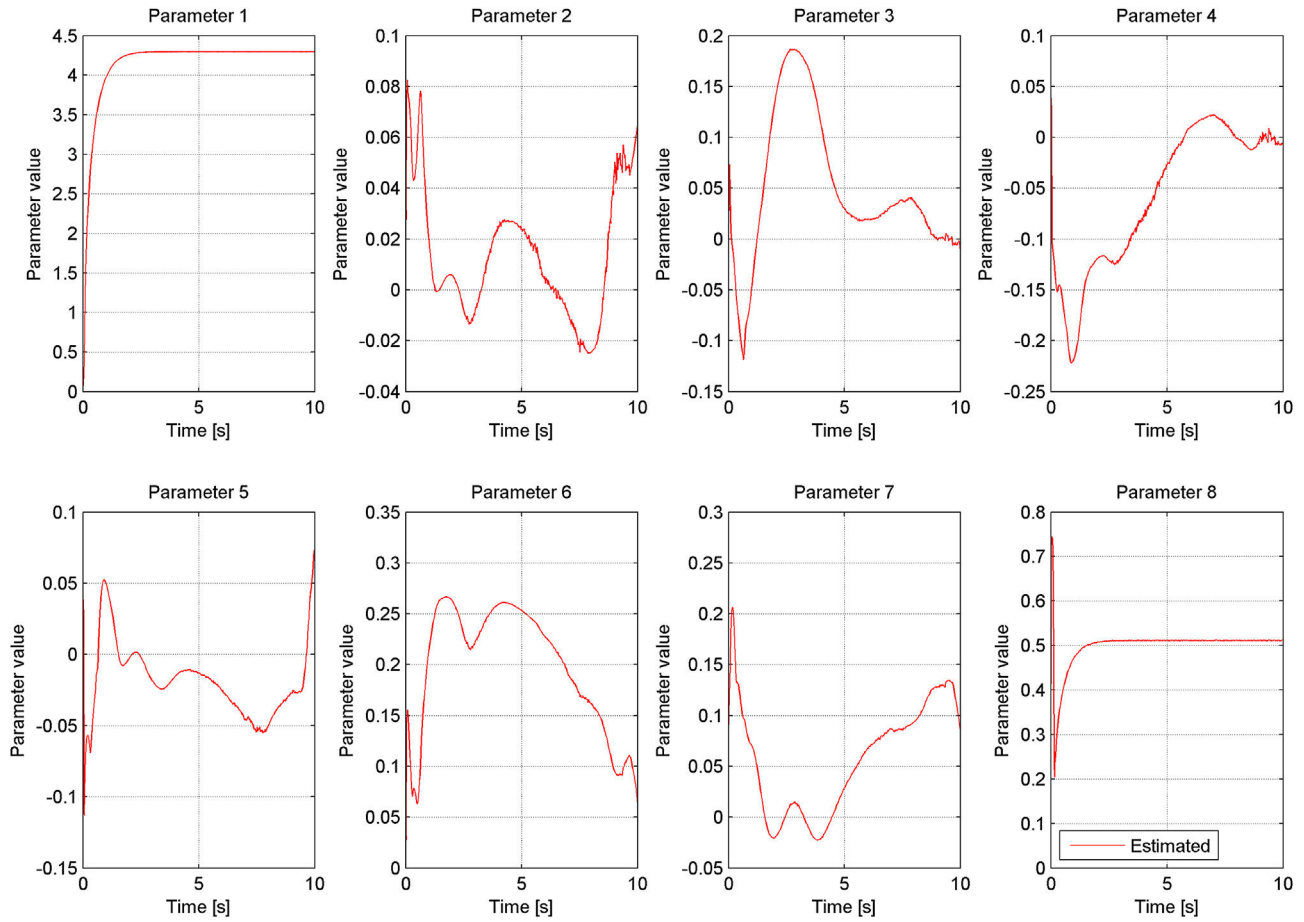


Fig. 6. Evolution of the parameters estimated by extended Kalman filter.

Euler approximation is used for the equations whose time derivative was calculated, as seen in Eq. (43):

$$\dot{x} \approx \frac{x^{k+1} - x^k}{\Delta t} \quad (43)$$

Therefore, from the equation above, $x^{k+1} = x^k + \Delta t \cdot \dot{x}^k$ is obtained, whose sampling time is $\Delta t = 0.02$. In turn, the initial state vectors used are initial error covariance matrix, process noise covariance matrix and observation covariance matrix, as presented in Eq. (44):

$$x^0 = 0, 1 \cdot \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}^T$$

$$P^0 \in \mathbb{R}^{18 \times 18}, \quad P_{ij} = 1$$

$$Q = \text{diag} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (44)$$

$$R = 10 \cdot \text{diag} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

The reference trajectories used in this identification process are the same applied to the least square methods, i.e. those presented in Figure 3. The evolution of the parameter estimation

Table 4
Estimated parameters.

Parameter	Least squares	Adaline	Hopfield	Kalman
λ_1^*	4.2969	4.2027	4.2984	4.2966
λ_2^*	0.1094	0.1450	0.0822	0.0647
λ_3^*	0.0677	0.0500	0.0551	-0.0114
λ_4^*	0.0671	0.0652	0.0692	-0.0080
λ_5^*	0.0271	0.0259	0.0354	0.0742
λ_6^*	0.0260	0.0181	0.0578	0.0634
λ_7^*	0.0251	0.0459	0.0287	0.0859
λ_8^*	0.5114	0.6056	0.5136	0.5119

curves is shown in Figure 6. In this case, the last value of the estimation is considered for error comparison.

5. Results

Table 4 shows the base parameters identified for each of the studied methods.

5.1. Trajectory validation

To validate the identified parameters, they are used in performance simulations on the redundant manipulator robot. In these simulations, validation trajectories different from those used in

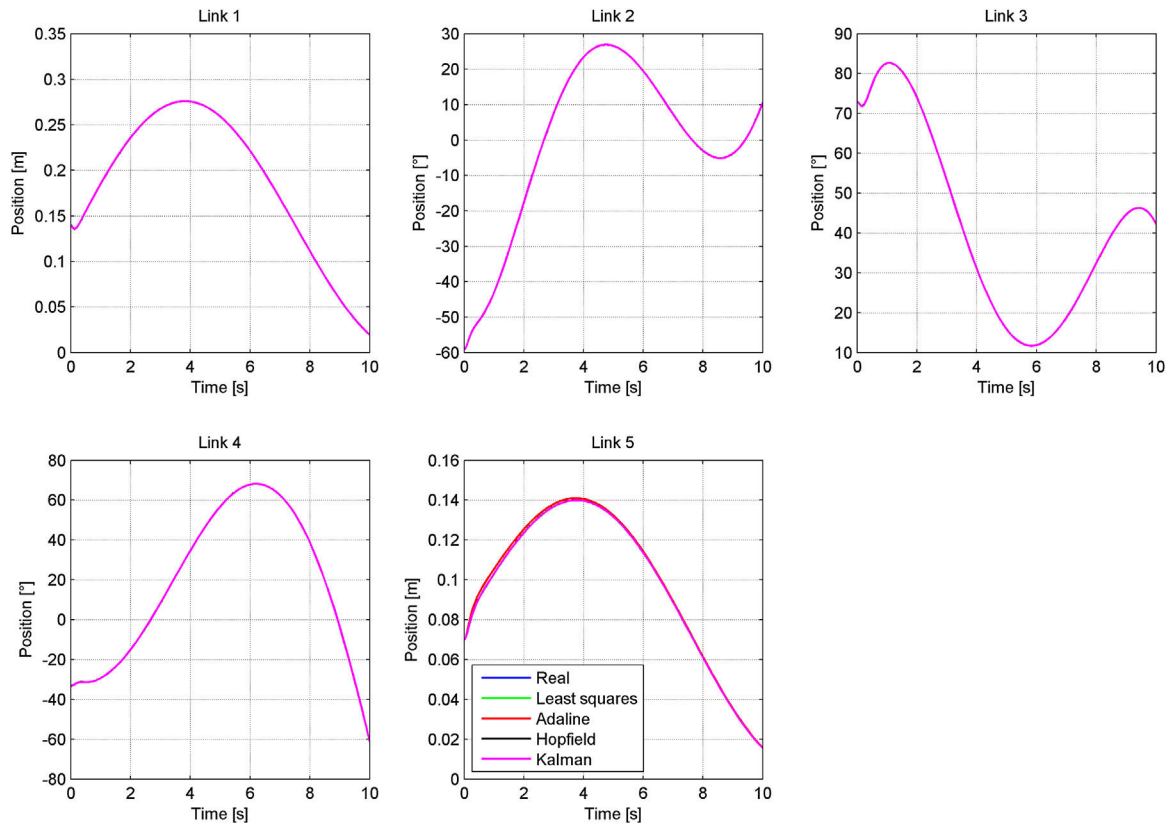


Fig. 7. Validation trajectories.

the estimation are imposed to the robot. To relate the trajectories performed by the redundant manipulator robot with the estimated parameters of the same, the following error indexes are calculated: Residual Mean Square (RMS), Residual Standard Deviation (RSD) and Agreement Index (AI); which are mathematically represented in Eqs. (45)–(47), respectively. RMS and RSD correspond to normalized percentage values between 0 and 1; indexes that explain the dispersion and deviation of the series, respectively. In addition, the AI index indicates the trend of the two series to be compared; the values that adopts may fluctuate between 0 and 1, where: 1 corresponds to a perfect match and 0 to absence of agreement. The trajectories performed by the redundant manipulator robot are shown in Figure 7, and their error indexes in Table 5. It should be noted that the trajectories presented in Figure 7 are basically overlapped, and that the values of the parameters do not show high sensitivity to following imposed trajectories:

Table 5
Error indexes.

		Algorithm			
	Joint	Least squares	Adaline	Hopfield	Kalman
1	RMS	0.0000	0.0000	0.0000	0.0000
	RSD	0.0000	0.0000	0.0000	0.0000
	AI	1.0000	1.0000	1.0000	1.0000
2	RMS	0.0000	0.0000	0.0000	0.0001
	RSD	0.0000	0.0001	0.0001	0.0002
	AI	1.0000	1.0000	1.0000	1.0000
3	RMS	0.0000	0.0000	0.0000	0.0001
	RSD	0.0000	0.0000	0.0000	0.0001
	AI	1.0000	1.0000	1.0000	1.0000
4	RMS	0.0000	0.0000	0.0000	0.0001
	RSD	0.0000	0.0000	0.0000	0.0002
	AI	1.0000	1.0000	1.0000	1.0000
5	RMS	0.0000	0.0012	0.0000	0.0000
	RSD	0.0000	0.0112	0.0003	0.0001
	AI	1.0000	0.9998	1.0000	1.0000

$$\text{RMS} = \sqrt{\frac{\sum_{i=1}^n (o_i - p_i)^2}{n}} \quad (45)$$

$$\text{RSD} = \sqrt{\frac{\sum_{i=1}^n (o_i - p_i)^2}{\sum_{i=1}^n (o_i)^2}} \quad (46)$$

$$\text{AI} = 1 - \frac{\sum_{i=1}^n (o_i - p_i)^2}{\sum_{i=1}^n (|o'_i| - |p'_i|)^2} \quad \begin{matrix} o'_i = o_i - o_m \\ p'_i = p_i - o_m \end{matrix} \quad (47)$$

where o_i are the values observed, o_m the mean value of the observations, p_i are the predicted values and n is the total number of observations.

To determine the sensitivity of the identified model, due to the variations of the estimated parameters, the redundant manipulator robot is simulated, considering that its parameters varied in a restricted range of values. Also, its RMS error index is calculated between the trajectories described. Table 6 presents the range of values for the variations in the $\pm 50\%$ parameters. In Figure 8,

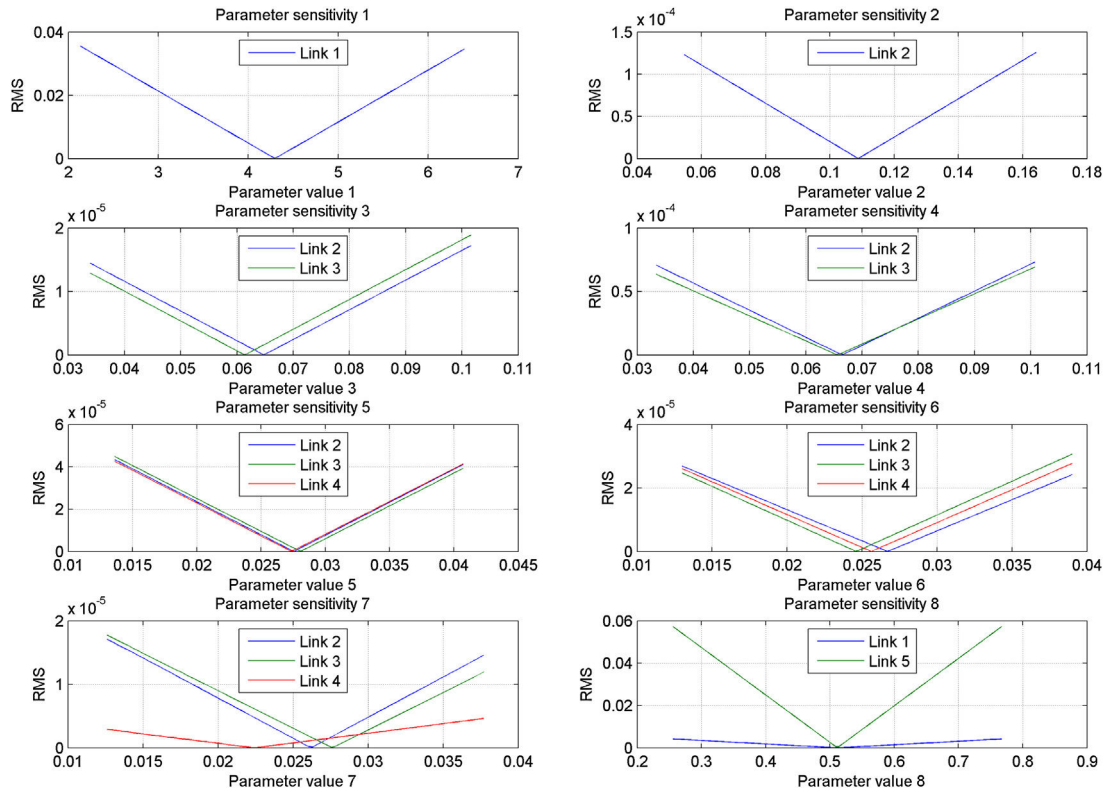


Fig. 8. Sensitivity of the model of the redundant manipulator robot.

Table 6
Range of estimated parameters.

Parameter	Minimum value	Maximum value
λ_1^*	2.1335	6.4004
λ_2^*	0.0547	0.1641
λ_3^*	0.0339	0.1016
λ_4^*	0.0334	0.1007
λ_5^*	0.0136	0.0407
λ_6^*	0.0130	0.0390
λ_7^*	0.0126	0.0377
λ_8^*	0.2557	0.7671

the evolution of the RMS index of the performed trajectories, with respect to each estimated parameter, is illustrated.

6. Conclusions

In this paper, we presented the identification of the parameters of an industrial redundant robot with 5 DOF. By means of the Lagrange–Euler formulation, the linear model with unknown parameters was obtained. To identify these parameters, four methods were used, assessed and compared, namely least squares, Adaline neural networks, Hopfield neural networks, and extended Kalman filter.

The identified parameters were validated by computationally simulating the performance of the redundant manipulator robot. To this end, reference trajectories different from those used in the estimation were imposed to the robot.

To relate the trajectories described by the redundant manipulator robot with its estimated parameters, the following error indexes were calculated: Residual Mean Square, Residual Standard Deviation and Agreement Index.

The sensitivity of the identified model to parameter variation is low.

The parameters of the redundant manipulator robot were successfully identified and, therefore, the robot mathematical model was known. Thanks to this achievement, it is now possible to design, simulate, compare, validate and implement diverse strategies for controlling this industrial redundant robot with 5 DOF.

Conflict of interest

The authors have no conflicts of interest to declare.

Acknowledgments

This work was supported by Proyectos Basales and the Vicerrectoría de Investigación, Desarrollo e Innovación of the Universidad de Santiago de Chile, Chile.

Annex A. Dynamic model

The dynamic model of the manipulator robot was developed in (Urrea & Kern, 2016). Below are presented the components of

the inertia matrix, the centrifugal force and Coriolis force vector and the gravity vector, respectively.

$$M_{11} = m_1 + m_2 + m_3 + m_4 + m_5 \quad (\text{A.1})$$

$$\begin{aligned} M_{12} = M_{21} = M_{13} = M_{31} = M_{14} = M_{41} = M_{25} \\ = M_{52} = M_{35} = M_{53} = M_{45} = M_{54} = 0 \end{aligned} \quad (\text{A.2})$$

$$M_{15} = M_{51} = -m_5 \quad (\text{A.3})$$

$$\begin{aligned} M_{22} = l_{c2}^2 m_2 + (l_2^2 + l_{c3}^2 + 2l_2 l_{c3} c_3) m_3 \\ + (l_2^2 + l_3^2 + l_{c4}^2 + 2l_2 l_3 c_3) m_4 + 2(l_3 l_{c4} c_4 + l_2 l_{c4} c_{34}) m_4 \\ + (l_2^2 + l_3^2 + l_4^2 + 2l_2 l_3 c_3) m_5 + 2(l_3 l_4 c_4 + l_2 l_4 c_{34}) m_5 \\ + I_{2zz} + I_{3zz} + I_{4zz} \end{aligned} \quad (\text{A.4})$$

$$\begin{aligned} M_{23} = M_{32} = (l_{c3}^2 + l_2 l_{c3} c_3) m_3 \\ + (l_3^2 + l_{c4}^2 + l_2 l_3 c_3 + 2l_3 l_{c4} c_4) m_4 + (l_2 l_{c4} c_{34}) m_4 \\ + (l_2 l_4 \cos_{34}) m_5 + (l_3^2 + l_4^2 + l_2 l_3 c_3 + 2l_3 l_4 c_4) m_5 \\ + I_{3zz} + I_{4zz} \end{aligned} \quad (\text{A.5})$$

$$\begin{aligned} M_{24} = M_{42} = (l_{c4}^2 + l_3 l_{c4} c_4 + l_2 l_{c4} c_{34}) m_4 \\ + (l_4^2 + l_3 l_4 c_4 + l_2 l_4 c_{34}) m_5 + I_{4zz} \end{aligned} \quad (\text{A.6})$$

$$\begin{aligned} M_{33} = l_{c3}^2 m_3 + (l_3^2 + l_{c4}^2 + 2l_3 l_{c4} c_4) m_4 \\ + (l_3^2 + l_4^2 + 2l_3 l_4 c_4) m_5 + I_{3zz} + I_{4zz} \end{aligned} \quad (\text{A.7})$$

$$M_{34} = M_{43} = (l_{c4}^2 + l_3 l_{c4} c_4) m_4 + (l_4^2 + l_3 l_4 c_4) m_5 + I_{4zz} \quad (\text{A.8})$$

$$M_{44} = l_{c4}^2 m_4 + l_4^2 m_5 + I_{4zz} \quad (\text{A.9})$$

$$M_{55} = m_5 \quad (\text{A.10})$$

$$C_{11} = C_{15} = 0 \quad (\text{A.11})$$

$$\begin{aligned} C_{12} = - \left(l_2 s_3 \cdot \dot{\theta}_3^2 + 2l_2 s_3 \cdot \dot{\theta}_2 \dot{\theta}_3 \right) (l_{c3} m_3 + l_3 m_4 + l_3 m_5) \\ + \left(l_2 s_{34} \cdot \dot{\theta}_3^2 - 2(l_3 s_4 + l_2 s_{34}) \dot{\theta}_3 \dot{\theta}_4 \right) (l_{c4} m_4 + l_4 m_5) \\ + \left(l_2 s_{34} \cdot \dot{\theta}_2 \dot{\theta}_3 - \left(\dot{\theta}_4^2 + 2\dot{\theta}_2 \dot{\theta}_4 \right) (l_2 s_{34} + l_3 s_4) \right) \\ \times (l_{c4} m_4 + l_4 m_5) \end{aligned} \quad (\text{A.12})$$

$$\begin{aligned} C_{13} = (l_{c3} m_3 + l_3 m_4 + l_3 m_5) l_2 s_3 \cdot \dot{\theta}_2^2 \\ + \left(l_2 s_{34} \cdot \dot{\theta}_2^2 - (2(\dot{\theta}_2 + \dot{\theta}_3) + \dot{\theta}_4) l_3 s_4 \dot{\theta}_4 \right) (l_{c4} m_4 + l_4 m_5) \end{aligned} \quad (\text{A.13})$$

$$\begin{aligned} C_{14} = (l_{c4} m_4 + l_4 m_5) (l_3 s_4 + l_2 s_{34}) \dot{\theta}_2^2 \\ + \left(l_3 s_4 \cdot \dot{\theta}_3^2 + 2l_3 s_4 \cdot \dot{\theta}_2 \dot{\theta}_3 \right) (l_{c4} m_4 + l_4 m_5) \end{aligned} \quad (\text{A.14})$$

$$\mathbf{G} = [(m_1 + m_2 + m_3 + m_4 + m_5) g_z \quad 0 \quad 0 \quad 0 \quad -m_5 g_z]^T \quad (\text{A.15})$$

where $s_3 = \sin \theta_3$, $s_4 = \sin \theta_4$, $c_3 = \cos \theta_3$, $c_4 = \cos \theta_4$, $s_{34} = \sin(\theta_3 + \theta_4)$ and $c_{34} = \cos(\theta_3 + \theta_4)$; m_1 , m_2 , m_3 , m_4 , m_5 and l_1 , l_2 , l_3 , l_4 , l_5 represent the masses and length of the first, second, third, fourth and fifth link, respectively; l_{c2} , l_{c3} and l_{c4} express the lengths from the origin to the mass center of the second, third and fourth link, accordingly. Finally, I_{2zz} , I_{3zz} and I_{4zz} represent the moments of inertia of the second, third and fourth links, referred to the z axis of each joint, respectively.

Annex B. Linear model with regrouped parameters

Below are shown the elements of the regressor $\Phi^* \in \mathbb{R}^{5 \times 8}$, obtained from the linear formulation with grouped parameters $\tau = \Phi^* \cdot \lambda^*$:

$$\tau = (F_1 \quad \tau_2 \quad \tau_3 \quad \tau_4 \quad F_5)^T \quad (\text{B.1})$$

$$\Phi^* = \begin{pmatrix} \phi_{11}^* & 0 & 0 & 0 & \phi_{15}^* \\ 0 & \phi_{22}^* & \phi_{23}^* & \phi_{24}^* & 0 \\ 0 & 0 & \phi_{33}^* & \phi_{34}^* & 0 \\ 0 & 0 & 0 & \phi_{44}^* & 0 \\ 0 & 0 & 0 & 0 & \phi_{55}^* \end{pmatrix} \quad (\text{B.2})$$

$$\phi_{11}^* = \ddot{d}_1 + g \quad (\text{B.3})$$

$$\phi_{15}^* = \ddot{d}_1 - \ddot{d}_5 + g \quad (\text{B.4})$$

$$\phi_{22}^* = \ddot{\theta}_2 \quad (\text{B.5})$$

$$\phi_{23}^* = \begin{pmatrix} \ddot{\theta}_2 + \ddot{\theta}_3 \\ (2\ddot{\theta}_2 + \ddot{\theta}_3) c_3 - (2\dot{\theta}_2 \dot{\theta}_3 + \dot{\theta}_3^2) s_3 \end{pmatrix}^T \quad (\text{B.6})$$

$$\phi_{24}^* = \begin{pmatrix} \ddot{\theta}_2 + \ddot{\theta}_3 + \ddot{\theta}_4 \\ (2\ddot{\theta}_2 + \ddot{\theta}_3 + \ddot{\theta}_4) c_{34} - (2(\dot{\theta}_2\dot{\theta}_4 + \dot{\theta}_3\dot{\theta}_4) + 2\dot{\theta}_2\dot{\theta}_3 + \dot{\theta}_3^2 + \dot{\theta}_4^2) s_{34} \\ (2\ddot{\theta}_2 + 2\ddot{\theta}_3 + \ddot{\theta}_4) c_4 - (2(\dot{\theta}_2\dot{\theta}_4 + \dot{\theta}_3\dot{\theta}_4) + \dot{\theta}_4^2) s_4 \end{pmatrix}^T \quad (\text{B.7})$$

$$\phi_{33}^* = \begin{pmatrix} \ddot{\theta}_2 + \ddot{\theta}_3 \\ \ddot{\theta}_2 c_3 + \dot{\theta}_2^2 s_3 \end{pmatrix}^T \quad (\text{B.8})$$

$$\phi_{34}^* = \begin{pmatrix} \ddot{\theta}_2 + \ddot{\theta}_3 + \ddot{\theta}_4 \\ \ddot{\theta}_2 c_{34} + \dot{\theta}_2^2 s_{34} \\ (2\ddot{\theta}_2 + 2\ddot{\theta}_3 + \ddot{\theta}_4) c_4 - (2(\dot{\theta}_4 + \dot{\theta}_3\dot{\theta}_4) + \dot{\theta}_4^2) s_4 \end{pmatrix}^T \quad (\text{B.9})$$

$$\phi_{44}^* = \begin{pmatrix} \ddot{\theta}_2 + \ddot{\theta}_3 + \ddot{\theta}_4 \\ \ddot{\theta}_2 c_{34} + \dot{\theta}_2^2 s_{34} \\ (\ddot{\theta}_2 + \ddot{\theta}_3) c_4 + (\dot{\theta}_2^2 + \dot{\theta}_3^2 + 2\dot{\theta}_2\dot{\theta}_3) s_4 \end{pmatrix}^T \quad (\text{B.10})$$

$$\phi_{55}^* = -\ddot{d}_1 + \ddot{d}_5 - g \quad (\text{B.11})$$

References

- Alonso, H., Mendonça, T., & Rocha, P. (2009). Hopfield neural networks for on-line parameter estimation. *Neural Networks*, 22(4), 450–462. <http://dx.doi.org/10.1016/j.neunet.2009.01.015>
- Anríquez, M. (2011). *Desarrollo de una nueva metodología de identificación de parámetros para sistemas no lineales: aplicación al modelo de Lorenz*. Santiago, Chile: Universidad de Santiago de Chile.
- Benimeli, F. (2005). *Estimación de parámetros dinámicos en robots manipuladores*. Valencia: Universidad Politécnica de Valencia.
- Bona, B., & Curatella, A. (2005). Identification of industrial robot parameters for advanced model-based controllers design. In *Proceedings of the 2005 IEEE international conference on robotics and automation* (pp. 1681–1686). IEEE. <http://dx.doi.org/10.1109/ROBOT.2005.1570355>
- Choi, J. S., Yoon, J. H., Park, J. H., & Kim, P. J. (2011). A numerical algorithm to identify independent grouped parameters of robot manipulator for control. In *2011 IEEE/ASME international conference on advanced intelligent mechatronics (AIM)* (pp. 373–378). IEEE. <http://dx.doi.org/10.1109/AIM.2011.6027068>
- Ha, I. J., Ko, M. S., & Kwon, S. K. (1989). An efficient estimation algorithm for the model parameters of robotic manipulators. *IEEE Transactions on Robotics and Automation*, 5(3), 386–394. <http://dx.doi.org/10.1109/70.34777>
- Isermann, R., & Münchhof, M. (2011). Neural networks and lookup tables for identification. *Identification of Dynamic Systems*, 501–537. <http://dx.doi.org/10.1007/978-3-540-78879-9-20>
- Kessman, K. J. (2011). *System identification: An introduction* (1st ed.). London: Springer-Verlag. <http://dx.doi.org/10.1007/978-0-85729-522-4>
- Ljung, L. (1987). *System identification: Theory for the user*. Prentice Hall.
- Mayeda, H., Yoshida, K., & Ohashi, K. (1989). Base parameters of dynamic models for manipulators with rotational and translational joints. In *International conference on robotics and automation* (pp. 1523–1528). IEEE Comput. Soc. Press. <http://dx.doi.org/10.1109/ROBOT.1989.100195>
- Mayeda, H., Yoshida, K., & Osuka, K. (1990). Base parameters of manipulator dynamic models. *IEEE Transactions on Robotics and Automation*, 6(3), 312–321. <http://dx.doi.org/10.1109/70.56663>
- Radkhah, K., Kulic, D., & Croft, E. (2007). *Dynamic parameters identification for the CRS A460 robot*. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 3842–3847).
- Swevers, J., Ganseman, C., De Schutter, J., & Van Brussel, H. (1996). Experimental robot identification using optimised periodic trajectories. *Mechanical Systems and Signal Processing*, 10(5), 561–577. <http://dx.doi.org/10.1006/mssp.1996.0039>
- Urrea, C., & Kern, J. (2016). Trajectory tracking control of a real redundant manipulator of the SCARA type. *Journal of Electrical Engineering & Technology*, 11(1), 215–226. <http://dx.doi.org/10.5370/JEET.2016.11.1.215>
- Welch, G., & Bishop, G. (2006). *An introduction to the Kalman filter (TR95-041)*. Chapel Hill: University of North Carolina at Chapel Hill, Department of Computer Science.
- Wu, J., Wang, J., & You, Z. (2010). An overview of dynamic parameter identification of robots. *Robotics and Computer-Integrated Manufacturing*, 26(5), 414–419. <http://dx.doi.org/10.1016/j.rcim.2010.03.013>