Multi-Inputs and Multi-Outputs equivalent model based on data driven controller for a robotic system

Josué Gómez [*] **Chidentree Treesatayapun** [*] **América Morales** [*]

[*] *Robotics and Advanced Manufacturing Programm CINVESTAV-Saltillo, Ramos Arizpe, 25903, México (e-mail:josuegomezcasas@gmail.com).*

Abstract: This paper proposes the control of a data driven model for an experimental robotic system. The components of the robotic system are a redundant robot and a motion capture system considered them as a Multi-Inputs and Mulit-Outputs system. The Pseudo Jacobian Matrix computes the equivalent model of the robotic system taking into account the input and output signals. Besides, we design the adaptive gains for a proportional controller using an artificial neuro-fuzzy network for the robot's end-effector control. The experimental results validate the proposed control scheme for a regulation control. We provided a Lyapunov analysis to guarantee convergence parameters of the controller.

Keywords: robotic system, equivalent model, pseudo Jacobian matrix, artificial neuro-fuzzy network, data driven control

1. INTRODUCTION

Currently the tendency to use the Data Driven Control (DDC) has captured the attention in fields as robotics, mechanical systems, and electronics devices due to the possibility to design controllers from system on-line information. The DDC allows to construct an equivalent model of the plant considering the system as a black box with a set of input and output signals. This approach reduces the parameters to obtain the system model, and as well omits the dependency of the classical modelling methods. An equivalent model based on data depends on the measured information from the input and output signals and through an estimation method relates the information to approximate the model.

Recently, the interest to work with equivalent model for robotic systems has increased using different estimation methods. A typical challenge to the estimation methods is the multi degrees of freedom of the robots, considering them as a Multi-Inputs and Multi-Outputs (MIMO) system. Some related works with equivalent model are reviewed as follow. The model for soft robots are complex to obtain by traditional methods, Li et al. (2018) introduced an equivalent model for a continuum robot, where the Jacobian matrix is estimated using a strong tracking Kalman filter algorithm through the input and output measured signals of the system. Chen et al. (2018) proposed a Jacobian matrix adaptation where the model of robots manipulators is considered unknown: the solution of this method transforms the robot modelling in an external, explicit and measurable structure of the input and output information of the system. Hou and Jin (2011) simulated the equivalent model for a 2 dof manipulator robot, this estimated model is based on the concept the Pseudo Jacobian Matrix (PJM), and the robot is a class of MIMO no-linear system. A controller with adaptive gains using artificial neural network has been proposed by Chiang and Chen (2017) to control a pneumatic robot. Also, Facundo et al. (2018) proposed a controller with adaptive gains based on a neuro-fuzzy network to control a cartesian robot.

⋆ CONACyT support acknowledgment.

The proposal of this work is to test in an experimental setup the estimated model by PJM algorithm to control a robotic system. In addition, we are proposing a proportional control with adaptive gains using a novel neuro-fuzzy network. The neuro-fuzzy network only requires the updating of one parameter in order to minimize the control error. Moreover, the stability analysis gurantees the convergence of the control parameters. The experimental robotic system is composed by an omnidirectional mobile-manipulator and a Motion Capture System (MOCAPS). The robot is from the academic platform KUKA youBot, and it has 3 dof in the omnidirectional mobile platform, and 5 dof in the robotic arm. This is a redundant robot considered as a MIMO system. The MOCAPS allows to locate objects in the scope space by markers attached above the robot and the objects, by this information it can be established a position error between the robot and the object.

The structure of this paper is: section 2 describes the equivalent model approach, section 3 describes the proposed robot control law, section 4 presents the experimental results, and section 5 gives the conclusions of this work.

2. EQUIVALENT MODEL

The position of the robot end-effector is defined in terms of the joint positions $\chi(t) = f(q(t))$ in continuous time. The velocity of the end-efector is $\dot{\chi}(t) = \frac{\partial \chi(t)}{\partial q(t)}\dot{q}(t)$, where $\frac{\partial \chi(t)}{\partial q(t)}$ is the Jacobian matrix, and $\dot{q}(t)$ is the velocity of the joints. For this paper we are working on discrete time domain, then the equation of the end-effector velocity is approximated by

$$\frac{\chi(k+1) - \chi(k)}{T_s} = J_A^*(k)\frac{q(k) - q(k-1)}{T_s}, \qquad (1)$$

where T_s is the sampling time. Generally speaking, for a first order kinematic control the Jacobian matrix represents the model of the robot. The real robotic Jacobian matrix is not fully known because of in the modelling methods exist uncertainties of the system. The classical modelling methods are based on the robot class and its physical characteristics, especially the robot model deals with parametric uncertainties and nonlinearities. Then, the modelling is unable to obtain the complete mathematical form of the robot, and by consequence the real Jacobian. The next expression represents the classical modelling approach:

$$J_A(k) = J_A^*(k) - \epsilon_a(k), \tag{2}$$

where $J_A(k)$ is the Jacobian matrix coming from a classical modelling method, $J_A^*(k)$ is the real Jacobian matrix, and $\epsilon_a(k)$ is the uncertainties in the model. On the other hand, the equivalent model coming from estimation methods are based on the input/output measurements information of the system, where the estimation algorithm intends to approximate the system by the on-line signals. The data driven estimation can not construct the complete system model, therefore there is an unknown information in the equivalent model. The representation of the estimation approach is

$$\hat{J}_A(k) = J_A^*(k) - \epsilon_b(k), \tag{3}$$

where $\hat{J}_A(k)$ is the Jacobian matrix coming from an estimation method, and $\epsilon_b(k)$ is the unknown information in the estimation. The position of robot's end-effector is in function of the joints' positions. We set as m the robot's end-effector $\chi(k+1)$ dof, and n the robot's joints $q(k)$ dof. The robotic Jacobian matrix is the direct relationship between the end-effector velocities (outputs) and the joints velocities (inputs), for this reason is to possible to apply data driven estimation method. Hou and Jin (2011) presented the Pseudo Jacobian Matrix (PJM) algorithm to compute an equivalent model. The PJM algorithm estimates the Jacobian matrix in one-step with low computational load and with on-line measured signals. The only expression to compute the model is the next:

$$\hat{J}_A(k) = \hat{J}_A(k-1) + \frac{\eta\left[\nu(k+1) - \hat{J}_A(k)\omega(k)\right]\omega^T(k)}{\mu + \parallel \omega(k) \parallel^2}, \tag{4}$$

where $\mu > 0$ is a weight parameter and $\eta \in \mathcal{D}(0,2]$ is a step parameter, $\nu(k+1)$ are the measured outputs signal as end-effector velocities, and $\omega(k)$ are the joint's velocity considered as inputs. The Jacobian matrix is approxmiated by the relationship of the input/output signals as:

$$\hat{J}_A(k) = \frac{\nu(k+1)}{\omega(k)}. \tag{5}$$

The next assumptions are established for this data driven estimation proposal:

Assumption 1: The output is observable, i.e $\nu(k+1) = \hat{J}_A(k)\omega(k)\ \forall k > 0$. From the measured output signals is possible to know the equivalent model of the system.

Assumption 2: The system is Lipschitz. A positive constant L must define the direct relationship between system input-output $\parallel \nu(k+1) \parallel \leq L \parallel \omega(k) \parallel$. This assumption imposes a bound for the change of the system output by the change of the system input.

Assumption 3: $\hat{J}_A(k)$ and $\omega(k)$ exist $\forall\ k$. To guarantee the control and estimation error converge to zero.

3. CONTROL LAW

The purpose of this control configuration is to work with an equivalent model for a position control in the task space of the robot. The inputs/outputs signals feed the estimation method to approximate the equivalent model. Due to the redundancy of the robot is computed a weighted pseudo inverse of the Jacobian matrix to generate the robot motion control. A neuro-fuzzy network is proposed to adapt the gains of a proportional control. The position error is defined as

$$e(k+1) = \chi(k+1) - \chi_d(k+1), \tag{6}$$

where $\chi(k+1)$ is the current position of the end-effector and $\chi_d(k+1)$ is the desired position. The position of the robot's end-effector is

$$\chi(k+1) = \chi(k) + \bar{J}_A(k)\omega(k), \tag{7}$$

where $\bar{J}_A(k) = J_A^*(k)T_s$ is an unknown function which includes the real Jacobian matrix and the sampling time. Then, from the equation (3) the real Jacobian is $J_A^*(k) = \hat{J}_A(k) + \epsilon_b(k)$, and the equation (7) becomes as

$$\chi(k+1) = \chi(k) + T_s\hat{J}_A(k)\omega(k) + T_s\epsilon_b(k)\omega(k), \tag{8}$$

substituting the current position equation (7) in the control error equation (6) we obtain

$$e(k+1) = \chi(k) + T_s\hat{J}_A(k)\omega(k) + T_s\epsilon_b(k)\omega(k) - \chi_d(k+1), \tag{9}$$

the pseudo inverse Jacobian matrix $\hat{J}_A^+(k)$ is

$$\hat{J}_A^+(k) = \hat{J}_A^T(k)\left[\ \hat{J}_A(k)\hat{J}_A^T(k) + \xi_b I\right]^{-1}, \tag{10}$$

where $\hat{J}_A^+(k)$ is build with estimation information of the PJM and $\xi_b = 0.1$. Now it is possible to calculate the control signals

$$\omega(k) = -\hat{J}_A^+(k)u(k), \tag{11}$$

where $u(k)$ is the controller, and the updated joints' positions are

$$q(k+1) = q(k) + \omega(k)T_s, \tag{12}$$

the signal in the equation (12) allows end-effector update to reach the desired position.

Now it is important to emphasize the differences between the classical model and data driven model.

i The classical model needs all the posible pysical parameters of the robot. In contrast, the data driven model only requires input and output signals.
ii The data driven model proposes values in the initial Jacobina matrix to include the criteria related to: the full rank of the Jacobian matrix and the holonomic constraint.

3.1 Neuro-fuzzy network

The controller is a proportional control with adaptive gains, a neuro-fuzzy network is used for the gains adaptation. The neuro-fuzzy network consists of the Fuzzy Rules Emulated Network (FREN) structure; their main characteristics are the online adaptation and the reasoning ability, proposed by Treesatayapun and Uatrongjit (2005). The architecture of the proposed artificial neuro-fuzzy network can be seen in the Fig. 1.

FREN is composed by four layers:

Layer 1: The error measurement $e(k)$ is the input of this layer which is sent to each node in the next layer directly.

Layer 2: This is called input membership function layer. Each node in this layer contains a membership function corresponding to one linguistic variable. The output at the ith node of this layer is calculated by $f(k)$ as

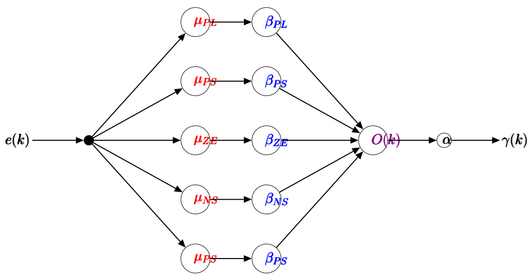


Fig. 1. Artificial Neural Network architecture.

$$f(k) = \mu_i(e(k)), \tag{13}$$

where μ_i denotes the membership function at the ith node ($i = 1, 2, ..., N$). The five linguistic variables are designed according to the physical characteristics of the robot axes. That means for x and y axes the robot can reach whichever position on the plane by the omnidirectional mobile platform. By the case of z axis the robotic arm can achieve positions until $0.7m$ high.

Fig. 2 shows the membership function for x and y axes, and Fig. 3 shows the five membership functions for z axis. Where the linguistic variables are: PL is positive large, PS is positive small, ZE is zero, NS is negative small, and NL is negative large. The IF-THEN rules can be established by the relationship between the position error and the adaptive gain

(1) IF $e(k)$ is Positive Large (PL), THEN $O(k)$ is Positive Large (PL),
(2) IF $e(k)$ is Positive Small (PS), THEN $O(k)$ is Positive Small (PS),
(3) IF $e(k)$ is Zero (ZE), THEN $O(k)$ is Zero (ZE),
(4) IF $e(k)$ is Negative Large (NL), THEN $O(k)$ is Negative Large (NL),
(5) IF $e(k)$ is Negative Small (NS), THEN $O(k)$ is Negative Small (NS).

where $O(k)$ is the output of FREN.

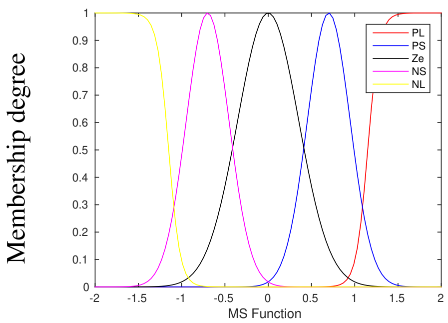


Fig. 2. The 5 membership functions designed in terms of $e_x(k)$ and $e_y(k)$.

Layer 3: This layer may be considered as a defuzzification step. It is called the linear consequence (LC) layer, where the parameters β_i remain constant with the values in Table 1:

Layer 4: This is the output of the artificial neural network and is calculated as

$$O(k) = \sum_{i=1}^{N} \beta_i \cdot \mu_i(e(k)), \tag{14}$$

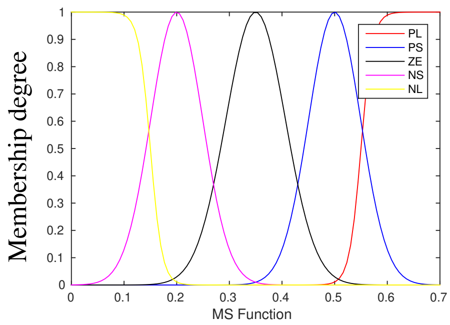


Fig. 3. The 5 membership functions designed in terms of $e_z(k)$.

Table 1. Value of β_i parameters.

Parameters	$\gamma_x(k)$ and $\gamma_y(k)$	$\gamma_z(k)$
β_{PL}	1.5	1.1
β_{NL}	0.7	0.8
β_{ZE}	0.5	0.7
β_{NS}	0.7	0.6
β_{NL}	1.5	0.5

where N represents the number of linguistic variables. The output of FREN $O(k)$ contains a positive values according to the membership functions taking values among 0 and 1, and β_i parameters in Table 1. Finally, the controller is

$$u(k) = O(k)\alpha e(k), \tag{15}$$

where $\gamma(k) = O(k)\alpha$, then the control can be rewritten as

$$u(k) = \gamma(k)e(k), \tag{16}$$

where α is a positive diagonal matrix with constant values and the $O(k)$ is the time-varying control gains coming from FREN structure.

3.2 Stability analysis

The Lyapunov function in terms of the position error is

$$V(k+1) = \frac{1}{2}e(k+1)e^T(k+1), \tag{17}$$

the change in the Lyapunov function is

$$\Delta V(k+1) = V(k+1) - V(k), \tag{18}$$

organizing the equation (18) in terms of $\Delta e(k+1)$ we obtain the next expression:

$$\Delta V(k+1) = \Delta e(k+1)\left[e(k) + \frac{1}{2}\Delta e(k+1)\right]^T, \tag{19}$$

and from the position error in the equation (9) and $\chi_d(k+1) = \chi_d(k) + \Delta\chi_d(k+1)$ the error equation becomes

$$\begin{aligned} e(k+1) =& \chi(k) - \chi_d(k) + T_s\hat{J}_A(k)\omega(k) \\ &+ T_s\epsilon_b(k)\omega(k) - \Delta\chi_d(k+1), \end{aligned} \tag{20}$$

and the $\Delta e(k+1)$ is:

$$\Delta e(k+1) = T_s\hat{J}_A(k)\omega(k) + T_s\epsilon_b(k)\omega(k), \tag{21}$$

In this paper the presented experiment is for regulation control, the end-effector reaches a fixed position in the space. Assuming this condition the change in the desired position $\Delta\chi_d(k+1) = 0$ and the control signal is $\omega(k) = -\hat{J}_A^{\dagger}(k)O(k)\alpha e(k)$.

(a) The end-effector home position. (b) The desired position in the x, y and z directions for regulation control (static object).

Fig. 4. The experimental scenario for reaching an object.

Therefore we can define $A_k = T_s \hat{J}_A(k) \hat{J}_A^{\dagger}(k) O(k) \alpha e(k)$ and $B_k = T_s \epsilon_b(k) \hat{J}_A^{\dagger}(k) O(k) \alpha e(k)$. Substituting the change in the error $\Delta e(k+1)$ in the change in the Lyapunov function $\Delta V(k+1)$ is obtained the next equation

$$\Delta V(k+1) = [-A_k - B_k] \left[e(k) - \frac{1}{2} [A_k + B_k] \right]^T, \quad (22)$$

according to Dietrich et al. (2015) the generalized pseudo inverse matrix $\hat{J}_A^{\dagger}(k)$ of a full row rank matrix $J_A^*(k) \in \mathbb{R}^{m \times n}$ and $\hat{J}_A(k) \in \mathbb{R}^{m \times n}$ with $m < n$ satisfy the next criterion $J_A^*(k) \hat{J}_A^{\dagger}(k) \approx I$ and $\hat{J}_A(k) \hat{J}_A^{\dagger}(k) \approx I$. In that sense we can define that $P_k^* = J_A^*(k) \hat{J}_A^{\dagger}(k)$, and $\hat{P}_k = \hat{J}_A(k) \hat{J}_A^{\dagger}(k)$ are positive definite matrices. Then, to probe the stability conditions the change in the Lyapunov function is

$$\Delta V(k+1) = -T_s P_k^* O(k) \alpha e(k) e^T(k) \left[I - \frac{1}{2} T_s P_k^* O(k) \alpha \right]^T \quad (23)$$

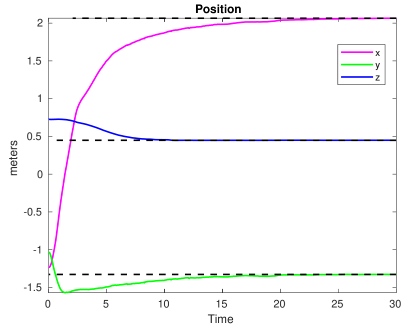
to fullfil the stability condition $\Delta V(k+1) < 0$, it is necessary to satisfy the next condition

$$\alpha < \frac{2}{T_s} \left[O_{Max}(k) \right]^{-1}, \quad (24)$$

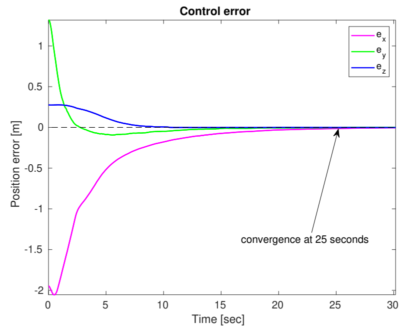
the values in the α matrix are bounded according to the maximum values in the output of ANN $O_{Max}(k)$.

4. RESULTS

The KUKA youBot is an academic platform with 3 dof for omnidirectional mobile platform and 5 dof for robotic arm i.e $n = 8$, see (http://www.youbot-store.com). The experiment is based on regulation control related with a static object. Fig. 4 shows the experimental scenario at the lab, where the robot starts from the home position, and it finishes to the object position $\chi_d(k+1)$, $20cm$ above the object to avoid collision among the end-effector and the object. The experimental setup is supported by a central computer which receives and sends information to the robot computer by means of Robotic Operating System (ROS, indigo version). Twelve cameras (MOCAPS) around the lab allow to know the robot position and object position online with with a resolution of 120 fps (https://optitrack.com). The central computer establishes communication with the robot-computer to send the control signals $\omega(k)$ to the robot actuators. As the robot moves, the cameras update the bodies position to close the feedback control and reduce to zero the position error. Mainly, the robot receives the control error $e(k)$ and the approximated end-effector velocities $\nu(k+1)$ (output-signals) to compute the control signals $\omega(k)$ (input-signals). The initial values of the estimated Jacobian matrix are selected close to the mathematical model values. The weight parameter



(a) End-effector position in the x, y and z directions during regulation control.



(b) Control position error in x, y and z.

Fig. 5. Experimental results of end-effector position and control error for regulation control.

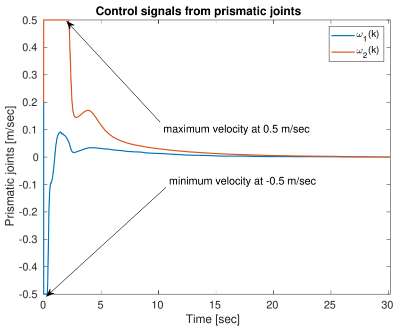
$\mu = 1$, and the step parameter $\eta = 0.1$, the initialization of the joint velocity vector is $\omega(0) = [0.1\ 0.1\ 0.2\ 0.2\ 0.2\ 0.2\ 0.2\ 0.2]^T$ for the prismatic joints $\omega_{\text{pris}}(0) = 0.1\frac{m}{s}$, and for the revolute joints $\omega_{\text{rev}}(0) = 0.2\frac{rad}{s}$. The values in the diagonal matrix α are determined according to the stability condition in the equation (24) as follow: $0 < \alpha_x < 2.17$, $0 < \alpha_y < 1.85$, and $0 < \alpha_z < 1.74$. The condition for the experimental setting are summarizing in the Table 2. The associated video can be found at https://www.youtube.com/watch?v=A4nsbtJ-eaM&t=150s.

Table 2. Parameters setting for robot control in the experimental setup.

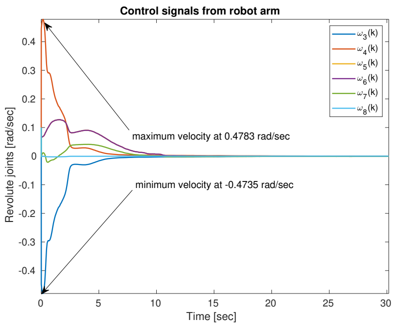
Parameters	Values	Remark
η	0.1	PJM algorithm (4)
μ	1	PJM algorithm (4)
α_x	0.5	stability condition (24)
α_y	0.5	stability condition (24)
α_z	1.5	stability condition (24)

In Fig. 5(a) is clearly observed that the convergence of the end-effector reaches the desired position above the static object at 25 seconds. The time of experiment convergence was designed to tune the gains under the limits of the robot actuators. Fig. 5(b) depicts the convergence of the control errors in the 3 axes, the proposed controller remedies the convergence of the control error successfully.

Fig. 6(a) shows the joint velocities as a control signal in the 2 prismatic joints and Fig. 6(b) shows the joint velocities as a control signal in the 5 revolute joints. It should be noted for revolute joints the control signals are smooth and under the bounded limits of the actuators $\omega_{\text{rev}}(k) \pm 06\frac{rad}{sec}$. By this form, it is guaranteed a satisfactory performance of the robot by the conditions selected in the stability analysis.



(a) Performance of the control signal for prismatic joints $\omega_1(k)$ and $\omega_2(k)$.



(b) Performance of the control signal for revolute joints: revolute joint of the mobile platform $\omega_3(k)$ and from $\omega_4(k)$ to $\omega_8(k)$ of the robotic arm.

Fig. 6. Control signals

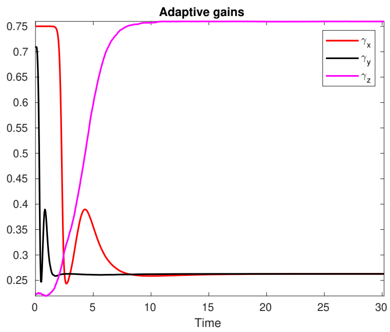


Fig. 7. Gains adaptation for $\gamma_x(k)$,$\gamma_y(k)$, and $\gamma_z(k)$.

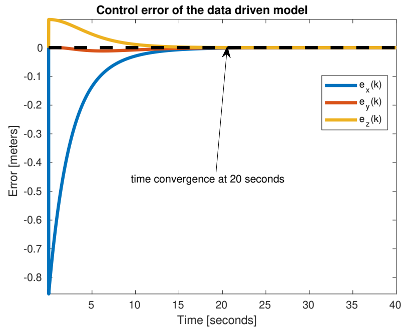
Fig. 7 depicts the adaptation of the gains $\gamma_x(k)$, $\gamma_y(k)$, and $\gamma_z(k)$. For the physical characteristics of the KUKA youBot the end-effector can reach easily the position on the x-y plane due to the omnidirectional platform, on the other hand by the position in z axis represents a challange due to by the control of the 5 dof in the robotic arm. For this reason we propose adaptive gains for the proportional controller based on the ANN, the membership function and the linear consequence parameters are designed according the characteristics of the robot.

Now the differences between the two methods using a proportional controller in simulations are illustrated. The classical model is obtained by the Denavit-Hartenberg convention for the KUKA youBot. The desired position is $\chi_d(k) = [1, 0, 0.55]^T$ and the conditions of the simulations are in Table 3.

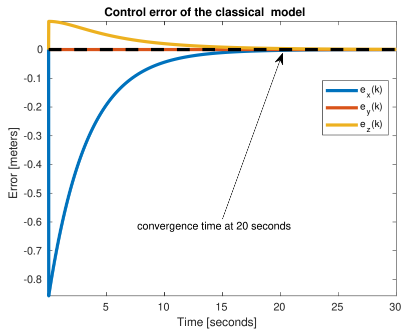
Fig.8(a) and 8(b) depict the position errors convergence, in both methods the convergence time occurs at 20 seconds. In the comparison of the prismatic joints signal, Fig.9(a) shows that

Table 3. Parameters setting for robot model and control in simualtions.

Parameters	Values	Remark
η	0.1	PJM algorithm
μ	1	PJM algorithm
ξ_a	0.1	damping factor (model)
ξ_b	0.01	damping factor (estimation)
K_p	0.35	control gain



(a) Control error $e(k)$ using data driven model.



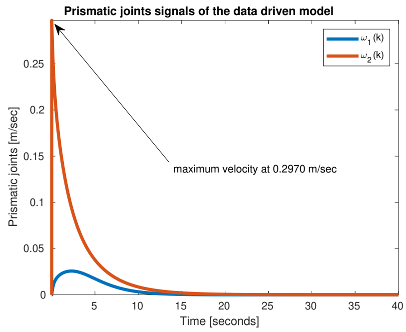
(b) Control error $e(k)$ using classical model.

Fig. 8. Comparison between the data driven model and the classical model for the position error.

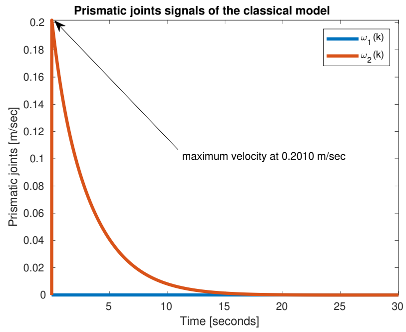
the maximum velocity in estimated method is 0.2970 m/s, while in Fig.9(b) the maximum velocity in classical method is 0.2010 m/sec. Nevertheless, both methods maintain the prismatic joints inside of the operation range. Fig.10(a) shows the revolute joints signals of the data driven model. While Fig.10(b) shows the revolute joints signals of the classical model, where the initial Jacobian matrix is known. By the case of the data driven model the maximum value of the angular velocity is 0.1236 rad/sec. Moreover, $\omega_4(k)$, $\omega_5(k)$ and $\omega_6(k)$are positive angular velocities and $\omega_7(k)$, $\omega_8(k)$ are negative angular velocities, as is seen in Fig.10(a). On the other hand, by the case of classical model in Fig.10(b) the maximum value of the angular velocity is 0.0813 rad/sec and the angular velocities $\omega_4(k)$ to $\omega_8(k)$ are positive values. In both cases the revolute joint of the mobile platform $\omega_3(k)$ is zero. The damping factor is higher in the classical model than the data driven model for the solution of the pseudo inverse matrix. Also, the data driven model reduces the number of instructions in the programing code.

5. CONCLUSIONS

We proposed a control for MIMO system in a robotic application. Mainly, the kinematic model is expressed by the estimation of the Jacobian matrix by PJM algorithm. The PJM algorithm only requires input/output signals to obtain an on-

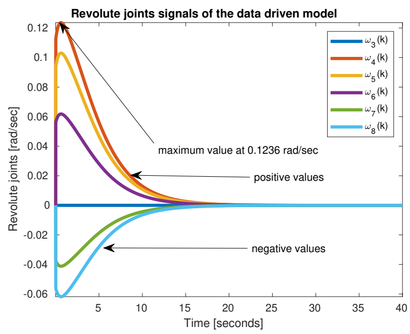


(a) Prismatic joints $\omega_1(k)$ and $\omega_2(k)$ using data driven model.

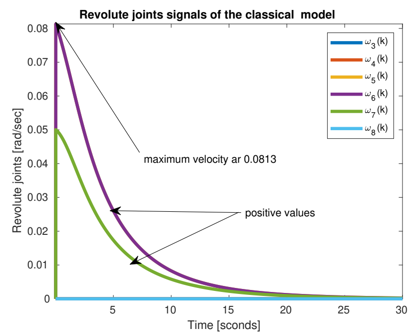


(b) Prismatic joints $\omega_1(k)$ and $\omega_2(k)$ using classical model.

Fig. 9. Comparison between the data driven model and the classical model for the prsimatic joints.



(a) Revolute joints $\omega_3(k)$ to $\omega_8(k)$ using data driven model.



(b) Revolute joints $\omega_3(k)$ to $\omega_8(k)$ using classical model.

Fig. 10. Comparison between the data driven model and the classical model for the revolute jonits.

line approximated model. The proposed proportional controller with adaptive gains is tuned by a novel neuro-fuzzy network. The main characteristic of this neuro-fuzzy network is the adaptation of only one parameter to guarantee the control error convergence. The DDC was tested in a experimental robotic system, which is composed by a redundant robot and a MOCAPS. It is important to emphasize that the MOCAPS can be replaced by any set of sensors that continuously read the position of the end-effector, for example an inertial measurement unit or a 3D electromagnetic tracking system. The experimental result validates the performance of data driven model and control. Moreover, the Lyapunov stability condition for the control law was determined. The equivalent model and the model error are included in the stability analysis. The approach presented omits the traditional modelling dependency based on robot structure, robot class, and physical parameters. As a future plan we will extend the work to cover the study of the estimated Jacobian matrix initialization, the robot's orientation control, the stability analysis of the estimated model and the tracking control experiment.

ACKNOWLEDGEMENTS

The authors would like to thank CONACyT for the financial support project through this work, which is called ”Análisis, control y sincronización de sistemas complejos con interconexiones dinámicas y acoplamientos flexibles, A1-S-26123”. The first author thanks CONACyT for his PhD scholarship.

REFERENCES

Chen, D., Zhang, Y., and Li, S. (2018). Tracking control of robot manipulators with unknown models: A jacobian-matrix-adaption method. *IEEE Transactions on Industrial Informatics*, 14(7), 3044–3053.

Chiang, C.J. and Chen, Y.C. (2017). Neural network fuzzy sliding mode control of pneumatic muscle actuators. *Engineering Applications of Artificial Intelligence*, 65, 68–86.

Dietrich, A., Ott, C., and Albu-Schäffer, A. (2015). An overview of null space projections for redundant, torque-controlled robots. *The International Journal of Robotics Research*, 34(11), 1385–1400.

Facundo, L., Gómez, J., Treesatayapun, C., Morales, A., and Baltazar, A. (2018). Adaptive control with sliding mode on a double fuzzy rule emulated network structure. *IFAC-PapersOnLine*, 51(13), 609–614.

Hou, Z. and Jin, S. (2011). Data-driven model-free adaptive control for a class of mimo nonlinear discrete-time systems. *IEEE Transactions on Neural Networks*, 22(12), 2173–2188.

Li, M., Kang, R., Branson, D.T., and Dai, J.S. (2018). Model-free control for continuum robots based on an adaptive kalman filter. *IEEE/ASME Transactions on Mechatronics*, 23(1), 286–297.

Treesatayapun, C. and Uatrongjit, S. (2005). Adaptive controller with fuzzy rules emulated structure and its applications. *Engineering Applications of Artificial Intelligence*, 18(5), 603–615.