Digitalization and Management Innovation
A.J. Tallón-Ballesteros and P. Santana-Morales (Eds.)

doi:10.3233/FAIA230066

Quaternion Modeling of a Delta Planar Robot and Training of an Enhanced Multilayer Neural Network to Solve the Inverse Kinematic Problem

Eusebio JIMÉNEZ-LÓPEZ[a,1], Renée Nicole ESPINOZA-MIRANDA[b], Aldo Augusto LÓPEZ-MARTÍNEZ[c], Eduardo NÚÑEZ- PÉREZ[b], Pablo Alberto LIMÓN-LEYVA[d], Kevin FIERRO-RUIZ[b] and Francisco CUENCA-JIMÉNEZ[e]
[a] *Universidad Tecnológica del Sur de Sonora-ULSA Noroeste, México*
[b] *Universidad La Salle Noroeste, México*
[c] *Centro de Ingeniería y Desarrollo Industrial (CIDESI), México*
[d] *Instituto Tecnológico de Sonora, México*
[e] *Universidad Nacional Autónoma de México, México*

Abstract. This paper presents the modeling of the inverse kinematic problem related to the motions of a delta planar robot using the algebra of unitary Quaternions. The mathematical model resulting from the inverse kinematic analysis has an associated system of 8 nonlinear algebraic equations with 8 polynomial unknowns. The Newton-Raphson method was used to solve the mathematical model of the robot. Subsequently, using the inverse model of the robot, a database was constructed that relates the Cartesian coordinates of the end effector to the angles and axes of the rotations of the links. This database was used to train a multilayer neural network in order to have an equivalent model of the inverse problem. A series of experiments were performed to obtain an improved network configuration by varying four training parameters. The results obtained show that the improved trained network can be used to solve the inverse problem of the studied robot.

Keywords. Robot, Neural networks, Newton-Raphson, kinematics

1. Introduction

Robot modeling is performed using various mathematical methods and tools, such as homogeneous matrices [1] and Quaternions [2]. In general, kinematic modeling of robots generates systems of nonlinear equations, so one of the most common numerical solution techniques applied to solve such models is the Newton-Raphson method [3]. Machine learning is an Artificial Intelligence (AI) technique that is currently being used to solve robot kinematic models as an alternative to Newton-Raphson, such as neural networks [4]. Other AI algorithms have been used for applications in robot kinematics, for example, in [5] a model of a Neuro-Fuzzy inference system was built to predict the position of the

[1] Corresponding Author, Eusebio JIMÉNEZ-LÓPEZ, Researcher assigned to the Research and Development Department, Universidad Tecnológica del Sur de Sonora, Cd. Obregón, México; E-mail: ejimenezl@msn.com.

end-effector of a parallel RRR-type robot within the workspace and tuned with a particle swarm optimization and a genetic algorithm. In [6], the direct kinematic problem of a Stewart platform was solved using soft computing and, subsequently, a particle swarm optimization method was used and a multilayer neural network was trained to solve the forward kinematics problem.

In this paper, the inverse kinematic modeling of a 2 GDL delta planar robot using unitary Quaternions [7] is presented. The mathematical model obtained is solved using the Newton-Raphson method and, subsequently, a neural network is trained and the results are compared. 81 experiments were performed by varying some training parameters of the neural network to obtain an improved network architecture.

2. Quaternion Algebra

Let the set $\Re^4$, on which the binary operations are defined $\oplus:\Re^4\times\Re^4\rightarrow\Re^4$ and $*: \Re^4\times\Re^4\rightarrow\Re^4$ be expressed [7]:

i) $(a,b,c,d) \oplus (\alpha,\beta,\gamma,\delta) = (a+\alpha,\ b+\beta\ ,\ c+\gamma\ ,\ d+\delta)$ (1)
ii) $(a,b,c,d) * (\alpha,\beta,\gamma,\delta) = (a\alpha - b\beta - c\gamma - d\delta,\ a\beta + b\alpha + c\delta - d\gamma,\ a\gamma - b\delta + c\alpha + d\beta,\ a\delta + b\gamma - c\beta + d\alpha\ ),\ \forall\ (a,b,c,d), (\alpha,\beta,\gamma,\delta) \in \Re^4$

The pairs $(\Re^4,\oplus)$ and $(\Re^4,*)$ form a commutative additive group and a non-commutative multiplicative group, respectively. The triple $(\Re^4,\oplus,*)$ forms a non-commutative field. the operation $\bullet:\Re\times\Re^4\rightarrow\Re^4$ is defined by $\alpha\bullet(a,b,c,d) = (\alpha a,\alpha b,\alpha c,\alpha d)$, $\forall(a,b,c,d)\in\Re^4$, $\alpha\in\Re$. It is a scalar product on $\Re^4$, so the term $(\Re^4,\oplus,\bullet)$ is a real vector space. The transformation $<\bullet,\bullet>:\Re^4\times \Re^4\rightarrow\Re$, it is an inner product in $\Re^4$. It is an inner product in $\Re^4$ and, therefore, the structure $\mathbf{Q}=(\Re^4,\oplus,*,\bullet,<\bullet,\bullet>)$ is a real vector space with inner product, and the norm associated with this inner product is the following: $||\mathbf{p}|| = <\mathbf{p},\mathbf{p}>^{1/2} = (p_0^2+p_1^2+p_2^2+p_3^2)^{½}$. For which the structure $\mathbf{Q}=(\Re^4,\oplus,*,\bullet, |\bullet|)$ is a normed space which will be called the vector space of quaternions, and its elements will be called quaternions [7]. In a similar way to the algebra of complex numbers, a conjugate quaternion can be defined as follows: $\bar{\mathbf{p}} = \left(p_0, -p_1, -p_2, -p_3\right)$.

2.1. Parametric representation of rotations of a rigid body

Let $\rho(p,\bullet):Q\rightarrow Q$, $p\in Q$ be a linear transformation defined by:

$$\rho(\mathbf{p},\mathbf{q}) = \mathbf{p}*\mathbf{q}*\mathbf{p}^{-1} = \frac{1}{||\mathbf{p}||^2} \bullet (\mathbf{p}*\mathbf{q}*\bar{\mathbf{p}}),\ \forall \mathbf{p},\mathbf{q} \in \mathbf{Q} \tag{2}$$

This function is a rotation that preserves the inner product, the norm, and the angle [7]. The transformation $\rho(\mathbf{p},\bullet):\mathbf{Q}\rightarrow\mathbf{Q}$, is linear and orthogonal. The geometrical relations between the Quaternion components $\mathbf{p}\in\Re^4$ with $||\mathbf{p}||=1$ are as follows:

$$p_0 = \cos(\theta/2),\quad \mathbf{p}_v = \pm \sin(\theta/2)\mathbf{w} \tag{3}$$

3. Kinematic model of the delta planar robot

The multibody to be modeled is a parallel delta planar robot (Figure 1). The objective of the modeling is to relate the origin of coordinates "O" where a fixed inertial base is located, with the point "pot" by means of the robot configuration. The explicit modeling of this robot is performed in two configurations, an initial and a final one [8]. In this paper, the final modeling will be presented.

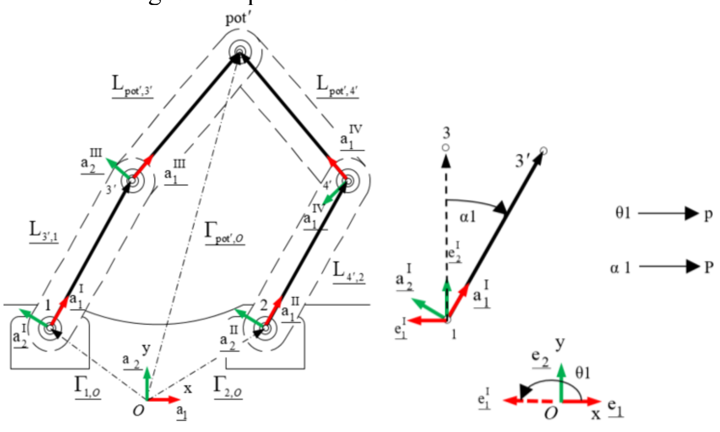


Figure 1. Vectorization of the final robot configuration: a) Position vectors and local bases and b) Sequence of rotations of the inertial base over the local bases [8].

According to Figure 1, the position of the "pot" point measured from the origin of coordinates can be determined by means of the following closed-loop equations:

$$\begin{aligned}\underline{\Gamma_{pot',O}} &= \underline{\Gamma_{1,O}} \oplus \underline{L_{3',1}} \oplus \underline{L_{pot',3'}}\\ \underline{\Gamma_{pot',O}} &= \underline{\Gamma_{2,O}} \oplus \underline{L_{4',2}} \oplus \underline{L_{pot',4'}}\end{aligned} \tag{4}$$

Each vector defined in R^4 associated with the links can be written in terms of a multiplication between a scalar and a unit vector. That is:

$$\begin{aligned}\underline{\Gamma_{pot',O}} &= \underline{\Gamma_{1,O}} \oplus L_{3',1} \bullet \underline{a_1^{I}} \oplus L_{pot',3'} \bullet \underline{a_1^{III}}\\ \underline{\Gamma_{pot',O}} &= \underline{\Gamma_{2,O}} \oplus L_{4',2} \bullet \underline{a_1^{II}} \oplus L_{pot',4'} \bullet \underline{a_1^{IV}}\end{aligned} \tag{5}$$

The local bases shown in Figure 1.a) are defined in the deformed or final configuration and are representations of the rigid rotations of the inertial base located at point O (Figure 1). These rotations are modeled below:

$$\begin{aligned}\underline{a_1^{I}} &= \rho\left(\mathbf{P}, \underline{e_1^{I}}\right) = \rho\left(\mathbf{P}, \rho\left(p, \underline{e_1}\right)\right) = \mathbf{P} * \mathbf{p} * \underline{e_1} * \bar{\mathbf{p}} * \bar{\mathbf{P}}\\ \underline{a_1^{I}} &= \rho\left(\mathbf{R}, \underline{e_1^{I}}\right) = \rho\left(\mathbf{R}, \rho\left(r, \underline{e_1}\right)\right) = \mathbf{R} * \mathbf{r} * \underline{e_1} * \bar{\mathbf{r}} * \bar{\mathbf{R}},\\ \underline{a_1^{I}} &= \rho\left(\mathbf{Q}, \underline{e_1^{I}}\right) = \rho\left(\mathbf{Q}, \rho\left(q, \underline{e_1}\right)\right) = \mathbf{Q} * \mathbf{q} * \underline{e_1} * \bar{\mathbf{q}} * \bar{\mathbf{Q}},\\ \underline{a_1^{I}} &= \rho\left(\mathbf{S}, \underline{e_1^{I}}\right) = \rho\left(\mathbf{S}, \rho\left(s, \underline{e_1}\right)\right) = \mathbf{S} * \mathbf{s} * \underline{e_1} * \bar{\mathbf{s}} * \bar{\mathbf{S}},\end{aligned} \tag{6}$$

Here, $\mathbf{P}, \mathbf{Q}, \mathbf{R}, \mathbf{S} \in \Re^4$ are the Quaternions related to the final or deformed configuration and $\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s} \in \Re^4$are Quaternions associated with the initial configuration. Figure 1.b) shows graphically the sequence of rotations of the local bases. The angles θ_i are associated with the Quaternions of the reference configuration and the angles α_i are associated with the Quaternions of the final configuration. The loop equations in terms of Quaternions are expressed as follows:

$$\begin{aligned} \underline{\Gamma_{pot',0}} &= \underline{\Gamma_{1,0}} \oplus L_{3',1} \bullet \left\{\mathbf{P}*\mathbf{p}*\underline{e_1}*\bar{\mathbf{p}}*\bar{\mathbf{P}}\right\} \oplus L_{pot',3'} \bullet \left\{\mathbf{R}*\mathbf{r}*\underline{e_1}*\bar{\mathbf{r}}*\bar{\mathbf{R}}\right\} \\ \underline{\Gamma_{pot',0}} &= \underline{\Gamma_{2,0}} \oplus L_{4',2} \bullet \left\{\mathbf{Q}*\mathbf{q}*\underline{e_1}*\bar{\mathbf{q}}*\bar{\mathbf{Q}}\right\} \oplus L_{pot',4'} \bullet \left\{\mathbf{S}*\mathbf{s}*\underline{e_1}*\bar{\mathbf{s}}*\bar{\mathbf{S}}\right\} \end{aligned} \tag{7}$$

In addition, it is necessary to: $\|\mathbf{P}\|=\|\mathbf{Q}\|=\|\mathbf{R}\|=\|\mathbf{S}\|=1$. The relationships between Quaternions $P, Q, R, S \in R^4$, angles $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in R$ and rotation axes are:

$$\begin{aligned} &\mathbf{P} = (P_0, 0,0, P_1); P_0 = \cos\frac{\alpha_1}{2}; P_0 = \pm\sin\frac{\alpha_1}{2}; \mathbf{Q} = (Q_0, 0,0, Q_1); Q_0 = \cos\frac{\alpha_2}{2}; Q_1 = \pm\sin\frac{\alpha_2}{2} \\ &\mathbf{R} = (R_0, 0,0, R_1); R_0 = \cos\frac{\alpha_3}{2}; R_1 = \pm\sin\frac{\alpha_3}{2}; \mathbf{S} = (S_0, 0,0, S_1); S_0 = \cos\frac{\alpha_4}{2}; S_1 = \pm\sin\frac{\alpha_4}{2} \end{aligned} \tag{8}$$

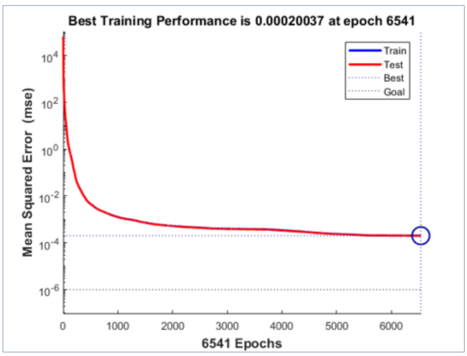
The inverse kinematic problem is formulated with the equations (8) and the expressions of the unit norm which are nonlinear and with the parameters of the Quaternions. Therefore, there are 8 nonlinear scalar equations with 8 unknowns (P_0, P_1, Q_0, Q_1, R_0, R_1, S_0, S_1). The input data are the coordinates of the "pot" point and the outputs are the angles and axes of each joint. The known data are the dimensions of the links and the quaternions $\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s} \in \Re^4$, as well as the coordinates of point "1" and "2" (Figure 1). The explicit model in scalar equations is generated with equations (1) and the unit norms of the quaternions.

4. Training of a neural network

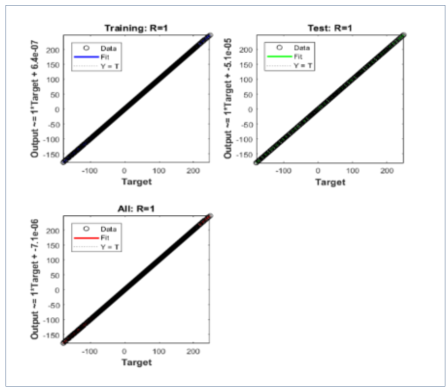
In this section we will describe the training of a multilayer neural network which is fed by a database generated by solving the kinematic problem related to the delta planar robot using the Newton-Raphson method. Subsequently, experimentation with the representative network will be performed by varying four training parameters and using a filtering method to generate an improved network.

The methodology used to carry out the training was: 1) Selection of a region in the plane of the robot's working area, 2) Calculation of the inverse kinematic problem using the Newton-Rapshon method considering a list of selected points, and the elimination of mirror configurations, 3) Generation of two databases: one for training and one for validation, 4) Normalization of the input data, 5) Design of the neural network topology, 6) Configuration of the neural network, 7) Training of the network, 8) Validation of the network and comparison with the Newton-Raphson, 8) Conformation of a reference network, 9) Selection of parameters to be modified for network improvement, 9) Running the experiment and 10) Filtering and selection of the improved network.

To carry out the training of the multilayer network and the experimentation to locate an improved network, Matlab software was used. To generate the database 10,000 points were considered, the network configuration was 2 layers with 3 neurons each, the activation function considered was Hyperbolic tangent sigmoid and the performance function was the mean square error (mse). The Bayesian Regulation training algorithm was selected to train the network and the statistical method for validation was linear regression. A total of 1000 epochs and a performance of 0.001 were taken into account. Figure 2 shows a plot of neural performance and the training state of the network. The representative network configuration was [2, 3, 1000, 0.001], i.e., [layers, neurons, epochs, throughput].



a)



b)

Figure 2. a) Neural performance and b) Training status

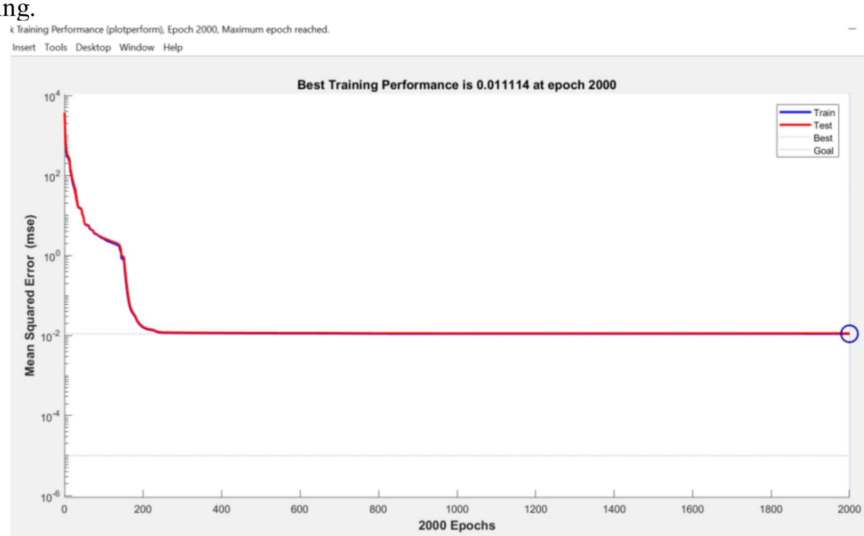
Table 1 shows the validation of the network and the error obtained between the trained network and the data obtained from the Newton-Raphson method.

Table 1. Neural network validation.

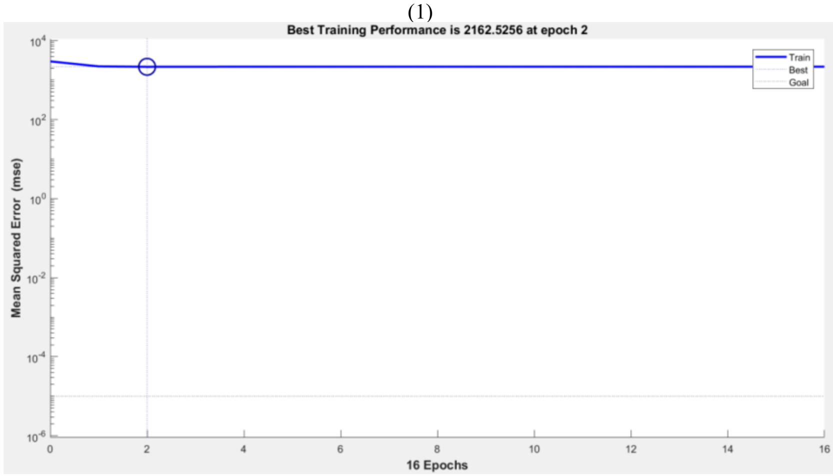
Neuronal Network									
Input X	Input Y	P0	P1	Q0	Q1	R0	R1	S0	S1
7.37	78.899	-0.1951515	0.98077311	0.89872536	-0.4385119	0.31058701	0.95054495	-0.9128551	-0.4082838
35.68	84.198	-0.0087858	0.9999614	0.90693017	-0.421281	0.56072273	0.82800363	-0.9684203	-0.2493232
32.97	120.0999	-0.1061099	0.99435441	0.98562882	-0.1689255	0.57365692	0.81909568	-0.9999799	0.00633318
Newton Raphson									
Input X	Input Y	P0	P1	Q0	Q1	R0	R1	S0	S1
7.37	78.899	-0.1951515	0.98077311	0.89872536	-0.4385119	0.31058701	0.95054495	-0.9128551	-0.4082838
35.68	84.198	-0.0087858	0.9999614	0.90693017	-0.421281	0.56072273	0.82800363	-0.9684203	-0.2493232
32.97	120.0999	-0.1061099	0.99435441	0.98562882	-0.1689255	0.57365692	0.81909568	-0.9999799	0.00633318
Input X	Input Y	Error							
7.37	78.899	3.75E-12	1.36E-12	4.63E-12	4.79E-12	1.59E-12	4.42E-12	7.79E-13	1.73E-12

35.68	84.198	2.48E-12	2.04E-12	-1.45E-12	4.25E-12	4.96E-12	7.79E-13	5.41E-14	-4.49E-12
32.97	120.0999	4.55E-13	3.98E-12	-3.89E-12	-3.77E-12	-1.36E-12	-3.43E-12	1.91E-12	3.63E-12

In order to obtain an improved neural network, experimentation was carried out taking into account and varying the number of layers (2, 3, 4), the number of neurons per layer (3, 4, 5), the number of epochs (1000, 2000, 3000) and the yields (0.001, 0.0001, 0.00001). The total number of runs was 81. The information generated was filtered by the following concepts: Yield, epochs, gradient, and test, and three types of networks were selected: best, worst, and fuzzy. Figure 3 shows the performances of the best [2, 4, 2000, 0.00001] and worst [3, 4, 2000, 0.00001] networks that were generated during training.



(1)



(2)

Figure 3. Best training Performance: 1) Red [2, 4, 2000, 0.00001] y 2) Red [3, 4, 2000, 0.00001]

5. Conclusion

A topology of an improved network obtained by modifying four parameters and filtering variables was found. Such a network was obtained from a set of 81 experiments and whose nature was that it learned to compute the inverse kinematics of a delta planar robot. Under the filtering parameters, the network topology [2, 4, 2000, 0.00001] was the best because it presented a better fit. The experimentation was performed after obtaining a representative network, which in this case was: [2, 3, 1000, 0.001]. The mathematical model associated with the robot developed with Quaternions generated a system of eight nonlinear algebraic equations and eight unknowns, and was the basis, together with the Newton-Raphson method to generate the database used to train and validate the representative and improved network.

References

[1] Ali MH, Kuralbay Y, Aitmaganbet A, Kamal MAS. Design of a 6-DOF robot manipulator for 3D printed construction. Mater. Today Proc. 2022, 49, 1462–1468

[2] Li L, Han J, Li X, Guo B, Wang X, Du G. A novel end-effector upper limb rehabilitation robot: Kinematics modeling based on dual quaternion and low-speed spiral motion tracking control. International Journal of Advanced Robotic Systems. 2022;19(4)

[3] Cruz, S.R., García, M. & Olais, J.M. NAO robots as context to teach numerical methods. Int J Interact Des Manuf, 2022, 16, 1337–1356.

[4] Jiménez E, de la Mora DS, Reyes LA, de la Mora RS, Melendez J, López A.A. Modeling of inverse kinematic of 3-DOF robot, using unit quaternions and artificial neural network. Robotica, 2021, 39, 1230–1250.

[5] Sayed AS, Azar AT, Ibrahim ZF, Ibrahim HA, Mohamed, NA, Ammar, HH. Deep Learning Based Kinematic Modeling of 3-RRR Parallel Manipulator. In: Hassanien, AE., Azar, A., Gaber, T., Oliva, D., Tolba, F. (eds) Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020). AICV 2020. Advances in Intelligent Systems and Computing, 2020, vol 1153. Springer, Cham. p. 308-321

[6] Chauhan, D.K.S.; Vundavilli, P.R. Forward Kinematics of the Stewart Parallel Manipulator Using Machine Learning. Int. J. Comput. Methods. 2022, 214, 2009.

[7] Reyes, L., “Quaternions: Une Representation Parametrique Systematique Des Rotation Finies. Partie I: Le Cadre Theorique,” In: Rapport de Recherche INRIA Rocquencourt France. 1990.

[8] Jiménez, E., Servín, D., Servín, R., Ochoa, F. J., Acosta, M., & Luna, G. Modeling in two configurations of a 5R 2-DoF planar parallel mechanism and solution to the inverse kinematic modeling using artificial neural network. IEEE ACCES. 2021, 9, p. 68583-68594.