Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot

Article *in* Computers & Electrical Engineering · September 2016

DOI: 10.1016/j.compeleceng.2016.09.004

Claudio Urrea
University of Santiago, Chile

Computers and Electrical Engineering 000 (2016) 1–16

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot[☆]

Claudio Urrea[*], José Pascal

Grupo de Automática, Department of Electrical Engineering, Universidad de Santiago de Chile, Avenida Ecuador 3519, Estación Central, Santiago, Chile

ARTICLE INFO

Article history:
Received 9 April 2016
Revised 2 September 2016
Accepted 5 September 2016
Available online xxx

Keywords:
System identification
Dynamical systems
Mathematical model
Robotics

ABSTRACT

This study discusses the design and assessment of different parameter identification methods applied to robot systems, such as least squares, extended Kalman filter, Adaptive Linear Neuron (Adaline) neural networks, Hopfield recurrent neural networks and genetic algorithms. First, the characteristics of the methods above mentioned are described. Second, using the software MatLab/Simulink, a simulation of a Selective Compliant Assembly Robot Arm (SCARA) robot with 3 Degrees of Freedom (DOF) is carried out by applying these parameter identification methods, thereby obtaining the performance indicators of the algorithms that allow for parameter identification. Therefore, this study enables the adequate selection of identification methods to obtain parameters that characterize the dynamics of industrial robots, particularly of the SCARA type. Hence, having the values of the base parameters of a robot contributes to the design of new control methods, since the robot characteristic dynamic model is known.

1. Introduction

System modeling is a fundamental part of process control. However, difficulties such as uncertain or variable parameters [1] occasionally arise when attempting to determine a model. In such cases, parameter identification algorithms are required, and there is a variety of methods that may be implemented [2–4].

Certain parameters are needed to determine a robot model. Some of them may be obtained by means of physical experiments, but, to that end, disassembling each link of the manipulator robot, and applying measures according to each parameter requirements would be necessary. In addition, special measuring instruments would be needed to obtain accurate results.

Currently, numerous industrial applications —such as welding, cutting, painting, polishing, element handling, material load or unload, packing, palletizing, among others— have included robotized systems, which constitute automation systems. These industrial applications require robots that perform tasks faster and more accurately, which implies more robust controllers and, in turn, more precise mathematical models with better estimated parameters. Therefore, parameters used in the dynamic model of an industrial robot are important to control algorithms based on models, the validation of simulation results and the accuracy of the trajectory planning algorithms [5].

[☆] Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. A. Bhojan.

[*] Corresponding author. Fax: 56 2 2718 3350.
E-mail addresses: claudio.urrea@usach.cl (C. Urrea), jose.pascal@usach.cl (J. Pascal).

http://dx.doi.org/10.1016/j.compeleceng.2016.09.004
0045-7906/© 2016 Elsevier Ltd. All rights reserved.

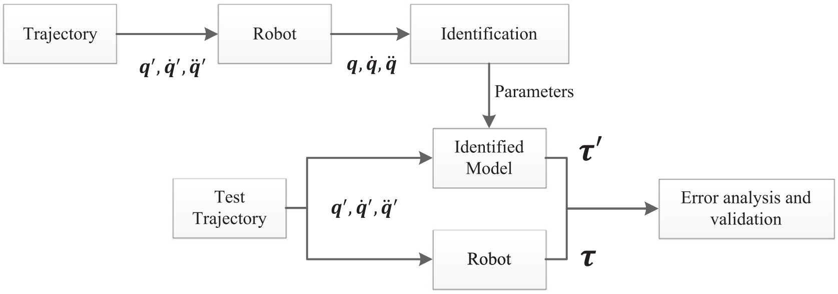

Fig. 1. Identification and validation scheme.

In this study, simulation is conducted using the MatLab/Simulink software to develop the dynamic model of a SCARA robot, and then to apply identification methods —such as least squares, extended Kalman filter, Adaline neural networks, Hopfield recurrent neural networks and genetic algorithms— to obtain the estimate's performance.

Parameter identification is widely used for creating models and developing controllers. There are several methods to estimate parameters in robotized systems. Furthermore, an adequate selection of such methods is closely related to their estimation accuracy and the complexity of their implementation in robotized systems.

One of the methods most used for identification is IDIM (Inverse Dynamic Identification Model), which consists in developing an inverse dynamic model in such a way that it is linear with respect to the dynamic parameters. The robot input and output signals are sampled while the robot is executing a trajectory that excites the dynamic system. Subsequently, parameters are estimated by using numerical optimization methods like least squares [5,6]. In fact, the least squares methodology has been applied and improved in a large number of robots and prototypes in order to identify inertial and friction parameters [7]. However, the least squares method requires the force/torque and joint position values, as well as the estimation of joint's velocity and acceleration through a band pass filter, which should be well adjusted.

An alternative method, more common in the automatic control community, is the use of a Kalman filter [8], which is based on the direct dynamic model. Gautier and Poignet presented an experimental comparison in a 2-DOF robot using weighted least squares and extended Kalman filter [5,9]. The comparison showed that parameter estimations are very similar for both methods, but the extended Kalman filter is very sensitive to initial conditions and its convergence velocity is slower.

Another method that has also been used is maximum-likelihood estimation [10]. Nevertheless, Olsen pointed out that this method should only be considered in cases in which both measurements, position and torque, are noisy [5]. The instrumental variables method has also been implemented and used under the inverse dynamic model [11].

Moreover, a new method for manipulator robots, which uses a parameter identification method and a dynamic compensator based on neural networks to improve the accuracy of estimation has been proposed [12].

Finally, the DIDIM (Direct Inverse Dynamic Identification Model) method solely requires the joint force/torque measurement, thus avoiding the calculation of speed and acceleration through the band pass filter of the position. In this closed-loop output error method, the usual position output is substituted by force/torque. It is based on a robot closed-loop simulation using a direct dynamic model, the same structure of control law, and the same reference trajectory, for both the real and the simulated robot [13].

First, these parameter identification methods are reviewed in Section 2. Second, Section 3 introduces the robot model and its parameters. Third, Section 4 shows simulations of these identification methods, whose results are presented in Section 5. Finally, conclusions are drawn in Section 6.

2. Parameters identification methods

In this section, the procedure of identification of the robot system is described, and the methods used, namely least squares, Adaline neural networks, Hopfield neural networks, Kalman filter and, genetic algorithms are addressed.

The identification of the robot system parameters starts by sampling the input and output signals while the robot is executing a trajectory that excites the dynamic system. However, not all the parameters have the same effect on the dynamic model and to estimate all the values of the dynamic parameters based on the joint motion or force/torque measurements is impossible, since the dynamic parameters of links are redundant to uniquely determine the dynamic model of the manipulator robot. Identifiable non-redundant dynamic parameters may be expressed as a minimum set of dynamic parameters whose values may determine a model. Such parameter sets are called base or minimum dynamic parameters, and are sufficient to describe the dynamic behavior of the mechanic system [14]. There are methodologies that allow for determining the parameters affecting the dynamic, as well as the set of minimum parameters to identify the model [7,14]. Once the parameters are identified, it is necessary to validate such results by applying them to the model. The objective of this validation is to verify the adequacy of the estimated robot model for future applications. Validation is carried out through a validation trajectory, and a subsequent comparison between measured and estimated torques, or the calculation of the error between trajectories. The validation scheme is shown in Fig. 1. [5].

Please cite this article as: C. Urrea, J. Pascal, Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.09.004

2.1. Least squares

Among the parameter estimation methods, least squares has an important part in the identification of manipulator robots [15], since it allows for 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 [5], because noise will limit the obtained parameters' accuracy 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 [16].

A robot dynamic model may be derived and expressed in Lagrange–Euler formulation as shown in (2):

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) \tag{2}$$

where τ corresponds to the generalized forces vector, $M(q)$ to the inertia matrix, $C(q,\dot{q})$ to the centrifugal force and Coriolis vector, $G(q)$ to the gravitational force vector, and q to the generalized coordinates vector.

The dynamic model may be written as a set of unknown parameters linear equations, as shown in (3):

$$y(\tau) = \Phi(q,\dot{q},\ddot{q})p \tag{3}$$

where $y(\tau)$ is the force/torque vector, p are the unknown dynamic parameters, and Φ is the observation matrix or regressor.

By applying the identification model to the model shown in (3), the linear equation system in p is obtained as:

$$y(\tau) = \Phi(q,\dot{q},\ddot{q})p + \rho \tag{4}$$

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

$$\hat{p} = min_p||\rho||^2 \tag{5}$$

which is the same as calculating:

$$\hat{p} = \left(\Phi^T\Phi\right)^{-1}\Phi^T y(\tau) \tag{6}$$

2.2. Adaline network

A neural network consists of a set of simple processing units, which communicate by sending signals to one another through weighted connections [17]. For a one-layer network with one output and one linear activation function, the output is given by (7).

$$y = \sum_{i=1}^{n} w_i x_i + \theta \tag{7}$$

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 of $y = d^p$ in the output when the set of values x_i^p, $i = 0,\ 1,\ \ldots,\ n$ is applied to the inputs. The problem lies in determining the coefficients (weights) w_i, $i = 0,\ 1,\ \ldots,\ n$, so the input-output response is correct for a large number of arbitrarily chosen signal sets. In this case, Widrow introduced the "delta rule" to adjust weights. Supposing that the network needs to be trained in such a way that the best adjustment for each set of data composed of the input values x^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 (8), i.e., the total error E is defined by:

$$E = \sum_p E^p = \frac{1}{2}\sum_p (d^p - y^p)^2 \tag{8}$$

where the p index extends over the set of input patterns and E^p represents the error over the p pattern. This procedure finds the value for all the weights that minimize the error function by means of the gradient descent method:

$$\Delta_p w_i = -\gamma \frac{E^p}{w_i} \tag{9}$$

where γ is a constant of proportionality. Therefore, $\Delta_p w_i = \gamma\delta^p x_i$, is obtained, 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 networks

Hopfield neural networks (HNNs) can be used for estimating parameters on-line. In Hopfield's formulation, the i neuron dynamics is described by the ordinary differential Eq. (10) [18].

$$\frac{du_i}{dt} = -\frac{1}{C_i}\left(\frac{1}{R_i}u_i(t) + \sum_j W_{ij} f_i\big(u_j(t)\big) + I_i\right) \tag{10}$$

Please cite this article as: C. Urrea, J. Pascal, Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.09.004

where u_i is the neuron input i, f_i is a strictly increasing, bounded, nonlinear continuous function, and C_i, R_i, W_{ij}, 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, respectively. 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. as shown in (3). Furthermore, it works under the following assumptions:

(1) $y,\ \Phi\ \in\ C^1$.
(2) $y,\ \Phi$ are bounded.
(3) $y(t),\ \Phi(t)$ are known in each t instant.
(4) $p\ \in\ ]-c,\ c[^n$ for some $c > 0$ known.

Given Abe's formulation, considering a network with N neurons, u_i denotes the total input of neuron i:

$$\frac{du_i}{dt}(t) = -\left(\sum_{j=1}^{N} W_{ij}(t)s_j(t) + I_i(t)\right) \tag{11}$$

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

$$s_i(t) = \alpha \cdot \tanh\left(\frac{u_i(t)}{\beta}\right) \tag{12}$$

where $\alpha,\ \beta > 0$, and, thus, $\forall i \in N,\ t \geq t_0\ s_i(t) \in\ ]-\alpha,\ \alpha[$. Using matrix notation, the network is as follows:

$$\frac{du}{dt}(t) = -(W(t)s(t) + I(t)) \tag{13}$$

$$s(t) = \alpha \cdot \tanh\left(\frac{u(t)}{\beta}\right) \tag{14}$$

where $p(t)$, $s(t)$, $I(t) \in \mathbb{R}_{Nx1}$ and $W \in \mathbb{R}_{NxN}$, under the state-space representation is:

$$\frac{ds}{dt}(t) = -\frac{1}{\alpha\beta} D_\alpha(s(t))(W(t)s(t) + I(t)) \tag{15}$$

where $D_\alpha(s(t)) = diag((\alpha^2 - s_i(t)^2)_{i\varepsilon N})$.
This is a nonlinear and non-autonomous dynamic system whose architecture is completely determined by the N number of neurons, with characteristics given by $\alpha,\ \beta,\ W\, and\, I$.

A HNN whose state over time, t, $s(t)$, is taking as the $\hat{p}(t)$ estimation of the p parameter; in other words, $\hat{p}(t) = s(t)$ is considered. Implicitly, there is a network with as many neurons as parameters to estimate. Therefore, taking $\alpha = c$, it is guaranteed by (12) that the trajectory generated for the network is in a feasible region of the estimation problem. Then, from the state representation (15), $W = \Phi(t)^T\Phi(t)$ y $I = -\Phi(t)^T y(t)$ is taken. In this way, the HNN to be used as an on-line estimation algorithm defines its dynamic by:

$$\frac{d\hat{p}}{dt}(t) = \frac{1}{c\beta} D_c\big(\hat{p}(t)\big)\Phi(t)^T\big(y(t) - \Phi(t)\hat{p}(t)\big) \tag{16}$$

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

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

$$I \subset [t_0, +\infty[,\ \bigcap_{t\in I} ker(\Phi(t)) = \{0\} \tag{17}$$

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 $x\ \in IR^n$ state of a process controlled in discrete time [8]. In this case, the process will be represented by the following non-linear equations:

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} \tag{18}$$

$$z_k = h(x_k) + v_k \tag{19}$$

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. Moreover, x_k is the state vector, z_k the system output or observation vector, w_{k-1} the white noise associated to the process, and v_k the white noise associated to the measurement. The Extended Kalman filter is executed as follows:

1. Prediction stage
   1.1 State prediction

$$\hat{x}_k^- = f(x_{k-1}, u_{k-1}) \tag{20}$$

   1.2 Prediction of error covariance matrix

$$P_k^- = F_{k-1} P_{k-1} F_{k-1}^T + Q \tag{21}$$

2. Correction or updating stage
   2.1 Kalman gain calculation

$$K_k = P_k^- H_k^T \cdot \left[H_k P_k^- H_k^T + R \right]^{-1} \tag{22}$$

   2.2 State update with z_k

$$\hat{x}_k = \hat{x}_k^- + K_k\left(z_k - h\left(\hat{x}_k^-\right)\right) \tag{23}$$

   2.3 Covariance matrix update

$$P_k = (I - K_k H_k) P_k^- \tag{24}$$

where F_{k-1} y H_k corresponds the Jacobian matrices described in (25).

$$F_{k-1} = \left.\frac{\partial f}{\partial x}\right|_{x_{k-1}, u_{k-1}} \qquad H_k = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_k^-} \tag{25}$$

where $\hat{x}_k$ is the estimate at k instant of the state. Kalman filter intends to obtain this estimate value so as to the mean squared error minimizes, defining the error as the difference between the state and estimate values as $e_k = x_k - \hat{x}_k$. Therefore, the objective will be to minimize$P_k = E[e_k \cdot e_k^T]$. P_k is known as the error covariance matrix. Q and R are the process and observation noise covariance matrices, respectively. In addition, they are diagonal and symmetric. On the other hand, $\hat{x}_k^- \in IR^n$ are the states estimated *a priori* at k instant, given the process behavior and $\hat{x}_k \in IR^n$, the *a posteriori* process estimated state at k instant, given z_k.

2.5. Genetic algorithms

Genetic algorithms are evolution optimization methods that resolve optimization problems in which is necessary to maximize or minimize an objective function [19]. Genetic algorithms work with an individuals' population. Each individual represents a feasible solution to a given problem and is assigned a value or score, related to the goodness of such solution. The greater is an individual's adaptation, the higher the probability that the individual is selected for reproduction and exchanges his/her genetic material with other individual selected in the same way. This exchange will generate new individuals, the offspring of the former ones, who will share some of their parents' characteristics. The lesser an individual's adaptation, the lower the probability that he/she is selected for reproduction and, thus, that his/her genetic material passes to subsequent generations.

Evolution develops from an initial individuals' population (generated randomly) and takes place in a series of iterations denominated generations. To use a genetic algorithm, it is necessary to define the following elements [20]:

– A chromosome able to represent a solution to the problem under study.
– An individual's evaluation function that allows the assessment of the suitability of he/she, based on the problem parameters.
– Genetic operators appropriate to cross (parents generate children) and modify (mutating genes in the chromosome) the current population individuals.

A simple algorithm flow chart is shown in Fig. 2. Individuals (possible solutions to the problem) can be represented as a set of parameters denominated genes. These genes group form a set of values, which are called chromosomes. Hence, the number of chromosomes determines the size of the population. Whether the population is insufficient, the genetic algorithm has scarce possibilities of reproduction, thus leading to a scarce and non-optimal search for solutions. Conversely, if the population is excessive, the genetic algorithm will be extremely slow.

3. Industrial manipulator robot model

The characteristics of the manipulator robot analyzed in this study are the following: a SCARA robot with 3 DOF, two rotary joints and one prismatic joint. Fig. 3 shows a representation of the manipulator robot with its coordinate systems assigned. Denavit–Hartenberg parameters for industrial robots are shown in Table 1.

Fig. 4 shows the parameters of the system under study. These parameters will be used to determine a dynamic model of the manipulator robot by applying the Lagrange–Euler formulation.

Please cite this article as: C. Urrea, J. Pascal, Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.09.004

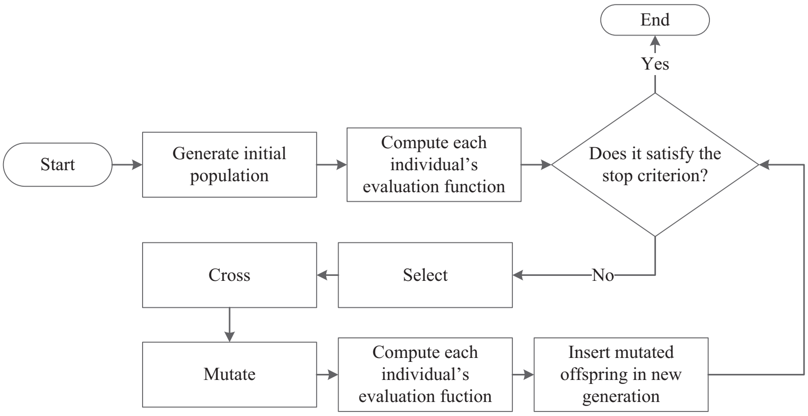
Start
Generate initial population
Compute each individual's evaluation function
Does it satisfy the stop criterion?
Yes
End
No
Select
Cross
Mutate
Compute each individual's evaluation fuction
Insert mutated offspring in new generation

Fig. 2. Simple genetic algorithm flow chart.

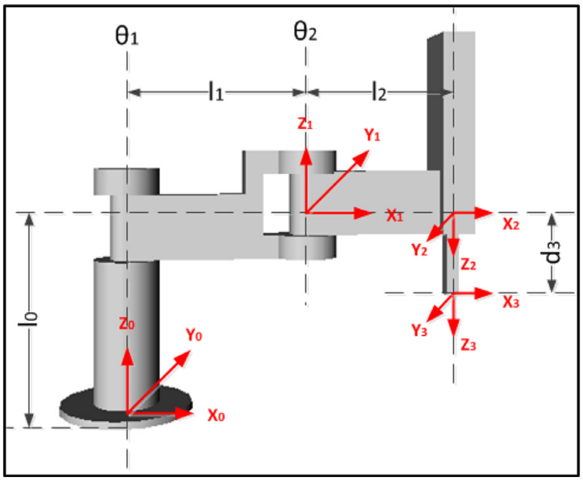

Fig. 3. Assignment of RPP manipulator coordinate systems.

Table 1
Denavit–Hartenberg parameters.

Joint i	θ_i	d_i	a_i	α_i
1	θ_1*	l_0	l_1	0
2	θ_2*	0	l_2	180°
3	0	d_3*	0	0

* Variables

where:

m_1: First link mass
m_2: Second link mass
m_3: Third link mass.
l_1: First link length.
l_2: Second link length.
l_3: Third link length.
lc_1: Length from the first link origin to its center of mass
lc_2: Length from the second link origin to its center of mass
lc_3: Length from the third link origin to its center of mass
I_1: Moment of inertia of first link
I_2: Moment of inertia of second link
I_3: Moment of inertia of third link

Please cite this article as: C. Urrea, J. Pascal, Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.09.004

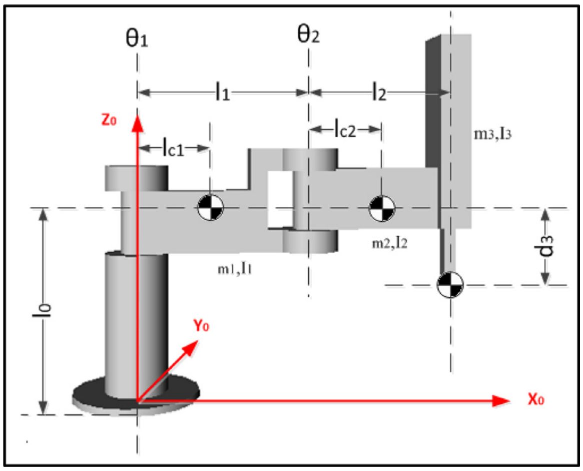

Fig. 4. RRP manipulator dynamics.

Table 2
Manipulator robot parameters.

Parameters	link 1	link 2	link 3
m_i $[Kg]$	12	6	2
l_i $[m]$	0.6	0.4	–
l_{ci} $[m]$	0.3	0.2	–
I_i $[Kgm^2]$	0.36	0.08	0.08

The values used in the dynamic parameters of each manipulator link (mass, length, center of mass and inertia) are described in Table 2 [21].

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:

$$L(q_i, \dot{q}_i) = T - U \tag{29}$$

From the Lagrangian (29), the dynamic system equations of motion are given by:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i \tag{30}$$

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. Finally, the general form of a manipulator robot dynamic model may be expressed as in (2). In this way, the following equations describing the manipulator dynamics are obtained:

$$\tau_1 = M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 - 2h\dot{\theta}_1\dot{\theta}_2 - h\dot{\theta}_2^2 \tag{31}$$

$$\tau_2 = M_{12}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 + h\dot{\theta}_1^2 \tag{32}$$

$$F_3 = M_{33}\ddot{d}_3 - G \tag{33}$$

where:

$$M_{11} = m_1 {l_{c1}}^2 + m_2\left({l_1}^2 + {l_{c2}}^2 + 2l_1 l_{c2}\cos(\theta_2)\right) + m_3\left({l_1}^2 + {l_2}^2 + 2l_1 l_2\cos(\theta_2)\right) + I_1 + I_2 + I_3 \tag{34}$$

$$M_{12} = m_2 l_{c2}(l_{c2} + l_1\cos(\theta_2)) + m_3\left(l_1 l_2\cos(\theta_2) + {l_2}^2\right) + I_2 + I_3 \tag{35}$$

$$h = (m_2 l_1 l_{c2} + m_3 l_1 l_2) sen(\theta_2) \tag{36}$$

$$M_{22} = m_2 {l_{c2}}^2 + m_3 {l_2}^2 + I_2 + I_3 \tag{37}$$

$$M_{33} = m_3 \tag{38}$$

$$G = m_3 g \tag{39}$$

4. Simulation of parameter identification methods

The SCARA robot simulation is conducted using the software MatLab/Simulink.

4.1. Least squares

The least squares procedure will consist in the following steps [7]:

- 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.
- Estimating link parameters using the least squares procedure.

Therefore, obtaining a model linear with respect to the parameters is necessary to apply the least squares procedure. Due to the reduced number of DOF of the robot, to obtain an empirical formulation is possible, taking into account that the regressor should be a full rank matrix.

Consider the dynamic equations of the manipulator (31)–(39) and the values of Table 2. Parameters are grouped as follows:

$$\begin{aligned} I_{ZZ1} &= m_1 {l_{c1}}^2 + I_1 + m_2 {l_1}^2 + m_2 {l_{c2}}^2 + I_2 + m_3 {l_1}^2 + m_3 {l_2}^2 + I_3 \\ I_{ZZ2} &= m_2 {l_{c2}}^2 + I_2 + m_3 {l_2}^2 + I_3 \\ m_r &= m_2 l_1 l_{c2} + m_3 l_1 l_2 \end{aligned} \tag{40}$$

where I_{ZZ1} corresponds to the 1, 2 and 3 link inertia on the Z_0 axis, I_{ZZ2} to the 2 and 3 link inertia on the Z_1axis, m_r to 1 and 2 the link center of mass, and m_3 to the 3 link mass.

The model linear with respect to the parameters (3) is expressed in matrices like (41), (42) and (43).

$$y(\tau) = \begin{bmatrix} \tau_1 & \tau_2 & F_3 \end{bmatrix}^T \tag{41}$$

$$\phi = \begin{vmatrix} \ddot{\theta}_1 & \ddot{\theta}_2 & \left(2\ddot{\theta}_1 + \ddot{\theta}_2\right)\cos(\theta_2) - \left(\dot{\theta}_2^2 + 2\dot{\theta}_1\dot{\theta}_2\right) sen(\theta_2) & 0 \\ 0 & \left(\ddot{\theta}_1 + \ddot{\theta}_1\right) & \left(\ddot{\theta}_1 \cos(\theta_2) + \ddot{\theta}_1^2 sen(\theta_2)\right) & 0 \\ 0 & 0 & 0 & \left(\ddot{d}_3 - g\right) \end{vmatrix} \tag{42}$$

$$p = \ [I_{ZZ1}\ I_{ZZ2}\ m_r\ m_3]^T \tag{43}$$

4.1.1. Excitation trajectories

When a system identification experiment is designed, it is necessary to consider a trajectory that excites sufficiently the manipulator robot parameters during the movement. Otherwise, some parameters will be impossible to identify, or will become too sensitive to noise [22].

Meanwhile, since the regressor Φ depends on joint coordinates and its time derivatives, when the manipulator robot executes a particular trajectory, $DOF \cdot n_{ptos}$ equations will be obtained, generating an over-determined system. Thus, considering the points traveled in the trajectory, the system is given by the expression (44)

$$\tau(\eta) = H \cdot p \tag{44}$$

where $\tau(\eta)$ is the $\in M_{DOF \cdot n_{ptos} x1}$ torque/force vector, H is the $\in M_{DOF \cdot n_{ptos} x n_p}$ system observation matrix, p is the $\in M_{n_p x1}$ base parameter vector, n_{ptos} is the number of points, n_p is the number of base parameters and M_{nxp} corresponds to a n by p matrix.

To identify the parameters, (45) should be calculated:

$$p = \left(H^T \cdot H\right)^{-1} H^T \cdot \tau \tag{45}$$

Parameters accuracy is closely related to the observation matrix W, so the purpose is to obtain a well-conditioned matrix to, thereby, enhance the estimation results [5].

The above described implies finding the denominated “excitation trajectories”, which correspond to an optimization problem that consists in minimizing a function in the (46) form, subjected to the constraints (47) given by the SCARA robot working space and by the actuator characteristics. The f function corresponds to the criterion $k(W) = \frac{\sigma_{max}}{\sigma_{min}}$, where σ_{max} is the highest singular value of W, and σ_{min} is the lowest singular value of W.

$$minimize\ \ f(q, \dot{q}, \ddot{q}) \tag{46}$$

$$\begin{aligned} q_{min} &\le q \le q_{max} \\ \dot{q}_{min} &\le \dot{q} \le \dot{q}_{max} \\ \ddot{q}_{min} &\le \ddot{q} \le \ddot{q}_{max} \end{aligned} \tag{47}$$

q, $\dot{q}$, $\ddot{q}$ are the generalized coordinates of position, velocity and acceleration for each joint, respectively.

Please cite this article as: C. Urrea, J. Pascal, Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.09.004

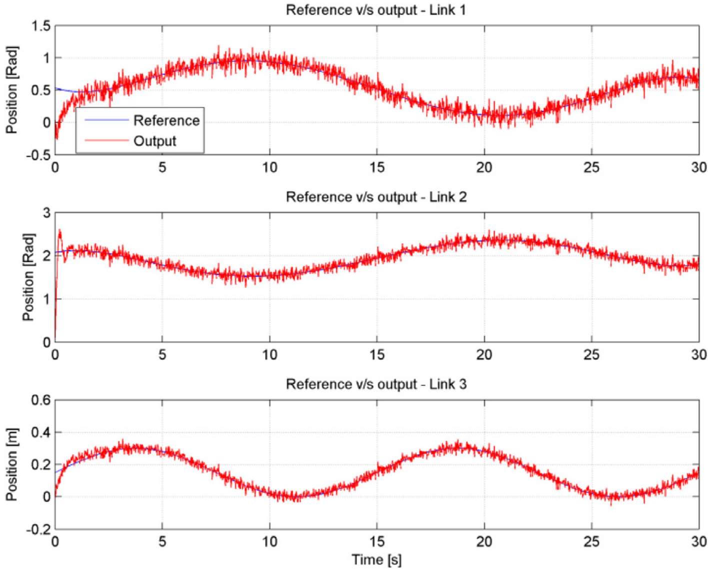

Fig. 5. 3-DOF SCARA robot reference v/s output.

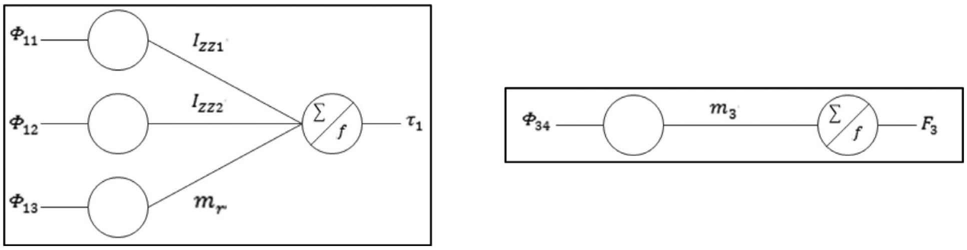

Fig. 6. Adaline neural network for identification.

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 trajectory is parameterized using a polynomial of degree five [23], therefore the position is as shown in (48) and the elements to be optimized are the coefficients of the polynomial $\{a_0, a_1, \ldots, a_5\}$.

$$q = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{48}$$

For the third link, a sinusoidal trajectory is applied as reference. In addition, the robot outputs, along with the torques, are considered affected by a Gaussian noise. The trajectories shown in Fig. 5 are used to identify the parameters in all the methods presented.

4.2. Adaline neural networks

Applying the linear model formulation to the parameters in (3), the following is obtained:

$$\begin{aligned} \tau_1 &= I_{ZZ1}\Phi_{11} + I_{ZZ2}\Phi_{12} + m_r\Phi_{13} \\ \tau_2 &= I_{ZZ2}\Phi_{22} + m_r\Phi_{23} \\ F_3 &= m_3\Phi_{34} \end{aligned} \tag{49}$$

Therefore, according to the Adaline network structure (7), the parameters to be identified should coincide with the network weights, and in this case, bias should be zero. In this way, one-layer neural networks should be built. Their inputs will be the elements corresponding to the regressor, while their outputs will be to torque/forces. The networks shown in Fig. 6 are used for parameter identification.

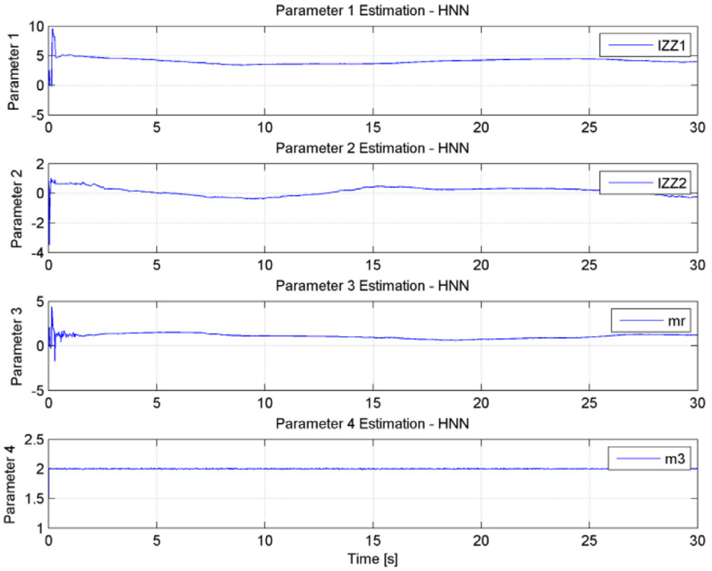

Fig. 7. Parameters estimated by Hopfield neural networks.

4.3. Hopfield networks

In this method, the modified formula proposed in (16), which uses a linear formula for the parameters to be identified in Eq. (3), is applied. Also, it is considered that $c=10$ and $\beta=0.01$. To achieve a satisfactory identification, conditions (17) must be met. Consequently, considering the (42) regressor, it is stated that:

$$\tilde{p} \in \ker(\phi(t)) \Leftrightarrow \begin{cases} \phi_{11}\tilde{p}_1 + \phi_{12}\tilde{p}_2 + \phi_{13}\tilde{p}_3 = 0 \\ \phi_{22}\tilde{p}_2 + \phi_{23}\tilde{p}_3 = 0 \\ \phi_{34}\tilde{p}_4 = 0 \end{cases} \tag{50}$$

Hence, condition (17) holds for all the $I \subset [0, +\infty[$ interval. In Fig. 7, the response produced by the HNN is shown. In this case, for parameter comparison, the estimated value is calculated by applying the mean value to the Hopfield networks response between 5 and 30 seconds of simulation.

4.4. Extended Kalman filter

The direct dynamic model is symbolically calculated from Lagrange–Euler formulation, considering the parameters group described in (40). Once this has been conducted, state variables are transformed, considering that to estimate parameters through this algorithm, an extended state including these parameters should be included.

Then $x=[\theta_1\ \dot{\theta}_1\ \theta_2\ \dot{\theta}_2\ d_3\ \dot{d}_3\ I_{zz1}\ I_{zz2}\ m_r\ m_3]^T$ and $\dot{x}=[\dot{\theta}_1\ \ddot{\theta}_1\ \dot{\theta}_2\ \ddot{\theta}_2\ \dot{d}_3\ \ddot{d}_3\ 0\ 0\ 0\ 0]^T$.Thereby, the formula in state variables proposed in (18) is obtained. To calculate the algorithm, i.e., Eqs. (20)–(24), the following is considered: Euler approximation for derivatives, as shown in (51).

$$\dot{x} \approx \frac{x^{k+1}-x^k}{\Delta t} \tag{51}$$

Using such approximation and substituting in the state equations, formulation (52) is obtained and, then, implemented in the Kalman filter with $\Delta t=0.01$.

$$x^{k+1} = x^k + \Delta t \dot{x}^k \tag{52}$$

Initial error covariance matrix, process noise covariance matrix and observation covariance matrix, respectively, are considered the initial value, which is given by:

$$x^0 = (1\ 1\ 1\ 1\ 1\ 1\ 5\ 0,5\ 1\ 1,5)^T,$$

$$P^0 = Q = 10 \cdot diag(1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$$
$$R = diag\left(\left(\tfrac{5\pi}{180}\right)^2\ \left(\tfrac{\pi}{180}\right)^2\ \left(\tfrac{5\pi}{180}\right)^2\ \left(\tfrac{\pi}{180}\right)^2\ (0,02)^2\ (0,01)^2\right)$$

The evolution of the parameter estimation curves is shown in Fig. 8.

Please cite this article as: C. Urrea, J. Pascal, Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.09.004

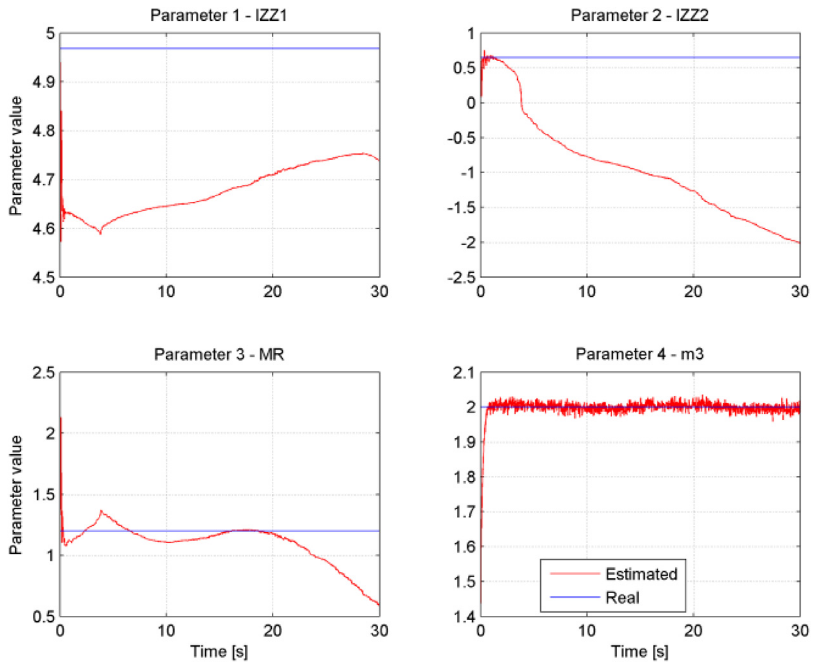

Fig. 8. Evolution of the parameters estimated by extended Kalman filter.

Table 3
Characteristics and properties of the implemented genetic algorithm.

Characteristic or parameter	Description or value
Number of data	1501
Number of equations	4503
Number of genes (individuals)	4
Number of population (chromosomes)	2
Number of generations	15,000
Evaluation function (*fitness*)	*Adjusted Fitness*
Selection	Random
Cross type	1 point
Cross probability	0.9
Mutation rate	0.1

4.5. Genetic algorithm

A linear model formulation is applied to the parameters (3). In this case, the problem consists in finding the solution to a linear equation system of the (53) form, with $3 \cdot n_{ptos}$ equations and 4 unknowns, where these unknowns are the parameters to be identified. Characteristics applied to the genetic algorithm are detailed in Table 3.

$$\begin{array}{c} I_{ZZ1}\Phi_{11}(1)+I_{ZZ2}\Phi_{12}(1)+m_r\Phi_{13}(1) \\ I_{ZZ2}\Phi_{22}(1)+m_r\Phi_{23}(1) \\ m_3\Phi_{34}(1) \\ \vdots \\ I_{ZZ1}\Phi_{11}(n)+I_{ZZ2}\Phi_{12}(n)+m_r\Phi_{13}(n) \\ I_{ZZ2}\Phi_{22}(n)+m_r\Phi_{23}(n) \\ m_3\Phi_{34}(n) \end{array} = \begin{array}{c} \tau_1(1) \\ \tau_2(1) \\ F_3(1) \\ \vdots \\ \tau_1(n) \\ \tau_2(n) \\ F_3(n) \end{array} \tag{53}$$

5. Results

Table 4 shows the base parameters identified for each of the studied methods, while Table 5 shows the errors associated with each estimated parameter. In addition, these results may be graphically observed in Fig. 9.

Please cite this article as: C. Urrea, J. Pascal, Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.09.004

Table 4
Summary of the parameters identified in the 3-DOF SCARA robot.

Identification method	Parameter			
	I_{ZZ1}	I_{ZZ2}	m_r	m_3
Real value	4.9680	0.6480	1.200	2.0000
Least squares	4.7825	0.6305	1.1243	1.9999
Adaline network	4.7167	0.6257	1.1041	1.9998
Hopfield network	3.9788	0.0910	1.0426	1.9997
Kalman filter	4.6792	-0.9285	1.0965	1.9965
Genetic algorithm	4.3837	0.0335	1.4701	2.0024

Table 5
Summary of relative errors of estimation.

Identification method	Relative error %			
	I_{ZZ1}	I_{ZZ2}	m_r	m_3
Least squares	3.7334%	2.7%	6.3054%	0.0063%
Adaline network	5.0586%	3.4449%	7.9919%	0.0079%
Hopfield network	19.9114%	85.9568%	13.1167%	0.015%
Kalman filter	5.8152%	234.287%	8.6%	0.175%
Genetic algorithm	11.7613%	94.8302%	22.5083%	0.12%

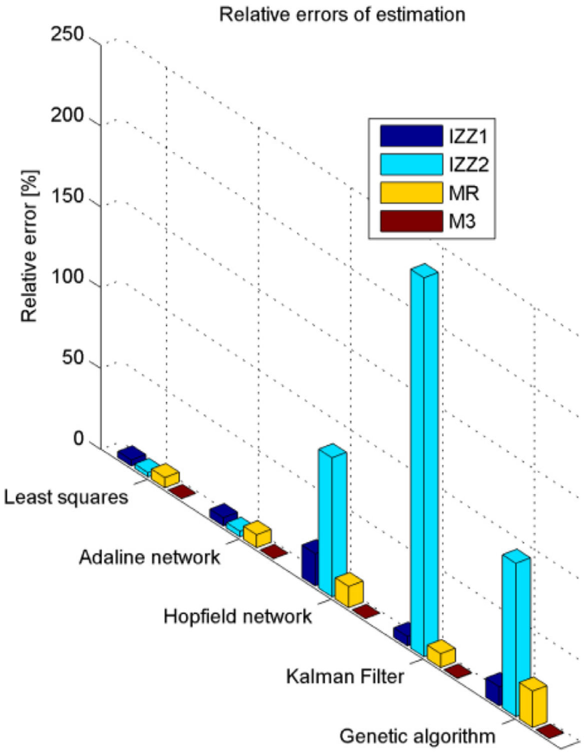

Fig. 9. Relative Errors of estimation.

Since the second parameter estimate I_{ZZ2} in the Hopfield networks, Kalman filter and genetic algorithm methods are far from the real value, the trajectories diverge when conducting simulations. Therefore, only the parameters identified by the least squares and Adaline neural networks methods are used for the validation trajectories, which are shown in Fig. 10.

It is observed that the trajectories performed by the manipulator robot when the identified parameters are used overlap with the trajectories performed when using real parameters.

Please cite this article as: C. Urrea, J. Pascal, Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.09.004

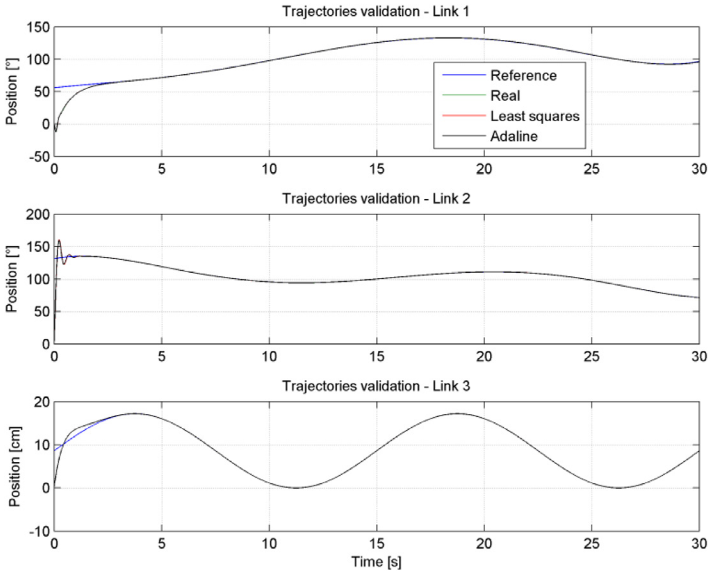

Fig. 10. Trajectories validation.

Table 6
Performance indicators.

Identification method	Indicator								
	RMS			RSD			AI		
	Link 1	Link 2	Link 3	Link 1	Link 2	Link 3	Link 1	Link 2	Link 3
Least squares	0.0003	0.0012	0.0000	0.0005	0.0021	0.0000	1.0000	1.0000	1.0000
Adaline network	0.0004	0.0007	0.0000	0.0006	0.0012	0.0000	1.0000	1.0000	1.0000
Hopfield network	NA	NA	NA	NA	NA	NA	NA	NA	NA
Kalman Filter	NA	NA	NA	NA	NA	NA	NA	NA	NA
Genetic Algorithm	NA	NA	NA	NA	NA	NA	NA	NA	NA

*NA: Not applicable

The measures used to indicate the error between the desired outputs or observed values and the predicted or estimated outputs are: residual standard deviation (RSD) (26), root mean square (RMS) (27) and adequacy index (AI) (28).

$$\mathrm{RSD} = \sqrt{\frac{\sum_{i=1}^{n} (o_i - p_i)^2}{n}} \tag{26}$$

$$\mathrm{RMS} = \sqrt{\frac{\sum_{i=1}^{n} (o_i - p_i)^2}{\sum_{i=1}^{n} (o_i)^2}} \tag{27}$$

$$\mathrm{AI} = 1 - \frac{\sum_{i=1}^{n} (o_i - p_i)^2}{\sum_{i=1}^{n} (|o_i{}'| - |p_i{}'|)^2} \tag{28}$$

where o_i are the observed values, p_i are the predicted values, n is the total number of observations, o_m is the observations' mean value, $p_i' = p_i - o_m$ and $o_i' = o_i - o_m$. The RMS and RSD indexes are standardized percentage values between 0 and 1, which indicate dispersion and deviation from the series, respectively. In addition, the IA index indicates the tendency of the two series to be compared. It may take values between 0 and 1, with 1 indicating total adequacy; and 0 indicating total inadequacy. These performance indicators for each trajectory are presented in Table 6.

In the methods simulated with the estimated parameters of least squares and Adaline network, lower error indexes were obtained in each link of the manipulator robot, i.e. RMS and RSD indicators are close to zero, and the AI indicator is close to unity.

Table 7
Computational time and number of iterations.

Identification method	Parameter	
	Computational time [s]	Number of iterations
Least squares	2	–
Adaline network	2	2
Hopfield network	17	–
Kalman filter	222	1501
Genetic algorithm	6	15,000

Table 8
Parameter sensitivity.

Identification method	Parameter range							
	I_{ZZ1}		I_{ZZ2}		m_r		m_3	
	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
Least squares	3.8260	5.7390	0.5044	0.7566	0.8994	1.3492	1.5999	2.3999
Adaline network	3.7734	5.6600	0.5006	0.7508	0.8833	1.3249	1.5998	2.3998
Hopfield network	NA	NA	NA	NA	NA	NA	NA	NA
Kalman filter	NA	NA	NA	NA	NA	NA	NA	NA
Genetic algorithm	NA	NA	NA	NA	NA	NA	NA	NA

*NA: Not applicable

Table 7 shows a summary of the computational time and number of iterations of each identification method used.

Overall, the importance of parameter values in the mathematical model of a system [24] is systematically investigated in the parametric sensitivity analysis. Table 8 presents the value range of identified parameters within which validation trajectories can be simulated, considering a maximum variation of 20% of the estimated value [25].

6. Conclusions

The results for least squares and Adaline networks are satisfactory, since relative errors by parameter are below 8%, as the validation trajectories and the performance indicators RMS, RSD and AI are.

In the case of Hopfield neural networks, the estimated parameters present an oscillation regarding its convergence, due to the presence of noise in the measurement.

Alternatively, Kalman filter is too sensitive to initial conditions. On the one hand, prior knowledge of the parameters to be estimated is necessary to obtain a good identification. On the other hand, since this procedure is recursive and uses symbolic variables, the computing time is longer in comparison with other methods.

During identification, it was perceived that results were always satisfactory for the third link, regardless the applied method, due to its linear nature. The reverse was true for the other parameters belonging to rotary links, with I_{ZZ2} presenting the highest sensitivity and error.

As a conclusion, this work demonstrated the possibility of adequately selecting diverse identification methods in order to obtain parameters characterizing the industrial robot dynamics, specifically in SCARA robots. Hence, the knowledge of this robot base parameters values enhances the design of new control methods, since the robot characteristic dynamic model is now known.

Acknowledgments

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

References

[1] Kessman K. System identification: an introduction. 1st ed. London: Springer-Verlag; 2011. doi:10.1007/978-0-85729-522-4.
[2] Nagar SK, Nagaraja T. A modified parameter identification algorithm in state space. Comput Electr Eng 1981;8:103–7. doi:10.1016/0045-7906(81)90024-0.
[3] Zhihong M. Parameter estimation of continuous linear systems using functional approximation. Comput Electr Eng 1995;21:183–7. doi:10.1016/0045-7906(95)00003-D.
[4] Hwang R-Y, Shih Y-P. Parameter identification of discrete systems via discrete legendre polynomials. Comput Electr Eng 1986;12:155–60. doi:10.1016/0045-7906(86)90007-8.
[5] Wu J, Wang J, You Z. An overview of dynamic parameter identification of robots. Robot Comput Integr Manuf 2010;26:414–19. doi:10.1016/j.rcim.2010.03.013.
[6] Jiang J, Zhang Y. A revisit to block and recursive least squares for parameter estimation. Comput Electr Eng 2004;30:403–16. doi:10.1016/j.compeleceng.2004.05.002.

[7] Radkhah K, Kulic D, Croft E. Dynamic parameters identification for the CRS A460 robot. In: IEEE/RSJ international conference on intelligent robot and systems; 2007. p. 3842–7.
[8] Sinha SK, Nagaraja T. Extended Kalman filter algorithm for continuous system parameter identification. Comput Electr Eng 1990;16:51–64. doi:10.1016/0045-7906(90)90008-4.
[9] Poignet P, Gautier M. Comparison of weighted least squares and extended Kalman filtering methods for dynamic identification of robots. In: IEEE international conference on robotics and automation. IEEE; 2000. p. 3622–7. 4. doi:10.1109/ROBOT.2000.845296.
[10] Bona B, Curatella A. Identification of industrial robot parameters for advanced model-based controllers design. Proc. eedings of the 2005 IEEE Intint. ernational Confconf. erence on Robotrobot. ics and Automautomation., IEEE; n.d., p. 1681–6. doi:10.1109/ROBOT.2005.1570355.
[11] Janot A, Vandanjon P-O, Gautier M. A generic instrumental variable approach for industrial robot identification. IEEE Trans Control Syst Technol 2014;22:132–45. doi:10.1109/TCST.2013.2246163.
[12] Jiang Z-H, Ishida T, Sunawada M. Neural network aided dynamic parameter identification of robot manipulators. In: IEEE international conference on systems, man and cybernetics; 2006. p. 3298–303. vol. 4, IEEE; 2006. doi:10.1109/ICSMC.2006.384627.
[13] Gautier M, Janot A, Vandanjon P-O. A new closed-loop output error method for parameter identification of robot dynamics. IEEE Trans Control Syst Technol 2013;21:428–44. doi:10.1109/TCST.2012.2185697.
[14] Mayeda H, Yoshida K, Osuka K. Base parameters of manipulator dynamic models. IEEE Trans Robot Autom 1990;6:312–21. doi:10.1109/70.56663.
[15] Khalil W, Gautier M, Lemoine P. Identification of the payload inertial parameters of industrial manipulators. In: IEEE international conference on robotics and automation. IEEE; 2007. p. 4943–8. doi:10.1109/ROBOT.2007.364241.
[16] Ha I-J, Ko M-S, Kwon SK. An efficient estimation algorithm for the model parameters of robotic manipulators. IEEE Trans Robot Autom 1989;5:386–94. doi:10.1109/70.34777.
[17] Kröse B, Van der Smagt P. An introduction to neural networks. 1996.
[18] Alonso H, Mendonça T, Rocha P. Hopfield neural networks for on-line parameter estimation. Neural Netw 2009;22:450–62. doi:10.1016/j.neunet.2009.01.015.
[19] Pourrajabian A, Ebrahimi R, Mirzaei M, Shams M. Applying genetic algorithms for solving nonlinear algebraic equations. Appl Math Comput 2013;219:11483–94. doi:10.1016/j.amc.2013.05.057.
[20] Contreras I. Una aproximación práctica a los algoritmos genéticos. Santiago, Chile: Universidad de Santiago de Chile; 2000.
[21] Mester G. Adaptive force and position control of rigid-link flexible-joint SCARA robots. In: Conference on IEEE industrial electronics. IEEE; 1994. p. 1639–44. 3. doi:10.1109/IECON.1994.398059.
[22] Gautier M. Optimal motion planning for robot's inertial parameters identification. In: IEEE conference on decision and control. IEEE; 1992. p. 70–3. doi:10.1109/CDC.1992.371788.
[23] Benimeli F. Estimación de parámetros dinámicos en robots manipuladores. Valencia: Universidad Politécnica de Valencia; 2005.
[24] Khalil H. Nonlinear systems. New Jersey: Prentice Hall; 2002.
[25] Abellán A. Use and validation of a three-dimensional simulation model to quantify the rockfall residual hazard on the funicular railway in vall de nuria. Universitat Politècnica de Catalunya; 2003.

Claudio Urrea was born in Santiago, Chile. 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. He is currently Director of the Doctorate in Engineering Sciences, Major in Automation, at the Universidad de Santiago de Chile.

José Pascal was born in Santiago, Chile. He is an Electrical Engineer graduated from Universidad de Santiago de Chile, Santiago, Chile, in 2016. He is currently a student of the Doctorate in Engineering Sciences program, Major in Automation, at Universidad de Santiago de Chile, and has actively participated in research on the development and application of identification methods for robots.

Please cite this article as: C. Urrea, J. Pascal, Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.09.004