

# Learning of operator hand movements via least angle regression to be teached in a manipulator

José de Jesús Rubio<sup>1</sup> · Enrique Garcia<sup>1</sup> · Gustavo Aquino<sup>1</sup> · Carlos Aguilar-Ibañez<sup>2</sup> · Jaime Pacheco<sup>1</sup> · Alejandro Zacarias<sup>1</sup>

Received: 7 November 2017 / Accepted: 24 March 2018  
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

## Abstract

In this document, a control system is developed to allow a manipulator to learn and plan references from demonstrations given by an operator hand. Data entry is acquired by a sensor and is learned by the generalized learning model with least angle regression to create a desired reference in three dimensional space. A fifth reference profile is employed to smooth the desired reference. Direct and inverse kinematics are gotten to represent the transformation between the three dimensional space and each of the manipulator links. A dynamic model is gotten using Newton–Euler formulation. An evolving proportional derivative (PD) control is applied to get that the manipulator end effector follows the operator hand movements. The monitoring and control systems are implemented in an embedded platform for testing purposes.

**Keywords** Manipulator · Least angle regression · Reference · Kinematics · Model · Embedded platform

## 1 Introduction

In these days, there exists many manipulators which make industrial home works and they have not been programed to make these activities. The ability of reprogramming in the manipulators gives more autonomy with the environment. This suggests that an operator develops acceptable movements to be considered as the desired reference to be learned by a manipulator.

There exists interesting studies made in controls applied to robotic systems. In Calderon et al. (2017), Li et al. (2017), and Rosado et al. (2017), controllers for prosthesis are developed. In Garcia et al. (2017), Liu and Liu (2017), Sa et al. (2017), and Rubio et al. (2016), controllers to be applied to robotic arms are recommended. In Hernandez et al. (2017),

Serrano et al. (2018), and Zhang et al. (2014), reference tracking of mobile manipulators are advised. In CarranzaM (2017) and Peng et al. (2017), controllers applied in pendulums are focused. The mentioned documents show that the control could help to get a manipulator which learn by demonstration.

There exists interesting studies about the learning of nonlinear systems. Evolving intelligent systems are distinguished by skills to adjust their parameters and structure to the varying features of environment. Some important results are described by Angelov and Yager (2012), Angelov et al. (2010), Iliadis et al. (2016), Kangin et al. (2016), Kasabov (2007), Khamassi et al. (2016), Maciel et al. (2016), Silva et al. (2014), Pratama et al. (2016), Pratama et al. (2016), and Mouchaweh and Lughofer (2012) for the theory, and in Abdallah et al. (2016), Angelov et al. (2011), Angelov (2012), Lughofer (2013), Lughofer et al. (2015), Lughofer (2016), Precup et al. (2012), Precup et al. (2014), and Venkatesan et al. (2016) for the applications. This document is focused in the applications. The detection of activity recognition is focused in Abdallah et al. (2016). In Angelov et al. (2011), an approach for automatic detection, object identification, and tracking in video streams is advised. Some models in real time have been implemented in Angelov (2012). In Lughofer (2013), a criteria for prediction is recommended.

✉ José de Jesús Rubio  
jrubio@ipn.mx; rubio.josedejesus@gmail.com

<sup>1</sup> Sección de Estudios de Posgrado e Investigación, Esime Azcapotzalco, Instituto Politécnico Nacional, Av. de las Granjas no. 682, Col. Santa Catarina, 02250 Mexico D.F., Mexico

<sup>2</sup> Centro de Investigación en Computación, Instituto Politécnico Nacional, Av. Juan de Dios Bátiz s/n, Col. San Pedro Zacatenco, 07738 Mexico D.F., Mexico

A methodology of identification problems is discussed in Lughofer et al. (2015). In Lughofer (2016), the usability of some models is analyzed. The control of tank systems is considered in Precup et al. (2012). In Precup et al. (2014), a model of crane systems is addressed. A classifier of high-speed streaming data is applied in Venkatesan et al. (2016). From the aforementioned papers, in Abdallah et al. (2016), Angelov and Yager (2012), Angelov et al. (2011), Angelov et al. (2010), Angelov (2012), Kangin et al. (2016), Khamassi et al. (2016), Lughofer (2013), Lughofer et al. (2015), Precup et al. (2012), and Precup et al. (2014), authors use several kind of learning algorithms; since they are highly used to adjust parameters of evolving intelligent systems, it would be desired to employ a learning algorithm to learn desired movements to be teached in a manipulator.

One strategy to perform a homework is to copy the movement of an operator hand. The solution advised in this document is to employ one sensor with a learning algorithm for which, from the learning of an operator hand, the manipulator makes iterative activities. The advantage of this technique is that the operator does not need to have knowledge about programming.

The first main contribution of this document is to propose a least angle regression (LAR) with a generalized learning model to learn the movements of an operator hand to generate desired references which will be applied in a manipulator. In specific, the movements received by an operator hand, are learned by the generalized learning model and least angle regression to generate a desired reference in three dimensional space. Furthermore, the least angle regression (LAR) and the L1 penalized regression (LASSO) are compared for two references to select the best.

The second main contribution of this document is to propose a fifth reference profile which include a constant term known as the number of transitory periods that can be chosen by an operator such that the manipulator reference must be softened, and it avoids dangerous movements on the manipulator. In specific, a three dimensional reference is smoothed with a fifth reference profile.

The third main contribution of this document is the suggestion of dynamics in the manipulator, it consists in getting the direct and inverse kinematics, and dynamic model of a manipulator with six degrees of freedom. In specific, a tridimensional smooth reference is employed in the inverse kinematics and dynamic model to get the desired references for each manipulator link.

The last main contribution of this document is the develop of an evolving proportional derivative (PD) control which is employed to get that the manipulator end effector follows the operator hand movements. In specific, the evolving proportional derivative (PD) controller is

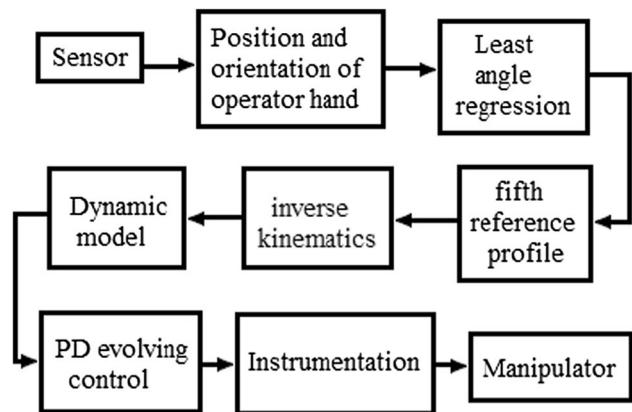
employed to get that each manipulator link follows each desired reference.

The document is structured in the next sentence. In Sect. 2, the all elements for the evolving PD control of the manipulator are focused. In Sect. 3, the performance of the evolving PD controller employed to get that the manipulator links follow desired references is taken into account. Finally, in Sect. 4 conclusions and future studies of the topic are explained.

## 2 Control for the manipulator which learns by demonstration

Figure 1 shows all elements which let the control of the manipulator. The goal is that the manipulator will follow a movement made by the operator hand. This procedure is described in the next sentence. The operator hand specifies the desired reference, for which, the sensor takes data from a operator hand. The movements received by the sensor, are learned by the generalized learning model and least angle regression to generate a desired reference in three dimensional space. Later, this three dimensional reference is smoothed with a fifth reference profile. After, this tridimensional smooth reference is employed in the inverse kinematics and dynamic model to get the desired references for each manipulator link. The evolving proportional derivative (PD) controller and the instrumentation are employed to get that each manipulator link follows each desired reference.

The next subsections will describe the elements which are divided in two important parts: the automatic learning and dynamics.



**Fig. 1** Flow diagram of the control system

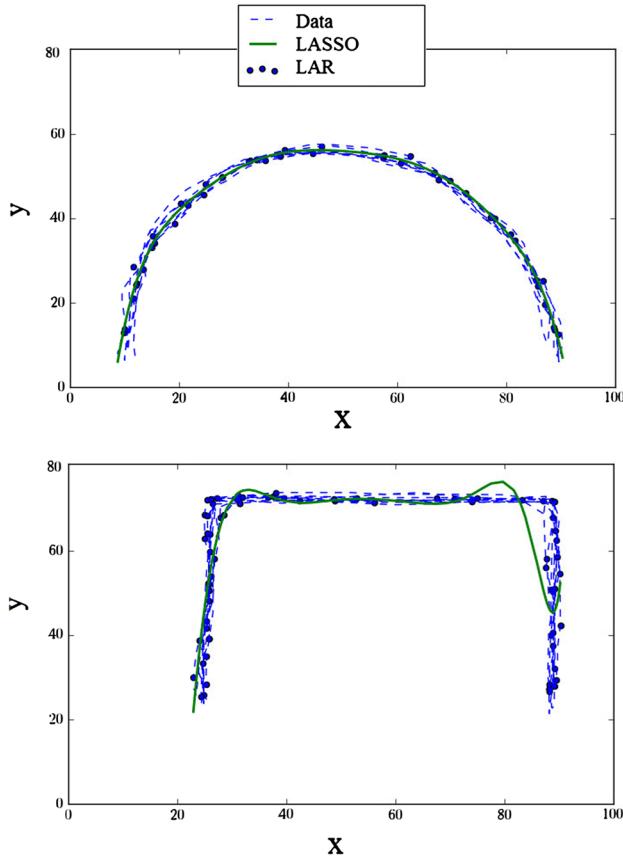
## 2.1 The automatic learning

The movements received by the sensor, are learned by the generalized learning model and least angle regression to generate a desired reference in the three dimensional space. The generalized learning model is employed to approximate a desired reference in three dimensional space, while the least angle regression is employed to update the parameters of the generalized learning model to adapt it to the desired reference. Later, this three dimensional reference is smoothed with a fifth reference profile.

The target of this subsection is to describe the generalized learning model and reference profile to get the position and orientation of the operator hand. The generated reference will be employed in the control of the next subsection.

### 2.1.1 The generalized learning model

Supervised generalized learning algorithms try to predict an outcome  $y$  from previous examples  $X$ . Linear regression models are also known as line best fit which can predict a real-valued number from data. In case there are several factors that affect the outcome and in order to improve accuracy we



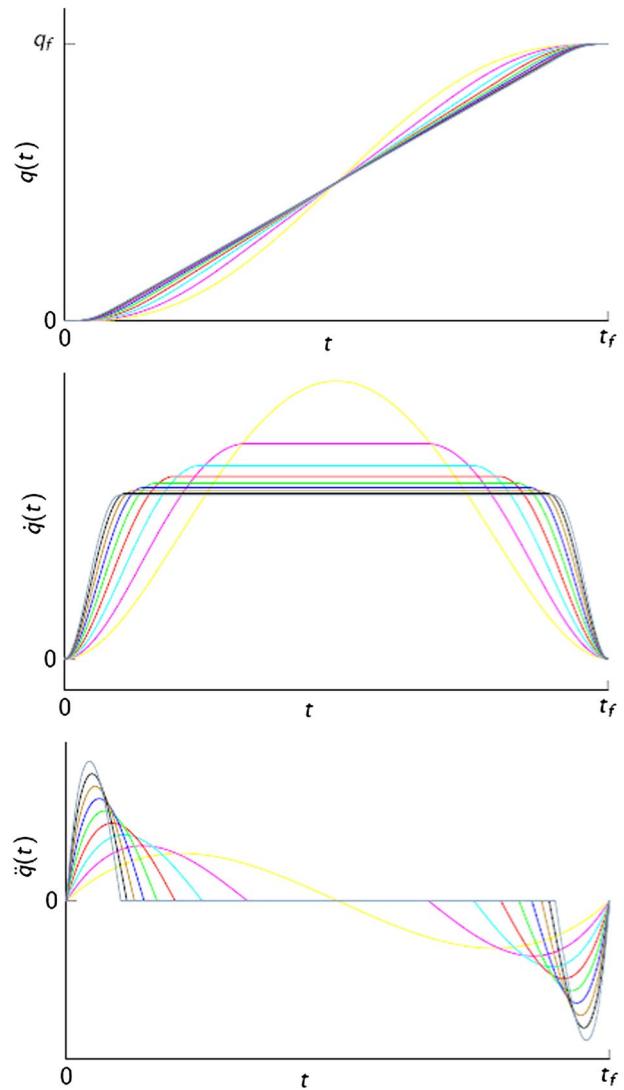
**Fig. 2** Comparison between the LAR and LASSO

use single output regression. As  $X$  represents the input data and  $y$  the output or targets then:

$$X = \begin{bmatrix} x_{11} & x_{21} & \dots & x_{d1} \\ x_{12} & x_{22} & \dots & x_{d2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1m} & x_{2m} & \dots & x_{dm} \end{bmatrix}$$

$$y = [y_1 \ y_2 \ \dots \ y_m]^T \quad (1)$$

where  $d$  is the number of inputs or features and  $m$  is the number of samples or training examples. For simple linear regression the prediction model is  $\hat{y} = ax + b$ , but if  $w$  represents the weights  $w = [w_1 \ w_2 \ \dots \ w_d]^T$  and  $x_i$  represent a feature vector  $x_i = [x_{1i} \ x_{2i} \ \dots \ x_{di}]^T$  both size  $d$ , then the simple regression model becomes  $\hat{y}_i = w^T x_i + b$ . If  $w_0 = b$



**Fig. 3** Trajectories for several values of  $n$

and  $x_0 = 1$  we can rewrite  $w$  as  $w = [w_0 \ w_1 \ w_2 \ \dots \ w_d]^T$  and  $x_i$  as  $x_i = [x_0 \ x_{1i} \ x_{2i} \ \dots \ x_{di}]^T$ , then the prediction model for single output regression becomes:

$$\hat{y}(k) = X(k)w(k) \quad (2)$$

### 2.1.2 The L1 penalized regression (LASSO)

Adding a column of completely random noise can improve our model so that the mean squared error  $MSE = \frac{1}{m} \sum_{i=1}^m (y_i(k) - \hat{y}_i(k))^2$  will be smaller, in fact adding more  $x^n$  features as in Taylor series can approximate any function, however besides the computational complexity making the code running slower, a high degree polynomial does not guarantee overfitting, it all depends on training data, in general we want  $d << m$ . The goal of L1 regularization is to select a small number of important features that predict the trend removing all the features that are just noise, this often is called sparsity, so the end result will be zero and only few of the weights will be non-zero. From the cost function for simple linear regression  $J(k) = \sum_{i=1}^m (y_i(k) - \hat{y}_i(k))^2$  L1 regularization uses a penalty term  $\lambda|w(k)|$ :

$$J(k) = \sum_{i=1}^m (y_i(k) - \hat{y}_i)^2 + \lambda|w(k)| \quad (3)$$

Vectorizing  $J(k)$ , Eq. (3) can be rewritten as  $J(k) = (y(k) - X(k)w(k))^T(y(k) - X(k)w(k)) + \lambda|w(k)| = y(k)^T y(k) - 2y^T(k)X(k)w(k) + w^T(k)X^T(k)X(k)w(k) + \lambda|w(k)|$ , then taking the derivative:

$$\frac{\partial J(k)}{\partial w(k)} = -2X^T(k)y(k) + 2X^T(k)X(k)w(k) + \lambda sign(w(k)) \quad (4)$$

In order to minimize the cost function,  $\frac{\partial J(k)}{\partial w(k)}$  is set to zero. As the *sign* function does not have an inverse function we can not solve for  $w(k)$ . In order to obtain the weights we are using the gradient descent algorithm.

The gradient descent algorithm is used to minimize functions, not just the cost function or linear regression, it is used widely in generalized learning. Instead of setting  $\frac{\partial J(k)}{\partial w(k)}$  to zero and solving for  $w(k)$ , we will just take small steps in the direction of the gradient. Since 2 is a constant we can just drop it since it can be absorbed into the learning rate. Then the algorithm is described as:

$w(k)$  = get some random value from a Gaussian with

$$\text{mean } \mu = 0 \text{ and variance } \sigma^2 = \frac{1}{d}$$

Repeat until converge  $\{w(k) = w(k-1) - \alpha X^T(k)(\hat{y}(k) - y(k)) + \lambda sign(w(k))\}$

where  $\alpha$  is the learning rate often called hyperparameter which is not part of the model itself, but is used to find a solution. If it is too big it will not converge and if it is too small the gradient descent will be slow.

### 2.1.3 The least angle regression (LAR)

In Fraley and Hesterberg (2009), the predictors and response are transformed to have mean 0. This assumes an intercept of the model. The model is rewritten as follows:

$$\hat{y}(k) = X(k)w(k) \quad (6)$$

where  $\hat{y}(k)$  denotes the response with its mean subtracted, and  $X(k)$  denotes the matrix of predictors. The method proceeds iteratively in a series of steps. Take the definitions of the next sentence.  $w(k)$  is the vector of coefficients at step  $k$ ,  $B(k)$  is the active set of predictors during step  $k$ ,  $q(k)$  is the number of active predictors,  $X(k)$  are the columns of  $X$  corresponding to  $B(k)$ ,  $Q(k)$  is a matrix of zeros and ones such that  $X(k) = XP^T(k)$ ,  $e(k)$  is the vector of partial residuals at step  $k$  which is described as:

$$e(k) = \hat{y}(k) - y(k) \quad (7)$$

Initially, all coefficients  $w(1)$  are zero, and  $B(1)$  is empty. A series of models are fitted in which predictors are successively added to or dropped from the active set, and coefficients are updated. The final model has a maximal set of linearly independent predictors. At the  $k$  step, the active set is updated, and a new set of coefficients  $w(k)$  by determining a step length  $\beta(k)$  along the direction  $v(k)$  from  $w(k-1)$ :

$$w(k) = w(k-1) - \beta(k)v(k) \quad (8)$$

with  $0 \leq \beta(k) \leq 1$ . The direction  $v(k)$  comes from the least squares estimate based on the active set. Let:

$$s(k) = \arg \min_s \|X(k)s - y(k)\| \quad (9)$$

be the least squares coefficients based on the current active set. Equivalently,  $s(k)$  satisfies the normal equations:

$$X^T(k)X(k)s(k) = X(k)y(k) \quad (10)$$

The direction  $v(k)$  is the vector with length  $d$ , which is equal to  $Q(k)w(k-1) - s(k)$  in the entries corresponding to the active predictors, and 0 elsewhere:

$$v(k) = Q^T(k)(Q(k)w(k-1) - s(k)) \quad (11)$$

The step length  $\beta(k)$  to be taken from  $w(k-1)$  along  $v(k)$  to get the new set of coefficients is determined by the algorithm based on the correlations of the predictors with the partial residuals:

$$cor(X, e(k)) = \frac{D_X^{-1} X^T(k)e(k)}{\|e(k)\|_2} \quad (12)$$

where:

$$D_X = \text{diag}(\|X_{*1}(k)\|_2, \|X_{*2}(k)\|_2, \dots, \|X_{*d}(k)\|_2) \quad (13)$$

The correlations are proportional to the inner products between partial residuals and predictors, standardized by their norms. Dropping the denominator for simplicity, we work with:

$$c(k) = D_X^{-1} X^T(k) e(k) \quad (14)$$

and with the more general:

$$c(w) = D_X^{-1} X^T(k)(\hat{y}(k) - y(k)) \quad (15)$$

At the start of a step, the active set is defined to be the predictors corresponding to the correlations that have the largest magnitude. These correlations would vanish if a unit step is taken in the direction  $v(k)$ . The first predictors to enter the active set are those with the largest absolute correlations with  $y(k)$ .

The step length  $\beta(k)$  in least angle regression, is the smallest step  $\beta$  such that one or more predictors that are not in the current active set  $B(k)$  for the next step have correlation equal in magnitude to the correlations of members of  $B(k)$  at  $w(k-1) - \beta(k)v(k)$ . Those predictors are added to the active set for the following step. The required computations are straightforward because:

$$c(w(k-1) - \beta(k)v(k)) = D_X^{-1} X^T(k)(\hat{y}(k) - y(k)) \quad (16)$$

is linear function of  $\beta$ . In least angle regression, predictors never leave the active set once they are added.

Figure 2 shows two examples of the comparison between the least angle regression denoted as LAR and the L1 penalized regression denoted as LASSO where the LAR improves the LASSO because the LAR follows better the real data denoted as Data. Thus, the LAR is the best option for the learning and it is chosen in this document.

#### 2.1.4 The reference profile

The manipulator must move from initial position  $q_0$  to a final position  $q_f$  during a time  $t_f$ . This transition must be regulated by concepts which use the manipulator movements, this smoothed reference is known as fifth reference profile:

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (17)$$

If  $n$  is the transitory periods of the reference, equations which define the positions of the manipulator movements are:

$$\begin{aligned} q(t) &= q_f \left( 1 - \frac{30(n-2)}{32+30(n-2)} \right) * \left[ \frac{5}{4} \left( \frac{t}{t_f} \right)^3 n^3 - \frac{15}{16} \left( \frac{t}{t_f} \right)^4 n^4 + \frac{3}{16} \left( \frac{t}{t_f} \right)^5 n^5 \right] \\ &\quad \forall t, 0 \leq t \leq \frac{t_f}{n} \\ q(t) &= q \left( \frac{t_f}{n} \right) + q \left( \frac{t_f}{n} \right) \left( t - \frac{t_f}{n} \right) \quad \forall t, \frac{t_f}{n} \leq t \leq \frac{n-1}{n} t_f \\ q(t) &= q_f \left( 1 - \frac{30(n-2)}{32+30(n-2)} \right) \\ &\quad * \left[ \frac{5}{4} \left( \frac{t - \frac{n-2}{n} t_f}{t_f} \right)^3 n^3 - \frac{15}{16} \left( \frac{t - \frac{n-2}{n} t_f}{t_f} \right)^4 n^4 + \frac{3}{16} \left( \frac{t - \frac{n-2}{n} t_f}{t_f} \right)^5 n^5 \right] \\ &\quad \forall t, \frac{n-1}{n} t_f \leq t \leq t_f \end{aligned} \quad (18)$$

Equations which define the velocities of the manipulator movements are:

$$\begin{aligned} \dot{q}(t) &= q_f \left( 1 - \frac{30(n-2)}{32+30(n-2)} \right) * \left[ \frac{15}{4} \frac{t^2}{t_f^3} n^3 - \frac{15}{4} \frac{t^3}{t_f^4} n^4 + \frac{15}{16} \frac{t^4}{t_f^5} n^5 \right] \\ &\quad \forall t, 0 \leq t \leq \frac{t_f}{n} \\ \dot{q}(t) &= q \left( \frac{t_f}{n} \right) \quad \forall t, \frac{t_f}{n} \leq t \leq \frac{n-1}{n} t_f \\ \dot{q}(t) &= q_f \left( 1 - \frac{30(n-2)}{32+30(n-2)} \right) \\ &\quad * \left[ \frac{15}{4} \frac{\left( t - \frac{n-2}{n} t_f \right)^2}{t_f^3} n^3 - \frac{15}{4} \frac{\left( t - \frac{n-2}{n} t_f \right)^3}{t_f^4} n^4 + \frac{15}{16} \frac{\left( t - \frac{n-2}{n} t_f \right)^4}{t_f^5} n^5 \right] \\ &\quad \forall t, \frac{n-1}{n} t_f \leq t \leq t_f \end{aligned} \quad (19)$$

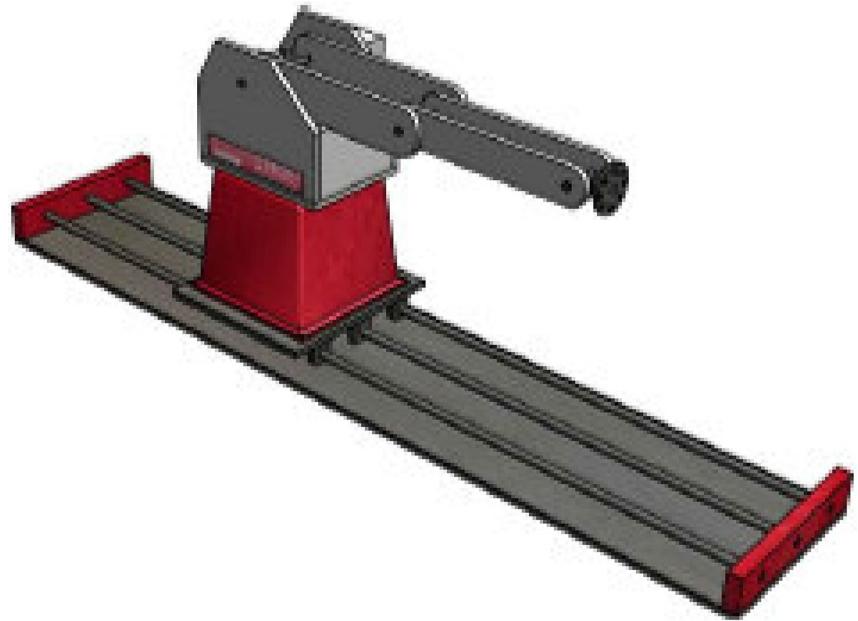
Equations which define the acceleration movements are:

$$\begin{aligned} \ddot{q}(t) &= q_f \left( 1 - \frac{30(n-2)}{32+30(n-2)} \right) * \left[ \frac{15}{4} \frac{t}{t_f^3} n^3 - \frac{45}{4} \frac{t^2}{t_f^4} n^4 + \frac{15}{4} \frac{t^3}{t_f^5} n^5 \right] \\ &\quad \forall t, 0 \leq t \leq \frac{t_f}{n} \\ \ddot{q}(t) &= 0 \quad \forall t, \frac{t_f}{n} \leq t \leq \frac{n-1}{n} t_f \\ \ddot{q}(t) &= q_f \left( 1 - \frac{30(n-2)}{32+30(n-2)} \right) \\ &\quad * \left[ \frac{15}{4} \frac{\left( t - \frac{n-2}{n} t_f \right)}{t_f^3} n^3 - \frac{45}{4} \frac{\left( t - \frac{n-2}{n} t_f \right)^2}{t_f^4} n^4 + \frac{15}{4} \frac{\left( t - \frac{n-2}{n} t_f \right)^3}{t_f^5} n^5 \right] \\ &\quad \forall t, \frac{n-1}{n} t_f \leq t \leq t_f \end{aligned} \quad (20)$$

Figure 3 shows the graphs which describe movements of Eqs. (18), (19), and (20) for several values of  $n$  where  $n = 2$  is in yellow color,  $n = 3$  is in magnet color,  $n = 4$  is in blue color,  $n = 5$  is in orange color,  $n = 6$  is in green color,  $n = 7$  is in purple color,  $n = 8$  is in coffee color,  $n = 9$  is in black color, and  $n = 10$  is in green color.

## 2.2 The dynamics

A tridimensional smooth reference obtained by the least regression angle and fifth reference profile is employed in the inverse kinematics and dynamic model to get the desired references for each manipulator link. For this

**Fig. 4** Pegasus robot

goal, the inverse kinematics are employed to get the positions of the angles and displacements in the desired references, while the dynamic model is employed to get the velocities of the angles and displacements in the desired references. Later, the positions and velocities of the desired references are employed in the evolving PD controller to get that the manipulator links follow the desired references.

The targets of this subsection are explained in the next sentence. First, get the kinematics and dynamic model are gotten using the Neuton-Euler strategy. Second, get the reference tracking of the manipulator by using an evolving PD

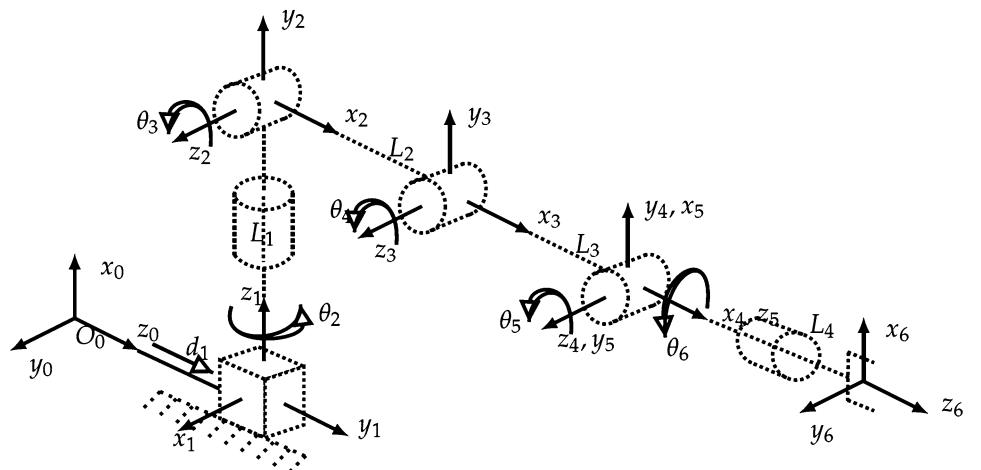
control. And third, implement an embedded control by using a NI compactRio platform.

Figure 4 shows the manipulator which is an Amatrol Pegasus robot with six degrees of freedom. The dynamics takes into account the kinematics, dynamic model, and control.

### 2.2.1 The direct kinematics

Figure 5 shows the Denavit and Hartenberg strategy to establish the coordinated system of each link.

The result of the direct kinematics is described in the next equation:

**Fig. 5** Direct kinematics

$$A_{eq} = A_1 A_2 A_3 A_4 A_5 A_6$$

$$= \begin{bmatrix} c_6 & -s_{345}s_6 & -c_{345} & L_1 + L_2 s_3 + L_3 s_{34} - L_4 c_{345} \\ c_2 c_{345} c_6 + s_2 s_6 & c_6 s_2 - c_2 c_{345} s_6 & c_2 s_{345} & c_2 (L_2 c_3 + L_3 c_{34} + L_4 s_{345}) \\ c_{345} c_6 s_2 - c_2 s_6 & -c_2 c_6 - c_{345} s_2 s_6 & s_2 s_{345} & d_1 + s_2 (L_2 c_3 + L_3 c_{34} + L_4 s_{345}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

where  $c_i = \cos(\theta_i)$ ,  $s_{ij} = \sin(\theta_i + \theta_j)$ ,  $c_{ijk} = \cos(\theta_i + \theta_j + \theta_k)$ ,  $s_i = \sin(\theta_i)$ ,  $s_{ij} = \sin(\theta_i + \theta_j)$ , and  $s_{ijk} = \sin(\theta_i + \theta_j + \theta_k)$ .

## 2.2.2 The inverse kinematics

Set  $p_j = [p_{xj}, p_{yj}, p_{zj}]$  as the position vector measured from the origin of j-th coordinated system until the sixth coordinated system. Set  $u_j = [u_{xj}, u_{yj}, u_{zj}]$  as one unitary vector with respect to the axis  $z_6$  and expressed in the j-th coordinated system. Then, by definition is:

$$\begin{aligned} p_6 &= [0 \ 0 \ 0 \ 1]^T \\ u_6 &= [0 \ 0 \ 1 \ 0]^T \end{aligned} \quad (22)$$

When they are expressed in the first coordinated system:

$$\begin{aligned} p_1 &= A_1 A_2 A_3 A_4 A_5 p_6 \\ u_1 &= A_1 A_2 A_3 A_4 A_5 u_6 \end{aligned} \quad (23)$$

From Eq. (23), the next transformation is used:

$$\begin{aligned} A_2^{-1} A_1^{-1} p_1 &= A_3 A_4 A_5 p_6 \\ A_2^{-1} A_1^{-1} u_1 &= A_3 A_4 A_5 u_6 \end{aligned} \quad (24)$$

From transformation (24), the equations system is:

$$\begin{aligned} -L_3 s_3 s_4 + c_3 (L_2 + L_3 c_4) &= q c_2 + (r - d_1) s_2 \\ L_3 c_3 s_4 + s_3 (L_2 + L_3 c_4) &= p - L_1 \\ q s_2 - (r - d_1) c_2 &= 0 \\ c_3 (s_4 c_5 + c_4 s_5) + s_3 (c_4 c_5 - s_4 s_5) &= v c_2 + w s_2 \\ s_3 (s_4 c_5 + c_4 s_5) - c_3 (c_4 c_5 - s_4 s_5) &= u \\ -w c_2 + v s_2 &= 0 \end{aligned} \quad (25)$$

The solution of Eq. (25) lets to find the next elements of the inverse kinematics:

$$\theta_2 = \arctan 2(w, v) = \begin{cases} v.w & \text{para } 0 \leq \frac{\pi}{2} \\ -v, w & \text{para } \frac{\pi}{2} \leq \pi \\ -v, -w & \text{para } -\pi \leq -\frac{\pi}{2} \\ v, -w & \text{para } -\frac{\pi}{2} \leq 0 \end{cases}$$

$$d_1 = r - q \tan \theta_2$$

$$\theta_4 = \arctan 2 \left( \sqrt{1 - \left( \frac{qv + w(r - d_1)^2 + p - L_1^2 - L_2^2 - L_3^2}{2L_2 L_3} \right)^2}, \frac{qv + w(r - d_1)^2 + p - L_1^2 - L_2^2 - L_3^2}{2L_2 L_3} \right)$$

$$\theta_3 = \arctan 2(\sqrt{1 - c_3^2}, c_3)$$

$$\theta_5 = \arctan 2(\sqrt{1 - f_8^2}, f_8)$$

$$\theta_6 = \arctan 2(l_y s_2 - l_z c_2, l_x s_{345} + c_{345}(l_y c_2 + l_z s_2)) \quad (26)$$

$$\text{w h e r e } f_8 = \frac{f_7(s_4 c_3 + c_4 s_3) + \sqrt{(c_4 c_3 - s_4 s_3)^2 [(s_4^2 + c_4^2)(c_3^2 + s_3^2) - f_7^2]}}{(s_4^2 + c_4^2)(c_3^2 + s_3^2)},$$

$$f_7 = c_3(s_4 c_5 + c_4 s_5) + s_3(c_4 c_5 - s_4 s_5).$$

## 2.2.3 The dynamic model

The Newton–Euler strategy is employed to get the dynamic model. The rotation of the system with reference 0 is:

$$z_0 = [0 \ 0 \ 1]^T \quad (27)$$

and rotation axes for links are:

$$\begin{aligned} b_1 &= (R_1^0)^T z_0 = [0 \ 1 \ 0]^T \\ b_2 &= (R_2^0)^T R_1^0 z_0 = [0 \ 0 \ 1]^T \\ b_3 &= (R_3^0)^T R_2^0 z_0 = [0 \ 0 \ 0]^T \\ b_4 &= (R_4^0)^T R_3^0 z_0 = [0 \ 0 \ 0]^T \\ b_5 &= (R_5^0)^T R_4^0 z_0 = [0 \ 0 \ 0]^T \\ b_6 &= (R_6^0)^T R_5^0 z_0 = [0 \ 0 \ 0]^T \end{aligned} \quad (28)$$

From the forward formulation, results are:

$$\begin{aligned}
w_0 &= \alpha_0 = a_{c0} = a_{e0} = 0 \\
w_1 &= b_1 q_1 \\
\alpha_1 &= b_1 \ddot{q}_1 + w_1 + b_1 \dot{q}_1 \\
a_{e1} &= w_1 \times r_{01} + w_1 \times (w_1 \times r_{01}) \\
a_{c1} &= w_1 \times r_{0c1} + w_1 \times (w_1 \times r_{0c1}) \\
w_2 &= (R_2^1)^T w_1 + b_2 \dot{q}_2 \\
\alpha_2 &= (R_2^1)^T \alpha_1 + b_2 \ddot{q}_2 + w_2 + b_2 \dot{q}_2 \\
a_{e2} &= (R_2^1)^T a_{e1} + w_2 \times r_{12} + w_2 \times (w_2 \times r_{12}) \\
a_{c2} &= (R_2^1)^T a_{c1} + w_2 \times r_{1c2} + w_2 \times (w_2 \times r_{1c2}) \\
w_3 &= (R_3^2)^T w_2 + b_3 \dot{q}_3 \\
\alpha_3 &= (R_3^2)^T \alpha_2 + b_3 \ddot{q}_3 + w_3 + b_3 \dot{q}_3 \\
a_{e3} &= (R_3^2)^T a_{e2} \\
a_{c3} &= (R_3^2)^T a_{c2} \\
w_4 &= (R_4^3)^T w_3 + b_4 \dot{q}_4 \\
\alpha_4 &= (R_4^3)^T \alpha_3 + b_4 \ddot{q}_4 + w_4 + b_4 \dot{q}_4 \\
a_{e4} &= (R_4^3)^T a_{e3} + w_4 \times r_{34} + w_4 \times (w_4 \times r_{34}) \\
a_{c4} &= (R_4^3)^T a_{c3} + w_4 \times r_{3c4} + w_4 \times (w_4 \times r_{3c4}) \\
w_5 &= (R_5^4)^T w_4 + b_5 \dot{q}_5 \\
\alpha_5 &= (R_5^4)^T \alpha_4 + b_5 \ddot{q}_5 + w_5 + b_5 \dot{q}_5 \\
a_{e5} &= (R_5^4)^T a_{e4} \\
a_{c5} &= (R_5^4)^T a_{c4} \\
w_6 &= (R_6^5)^T w_5 + b_6 \dot{q}_5 \\
\alpha_6 &= (R_6^5)^T \alpha_5 + b_6 \ddot{q}_5 + w_6 + b_6 \dot{q}_6 \\
a_{e6} &= (R_6^5)^T a_{e5} + w_6 \times r_{56} + w_6 \times (w_6 \times r_{56}) \\
a_{c6} &= (R_6^5)^T a_{c5} + w_6 \times r_{5c6} + w_6 \times (w_6 \times r_{5c6})
\end{aligned} \tag{30}$$

From the backward formulation, results are:

$$\begin{aligned}
F_7 &= \tau_7 = 0 \\
g_0 &= [0 \ 0 \ -g]^T \\
g_6 &= (R_6^0)^T g_0 \\
F_6 &= m_6 a_{c6} - m_6 g_6 \\
\tau_6 &= -F_6 \times r_{5c6} + w_6 \times (l_6 w_6) + l_6 \alpha_6 \\
g_5 &= (R_5^0)^T g_0 \\
F_5 &= R_5^5 F_6 \\
\tau_5 &= R_5^5 \tau_6 + w_5 \times (l_5 w_5) + l_5 \alpha_5 \\
g_4 &= (R_4^0)^T g_0 \\
F_4 &= R_4^4 F_5 + m_4 a_{c4} - m_4 g_4 \\
\tau_5 &= R_5^4 \tau_5 - F_4 \times r_{3c4} + R_5^4 F_5 \times r_{4c4} + w_4 \times (l_4 w_4) + l_4 \alpha_4
\end{aligned} \tag{31}$$

$$\begin{aligned}
g_3 &= (R_3^0)^T g_0 \\
F_3 &= R_3^3 F_4 \\
\tau_3 &= R_3^3 \tau_4 + w_3 \times (l_3 w_4) + l_3 \alpha_3 \\
g_2 &= (R_2^0)^T g_0 \\
F_2 &= R_2^2 F_3 + m_2 a_{c2} - m_2 g_2 \\
\tau_2 &= R_2^2 \tau_3 - F_2 \times r_{1c2} + R_2^2 F_3 \times r_{2c2} + w_2 \times (l_2 w_2) + l_2 \alpha_2 \\
g_1 &= (R_1^0)^T g_0 \\
F_1 &= R_1^1 F_2 + m_1 a_{c1} - m_1 g_1 \\
\tau_1 &= R_1^1 \tau_2 - F_1 \times r_{0c1} + R_1^1 F_2 \times r_{1c1} + w_1 \times (l_1 w_1) + l_1 \alpha_1
\end{aligned} \tag{32}$$

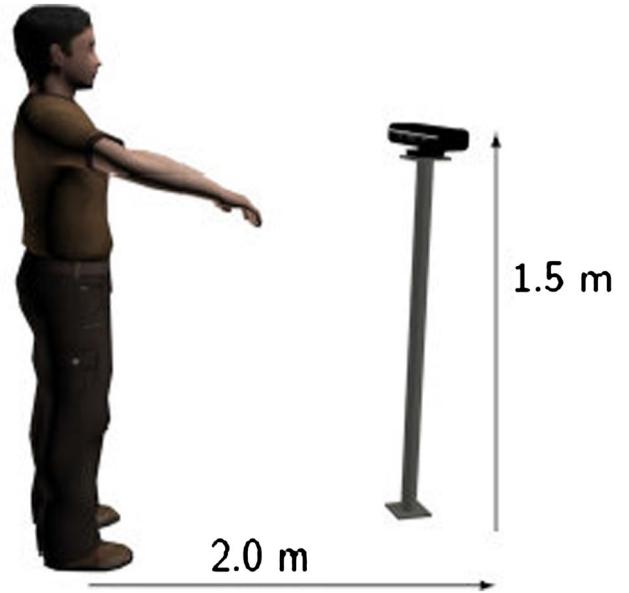
The dynamic model of the manipulator of Eqs. (31) and (32) can be rewritten as:

$$\begin{aligned}
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) &= u \\
q &= [q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ q_6]^T \\
u &= [\tau_1 \ \tau_2 \ \tau_3 \ \tau_4 \ \tau_5 \ \tau_6]^T
\end{aligned} \tag{33}$$

where  $M(q) \in \mathbb{R}^{6 \times 6}$ ,  $C(q, \dot{q}) \in \mathbb{R}^{6 \times 6}$ , and  $G(q) \in \mathbb{R}^{6 \times 1}$ .

## 2.2.4 The system control

The evolving proportional derivative (PD) controller is employed to get that each manipulator link follows each desired reference. The evolving proportional derivative (PD) control has been shown an acceptable performance in industrial systems. The evolving PD controller is independently



**Fig. 6** Position of the sensor

applied to each of the manipulator links, i.e., each control input controls each output. The evolving PD control of each link is:

$$u_i = -K_{pi}(k)(p_{refi} - q_i) - K_{di}(k)\dot{q}_i \quad (34)$$

where  $i = 1, \dots, 6$ ,  $p_{refi}$  is the desired reference for each link  $i$ ,  $K_{pi} \in \Re$  is the proportional gain, and  $K_{di}$  is the derivative gain.

From Angelov et al. (2015), the gains of the evolving PD control are adapted as:

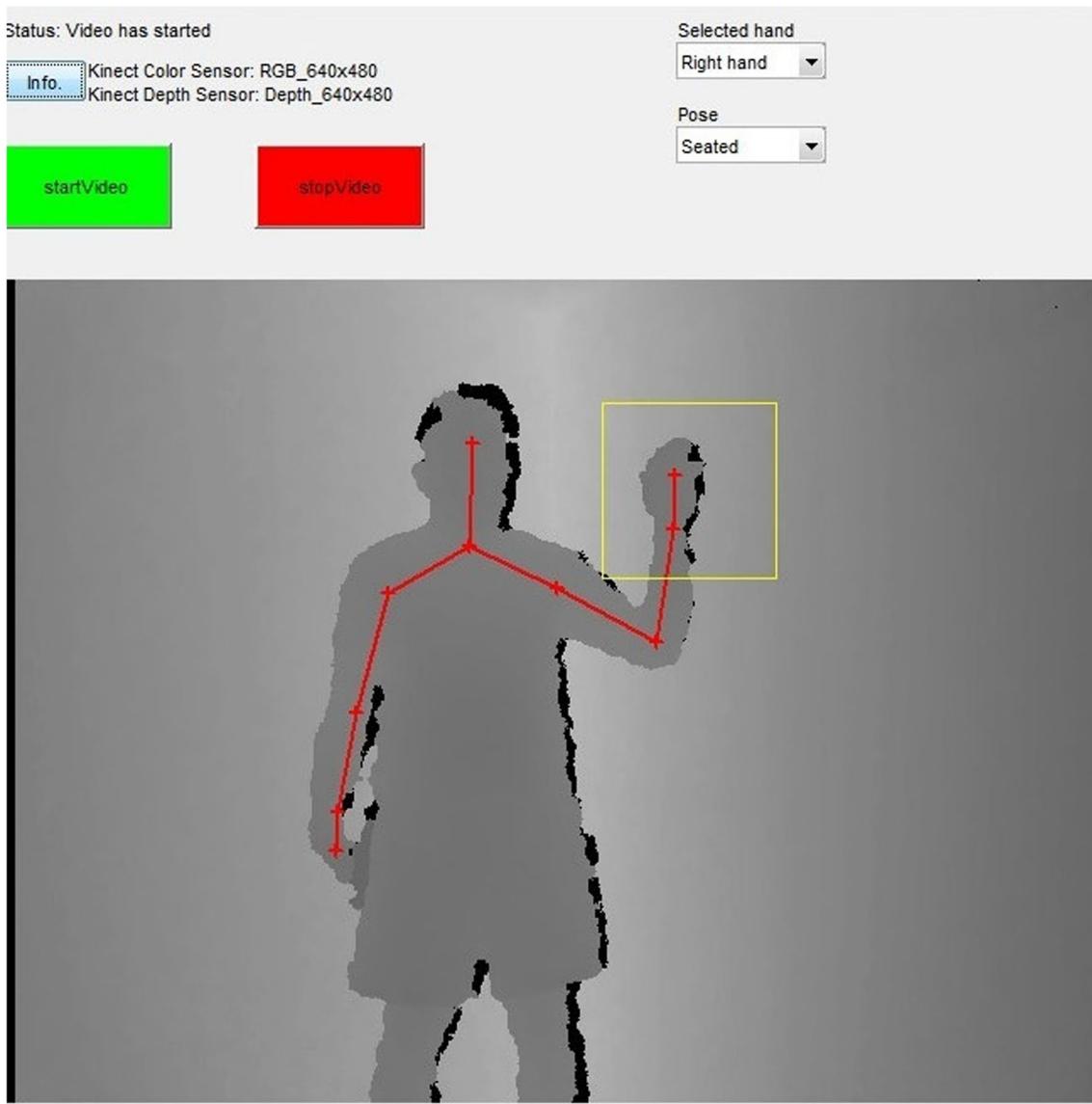
$$\begin{aligned} K_{pi}(k) &= K_{pi}(k-1) - \Delta K_{pi}(k) \\ K_{di}(k) &= K_{di}(k-1) - \Delta K_{di}(k) \end{aligned} \quad (35)$$

where the parameter changes are calculated as:

$$\begin{aligned} \Delta K_{pi}(k) &= \rho_{pi} \text{sign}(\lambda_{pi}(k))(p_{refi} - q_i) \\ \Delta K_{di}(k) &= \rho_{di} \text{sign}(\lambda_{di}(k))(p_{refi} - q_i) \end{aligned} \quad (36)$$

where  $\text{sign}(\cdot)$  is the signum function,  $\rho_{pi}$  and  $\rho_{di}$  are the adaptive gains, the normalized relative density is:

$$\begin{aligned} \lambda_{pi}(k) &= \frac{\gamma_{pi}(k)}{\sum_{j=1}^6 \gamma_{pi}(k)} \\ \lambda_{di}(k) &= \frac{\gamma_{di}(k)}{\sum_{j=1}^6 \gamma_{di}(k)} \end{aligned} \quad (37)$$



**Fig. 7** Interface to get the data

the relative density is updated as:

$$\gamma_{pi}(k) = \frac{1}{1 + \|x(k) - \mu_{pi}(k)\|^2 + \Lambda_{pi}(k) - \|\mu_{pi}(k)\|^2} \quad (38)$$

$$\gamma_{di}(k) = \frac{1}{1 + \|x(k) - \mu_{di}(k)\|^2 + \Lambda_{di}(k) - \|\mu_{di}(k)\|^2}$$

where  $\Lambda_{pi}(k)$  and  $\Lambda_{di}(k)$  denote the scalar of the data  $x(k)$ :

$$\Lambda_{pi}(k) = \frac{k-1}{k} \Lambda_{pi}(k-1) + \frac{1}{k} \|x_{pi}(k)\|^2 \quad (39)$$

$$\Lambda_{di}(k) = \frac{k-1}{k} \Lambda_{di}(k-1) + \frac{1}{k} \|x_{di}(k)\|^2$$

with the starting condition  $\Lambda_{pi}(1) = \|x_{pi}(1)\|^2$  and  $\Lambda_{di}(1) = \|x_{di}(1)\|^2$ . The update of the mean value  $\mu_{pi}(k)$  and  $\mu_{di}(k)$  is straightforward:

$$\mu_{pi}(k) = \frac{k-1}{k} \mu_{pi}(k-1) + \frac{1}{k} x_{pi}(k) \quad (40)$$

$$\mu_{di}(k) = \frac{k-1}{k} \mu_{di}(k-1) + \frac{1}{k} x_{di}(k)$$

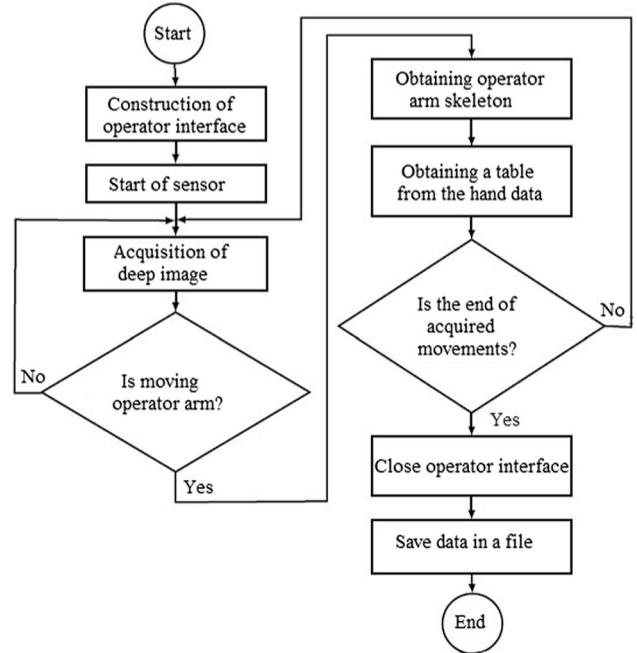
with the starting condition  $\mu_{pi}(1) = x_{pi}(1)$  and  $\mu_{di}(1) = x_{di}(1)$ .

### 3 The experimental results

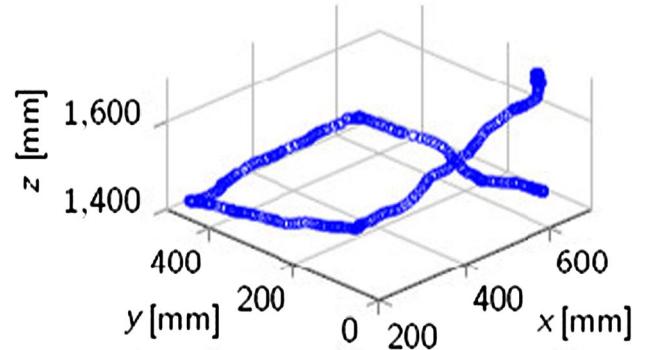
In this section, the experimental results of the sensor and control are exposed.

#### 3.1 Results of the sensor

The Kinect for Windows sensor is frequently employed as interface in this document. This device can make a three dimensional following of the operator body by employing a deep camera and a RGB camera. These cameras in combination with a automatic learning algorithm let to get positions of 20 parts in the operator body with a frequency of 30 Hz and with a resolution of  $640 \times 480$  pixels. The deep camera operates with the projection and reception of laser lights. The design of an algorithm to plan references is needed to avoid obstacles which could affect the generated reference. Since the learning of the operator hand movements must be performed by a program which updates itself, the least angle regression is employed in this document. Figure 6 shows the



**Fig. 8** Flow diagram to get data

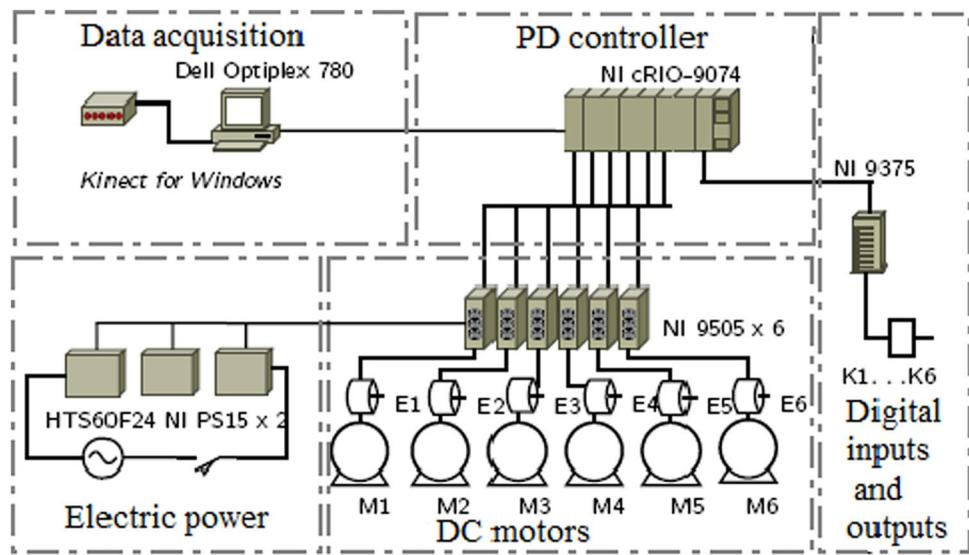


**Fig. 9** Data acquired by the sensor

**Table 1** Data acquired by the sensor

Time	Position x	Position y	Position z
0.0000	523.90873	345.24545	1345.11650
0.07157	525.29728	344.84822	1343.32275
0.13678	527.31872	345.91395	1341.09604
0.20471	528.66966	346.75002	1339.42235

**Fig. 10** Diagram of the evolving PD controller for the manipulator



position of the sensor to acquire the data. Figure 7 shows the graphic interface constructed employing image acquisition toolbox and Kinect for Windows SDK of Matlab. Figure 8 shows the flow diagram of the interface to get the data. Figure 9 shows the graphic acquired data during the first second. Table 1 shows data gotten during the first second where the time is in s and positions are in mm.

### 3.2 The results of the controller

Figure 10 shows the diagram of the evolving PD control for the manipulator. The acquisition data module is composed by a Kinect for Windows sensor which is connected to a Dell Optiplex 780 computer by a 2.0 USB port, this sensor sends depth images to the computer to be analyzed by a software. The computer screen shows a interface, as is seen in Fig. 7 which lets to get the operator hand movements to generate a desired reference which is saved in a text file. The controller developed in the NI cRIO-9074 reads the text file. The control software, programmed in a FPGA memory sends the desired references to the controllers NI 9505 which move the direct current motors of the six manipulator links, the controllers NI 9505 employ the digital inputs and outputs NI 9375 to move the manipulator links. The mentioned devices work due to the feed of the power modules HTS60F24 and NI PS15.

Figure 10 shows the diagram of the evolving PD controller for the manipulator. Figure 11 shows the manipulator used in experiments. Figure 12 shows the instrumentation employed by the evolving PD controller for experiments.

Figures 13, 14, 15, 16, 17, and 18 show results of the advised evolving PD controller for the manipulator links 1, 2, 3, 4, 5, and 6, respectively. In the mentioned figures, Theoretic corresponds to the movement of desired references, while Real corresponds to the movement of manipulator links. In all the cases, position, velocity, and acceleration movements are shown. The goal of the evolving PD controller is that the manipulator links follow the desired references.

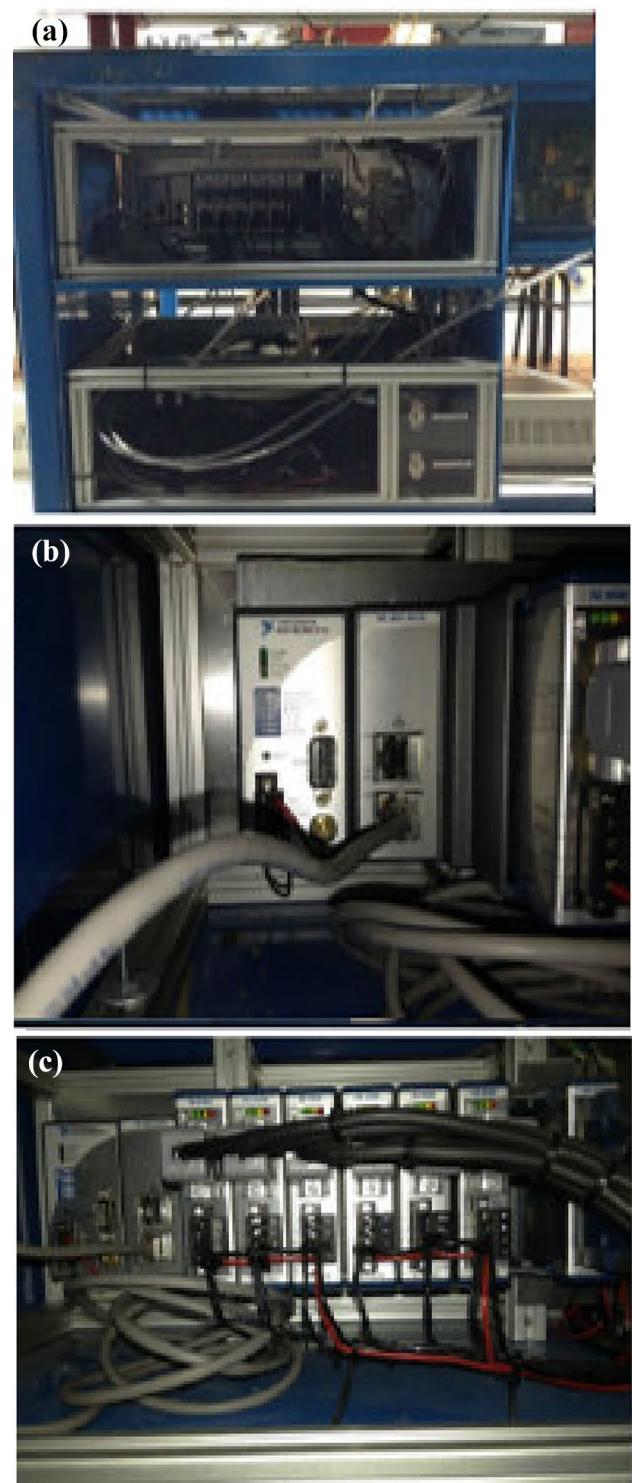
The movements of the manipulator links and desired references are similar, they are not equal due to some dynamics which are not considered in the dynamic model. However, they are almost equal. In addition, the data acquisition system has shown an acceptable performance in controlled conditions. The employ of a least angle regression and a fifth reference profile let to smooth the manipulator movements when a sense inversion is taken into account in each of the manipulator links, thus, the advised approach works satisfactorily.



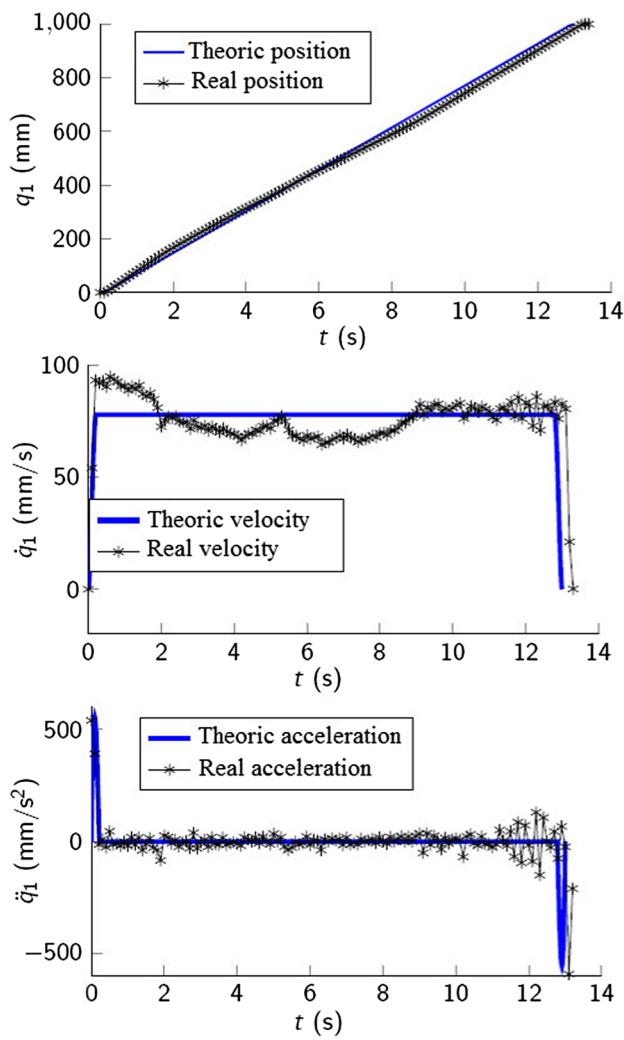
**Fig. 11** The manipulator

#### 4 Conclusions

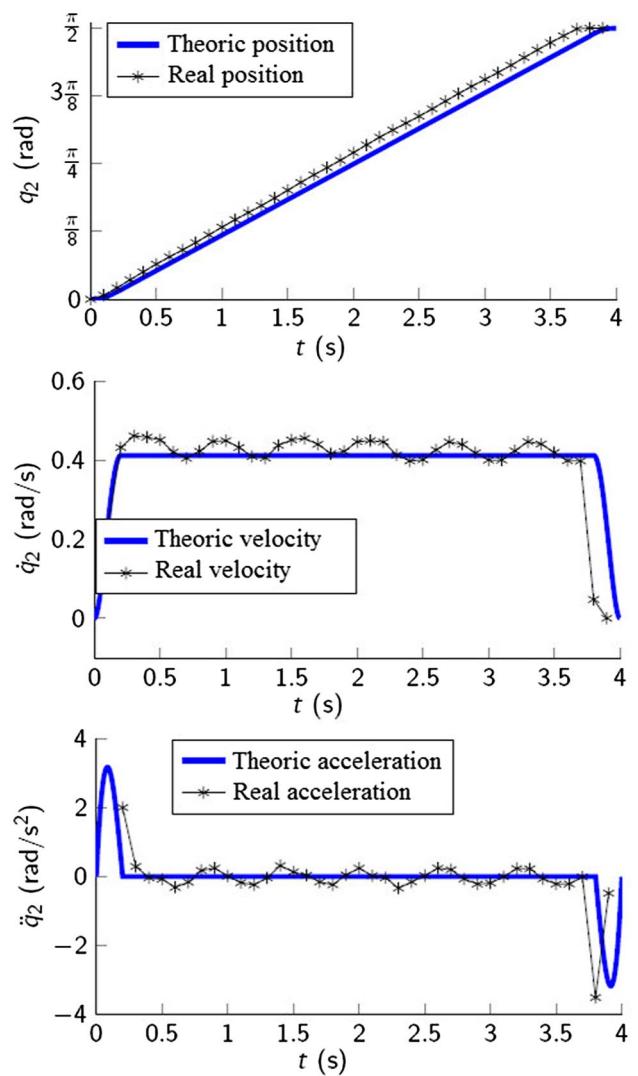
The target of this document is the automatic learning of movements for a manipulator from the demonstrations of an operator hand, avoiding the programming of each homework. The generalized learning model with the least angle regression reach their target, the learning of a smooth reference. The evolving PD control reaches its target, that the manipulator end effector follows the operator hand movements. The implementation of the experimental sensor and controller are tuned by try and error. The sensor and controller were sent to an embedded platform; consequently, this platform could be applied to any kind of industrial manipulator. The dynamic model consider ideal conditions which could be lost in any moment by for example a shock; then, this dynamic model could be improved by considering more exact mass centers and inertia matrices, and considering the friction in the manipulator links.



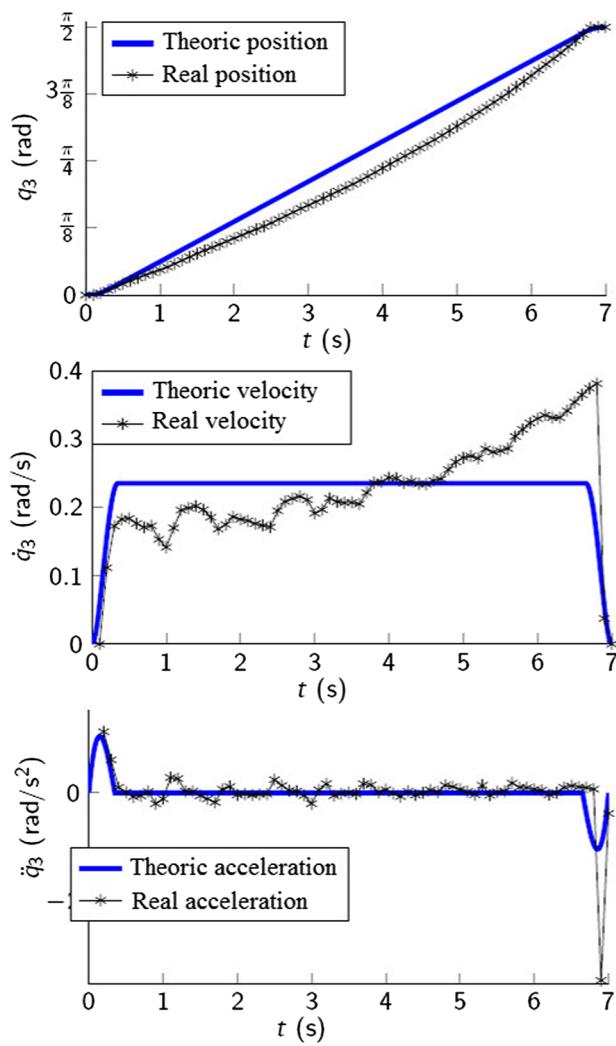
**Fig. 12** The instrumentation employed by the PD evolving controller



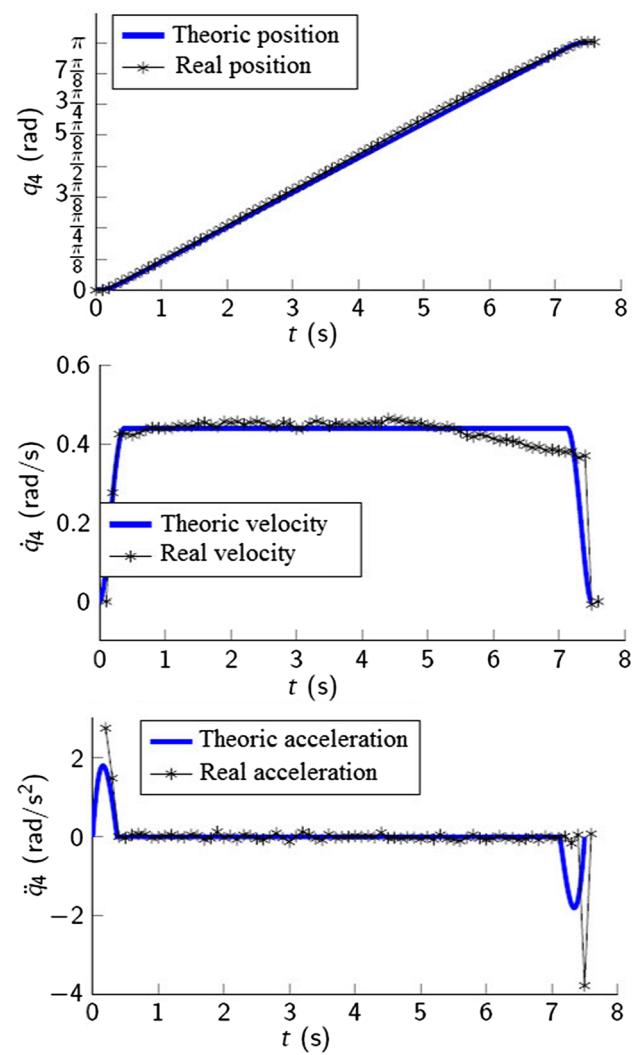
**Fig. 13** Results for the manipulator link 1



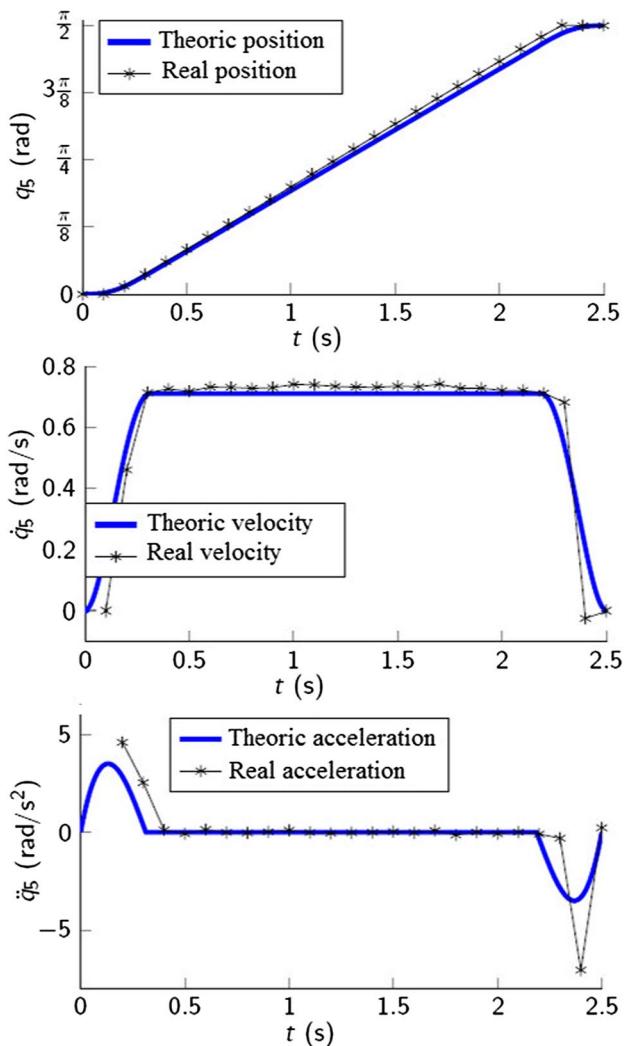
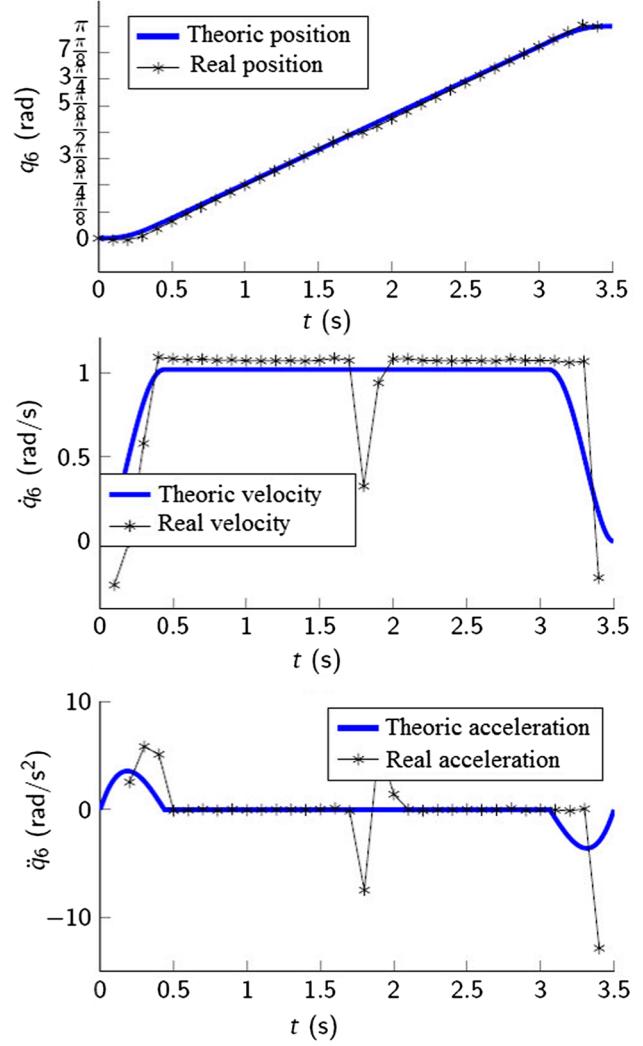
**Fig. 14** Results for the manipulator link 2



**Fig. 15** Results for the manipulator link 3



**Fig. 16** Results for the manipulator link 4

**Fig. 17** Results for the manipulator link 5**Fig. 18** Results for the manipulator link 6

**Acknowledgements** Authors thank the Editor in Chief, and reviewers for their valuable comments which let to improve this result. They thank the Instituto Politécnico Nacional, the Consejo Nacional de Ciencia y Tecnología, and Secretaría de Investigación y Posgrado for their support.

## References

- Abdallah ZS, Gaber MM, Srinivasan B, Krishnaswamy S (2016) AnyNovel: detection of novel concepts in evolving data streams—an application for activity recognition. *Evol Syst* 7(2):73–93
- Angelov P, Yager R (2012) A new type of simplified fuzzy rule-based system. *Int J Gen Syst* 41(2):163–185
- Angelov P, Skrjanc I, Blazic S (2015) A robust evolving cloud-based controller. In: Springer Handbook of Computational Intelligence, Springer, pp 1435–1449
- Angelov P, Sadeghi-Tehran P, Ramezani R (2011) An approach to automatic real-time novelty detection, object identification, and tracking in video streams based on recursive density estimation and evolving Takagi–Sugeno fuzzy systems. *Int J Intell Syst* 26:189–205
- Angelov P, Filev D, Kasabov N (2010) Evolving intelligent systems—methodology and applications. Wiley, New York
- Angelov P (2012) Autonomous learning systems: from data streams to knowledge in real-time. Wiley, New York
- Calderon CA, Ramirez C, Barros V, Punin G (2017) Design and deployment of grasp control system applied to robotic hand prosthesis. *IEEE Latin Am Trans* 15(2):181–188
- Fraley C, Hesterberg T (2009) Least angle regression and LASSO for large datasets. *Stat Anal Data Min* 1:251–259
- Garcia MA, Gallardo J, Rodriguez R, Alcaraz LA (2017) A new four-degrees-of-freedom parallel manipulator. *IEEE Latin Am Trans* 15(5):928–934
- Hernandez K, Bacca B, Posso B (2017) Multi-goal path planning autonomous system for picking up and delivery tasks in mobile robotics. *IEEE Latin Am Trans* 15(2):232–238
- Iliadis L, Margaritis K, Maglogiannis I (2016) Timely advances in evolving neural-based systems special issue. *Evol Syst* 8(1):1–2
- Kangin D, Angelov P, Iglesias JA (2016) Autonomously evolving classifier TEDAClass. *Inf Sci* 366:1–11
- Kasabov N (2007) Evolving connectionist systems: the knowledge engineering approach, 2nd edn. Springer, London
- Khamassi I, Sayed-Mouchaweh M, Hammami M, Ghedira K (2016) Discussion and review on evolving data streams and concept drift adapting. *Evol Syst*. <https://doi.org/10.1007/s12530-016-9168-2>
- Maciel L, Ballini R, Gomide F (2016) Evolving granular analytics for interval time series forecasting. *Granul Comput* 1:213–224
- Marques Silva A, Caminhos W, Lemos A, Gomide F (2014) A fast learning algorithm for evolving neo-fuzzy neuron. *Appl Soft Comput* 14:194–209
- Li X, Pan Y, Chen G, Yu H (2017) Multi-modal control scheme for rehabilitation robotic exoskeletons. *Int J Robot Res* 36(5):759–777
- Liu Z, Liu J (2017) Adaptive iterative learning boundary control of a flexible manipulator with guaranteed transient performance. *Asian J Control* 19(4):1–12
- Lughofer E (2013) On-line assurance of interpretability criteria in evolving fuzzy systems—achievements, new concepts and open issues. *Inf Sci* 251:22–46
- Lughofer E, Cernuda C, Kindermann S, Pratama M (2015) Generalized smart evolving fuzzy systems. *Evol Syst* 6:269–292
- Lughofer E (2016) Evolving fuzzy systems—fundamentals, reliability, interpretability and useability. In: Angelov P (ed) *Handbook of computational intelligence*. World Scientific, New York, pp 67–135
- Paramo Carranza LA, Meda Campaña JA, Rubio JJ, Tapia Herrera R, Curtidor Lopez AV, Grande Meza A, Cazares Ramirez I (2017) Discrete-time Kalman filter for Takagi–Sugeno fuzzy models. *Evol Syst* 8(3):211–219
- Peng Y, Liu J, He W (2017) Boundary control for a flexible inverted pendulum system based on a pde model. *Asian J Control* 19(2):1–10
- Pratama M, Lu J, Lughofer E, Zhang G, Anavatti S (2016) Scaffolding type-2 classifier for incremental learning under concept drifts. *Neurocomputing* 191:304–329
- Pratama M, Lu J, Anavatti S, Lughofer E, Lim C-P (2016) An incremental meta-cognitive-based scaffolding fuzzy neural network. *Neurocomputing* 171:89–105
- Precup R-E, Tomescu ML, Radac M-B, Petriu EM, Preitl S, Dragos C-A (2012) Iterative performance improvement of fuzzy control systems for three tank systems. *Expert Syst Appl* 39:8288–8299
- Precup R-E, Filip H-I, Radac M-B, Petriu EM, Preitl S, Dragos C-A (2014) Online identification of evolving Takagi–Sugeno–Kang fuzzy models for crane systems. *Appl Soft Comput* 24:1155–1163
- Rosado WMA, Valdes LGV, Ortega AB, Ascencio JR, Beltran CDG (2017) Passive rehabilitation exercises with an ankle rehabilitation prototype based in a robot parallel structure. *IEEE Latin Am Trans* 15(1):48–56
- Rubio JJ, Cruz P, Paramo LA, Meda JA, Mujica D, Ortigoza RS (2016) PID anti-vibration control of a robotic arm. *IEEE Latin Am Trans* 14(7):3144–3150
- Sa STL, Fernandes CC, Yanaguibashi EA, Barros RP, Burlamaqui AMF, Goncalves LMG (2017) Educaval: towards assessment of educational robotics softwares. *IEEE Latin Am Trans* 15(4):720–728
- Sayed-Mouchaweh M, Lughofer E (2012) Learning in non-stationary environments: methods and applications. Springer, New York
- Serrano ME, Godoy SA, Romoli S, Scaglia GJE (2018) A numerical approximation-based controller for mobile robots with velocity limitation. *Asian J Control* 20(1):1–13
- Venkatesan R, Er MJ, Dave M, Pratama M, Wu S (2016) A novel online multi-label classifier for high-speed streaming data applications. *Evol Syst* 8(4):303–315
- Zhang C, Sun T, Pan Y (2014) Neural network observer-based finite-time formation control of mobile robots. *Math Probl Eng* 2014:1–9