Scope of Linear Genetic Programming in the Search for Kinematics Models for a 3 DOF SCARA Robot Subjected to Path Following

Humberto Velasco Arellano, Martín Montes Rivera

Universidad Politécnica de Aguascalientes,
Maestría en Ciencias en Ingeniería,
Mexico

mc160006@alumnos.upa.edu.mx, martin.montes@upa.edu.mx

Abstract. Robotic arms have played a significant role in industrial and technological development, automating several jobs efficiently. Its main task is the tracking of defined trajectories, for which it uses a model known as inverse kinematics. This work uses linear genetic programming to solve the trajectory tracking in robots of 3 degrees of freedom with the least amount of information possible. The efficiency is analyzed on the requested route and evaluating the resulting model on different paths. This work demonstrates the tendency to optimize the local minimum and how the results are only valid for the optimized route due to the lack of information about other possible points or trajectories.

Keywords: Linear Genetic programming, path following, automatic robot modeling, inverse kinematics.

1 Introduction

Robotics is the study and design of multifunctional manipulators, capable of moving materials, parts, tools, or special devices, according to variable trajectories, programmed to perform various tasks [1]. However, this branch of science has transcended throughout history, showing scientific and technological advances applied to multiple areas of knowledge [2].

The kinematics considers robots as input-output systems, where the articular variables or inputs are associated with the movement of each motor independently, and the output or Cartesian variables describe the position and orientation of the end effector [1].

Being of interest inverse kinematics (IK onwards), to know the position and orientation of the end effector, in this way, it is possible to see the status of each joint and its movement to follow a given path [3].

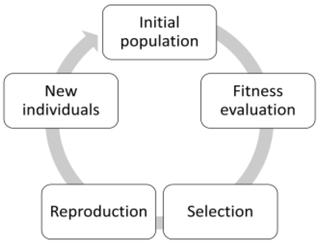

Fig. 1. GP operation in natural selection cycle [8].

There are deterministic methods to obtain the IK model that defines each robot; however, this process becomes complex with increasing degrees of freedom (DOF onwards) [4].

Machine learning allows solving high complexity problems automatically, being an alternative to solving the IK model. Genetic programming algorithms (GP onwards) are backward search algorithms that can solve problems under an unknown operation, also called black-box testing. However, GP only covers the structural search and not the content of the solution [5- 7].

The GP algorithm follows the natural selection cycle of evolutionary algorithms, as shown in Fig 1.

GP conventionally uses individuals defined as a tree structure. However, individuals of linear GP with equation structures produce a lower computational cost [8].

1.1 Related Works

In recent years the works that describe the IK model produced contributions using algebraic models for tool manipulation [9], a human arm description using the integral workspace [10], and even mimicry mechanisms for the path of the arms in an NAO robot [11].

On the other hand, computational science tries to solve using neural networks [12-14] and genetic algorithms **[**15, 16].

Some works identify the arm trajectory of an NAO robot using IK [18], the IK model obtained with iterative methods **[**19, 20**]**, as well as specialized controls in the upward approach of robotic arms using the Jacobian [21] and offline strategies [22].

These problems have been transferred to parallel robots [23], human-robot interaction interfaces [24] and movile robots [25].

The GP properties give a proposal to cover the problem described. This method reported [26] highlighted the benefits of this method. However, it was not until the works [17,27] that the proposed method was applied, finding an approach of the IK solution in 6 DOF robots.

2 Theoretical Framework

A robot manipulator composition includes joints and links; the links are the rigid structural elements, while the joints are the existing connections between links, which allow the relative movement between them; each joint adds DOF to the robot, depending on its morphology [1].

Kinematics is the science that analyzes these structures, in this case, called robots, and their movements in space independently of the force that generates them [4].

2.1 Inverse Kinematics

The inverse kinematics approach produces nonlinear and coupled equations for describing the three-dimensional motion in conjunction with orientation in a Cartesian space [2].

This procedure needs to obtain the homogeneous matrices describing each link under the Denavit-Hartenberg convention and thus obtain the inverses of these matrices [1].

These matrices premultiplied the transformation matrix described in (1):

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Applying the process in (2), are obtain a set of cleared equations for each joint:

$$(A_1^0)^{-1} * T = A_2^1 * \ldots A_n^{n-1} \tag{2}$$

Path following. Most robot applications require a proposed route, approaching the search of trajectories from different points of view [4].

One of the problems is the search for the optimal path [28]. Other works focus on the lower consumption of resources by the motors involving the dynamics of the robot [29] and the smooth path approach for specific applications [30].

2.2 Linear Genetic Programming

The GP is an evolutionary algorithm specialized in developing computer programs to solve given problems; this algorithm is a specialization of the genetic algorithms with the difference that the individuals used are complete programs [5, 31].

These algorithms are inspired by the adaptation of species and follow the cycle shown in Fig. 1.

The algorithm starts with initializing the population of individuals representing possible solutions; evaluation of the fitness. Then the tournament selection takes individuals better able to reproduce and inherit their genes, as illustrated in Fig. 2.

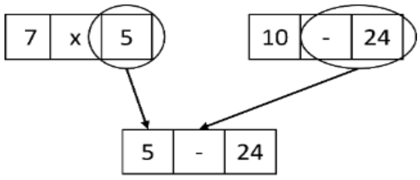

Fig. 2. Selection and crossover representation for GP algorithms [32].

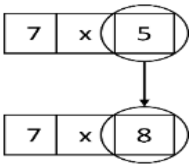

Fig. 3. Mutation representation for GP algorithms [32].

These processes increase the possibility of mutation, where one allele is randomly changed, using the Node Replacement Mutation [5, 33], shown in Fig. 3.

Furthermore, this process is repeated over a known number of generations or until the desired goal is reached [5].

3 Methodology

This section describes the experimental setup and the GP solution to find the model of a robotic arm following a known path and check whether the results obtained describe the IK or optimize a local minimum.

3.1 Design of Experiment

Start from the assumption that the purpose of a robotic arm must follow a path for a given task; the problem consists of knowing the kinematic model that describes the robot along with that movement.

It is considered an input-output system, where the expected coordinates of the end effector along the path are known as input and need to know the positions of each joint to achieve the task as output, restricting the movement between point and point to involve a slight change in the joints.

A linear GP algorithm builds its individuals using integers from 0 to 9, operators for addition, subtraction, multiplication and division operations, sine and cosine functions, and Cartesian coordinates, x, y, and z, as variables.

Was used a proposed 3 DOF SCARA morphology, with links of 200 mm in length each, because it is a robot with a known kinematic model and is typical morphology in electronic assembly lines.

Were set two different study cases; both paths move from end to end in the work area; the first route described in (3), with x1, y1, z1, contemplates a straight trajectory

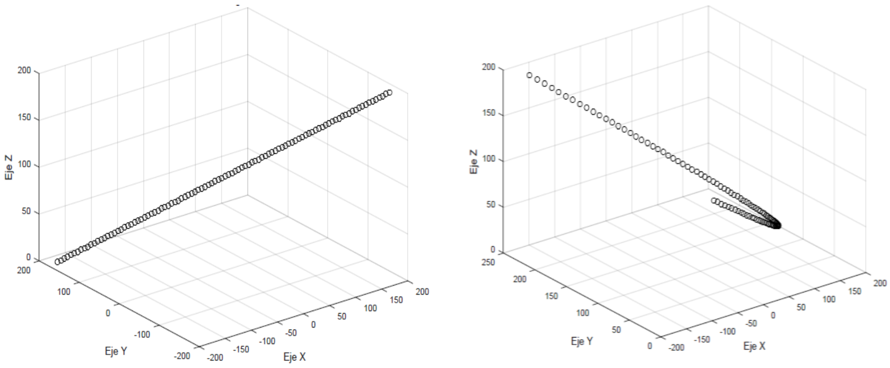

Fig. 4. a) The first case path follows; b) The second case path follows.

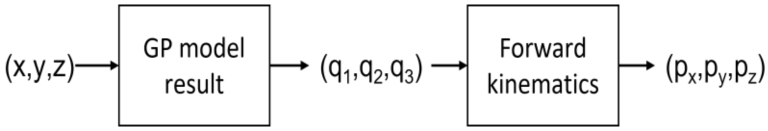

Fig. 5. Block diagram for GP objective function used.

shown in Fig. 4 a). The second case corresponds to a curved path in (3), with x2, y2, z2, and its trajectory corresponds to Fig. 4 b):

$$\begin{array}{ll} x1= [-180,180]; & x2= [-180,180]; \\ y1= [180, -180]; & y2=(x/12)^2; \\ z1= [0,200]; & z2= [200,0]; \end{array} \tag{3}$$

All tests were simulated in MATLAB 2015b using structured code without using special commands.

3.2 Objective Function

To evaluate the individual's fitness were separated into three equations, which represented the model of each joint; then, feeding the equations with the points of the trajectories mentioned above (x,y,z). Finally, the results obtained (q1, q2, q3), fed to the model in direct kinematics and checked if these results (px, py, pz) led to the requested path shown in Fig. 5.

The individual fitness was the absolute difference in millimeters between the requested point and the obtained position, as shown in (4):

$$\text{Fitness}=\text{abs}(x-p_x) + \text{abs}(y-p_y) + \text{abs}(z-p_z) \tag{4}$$

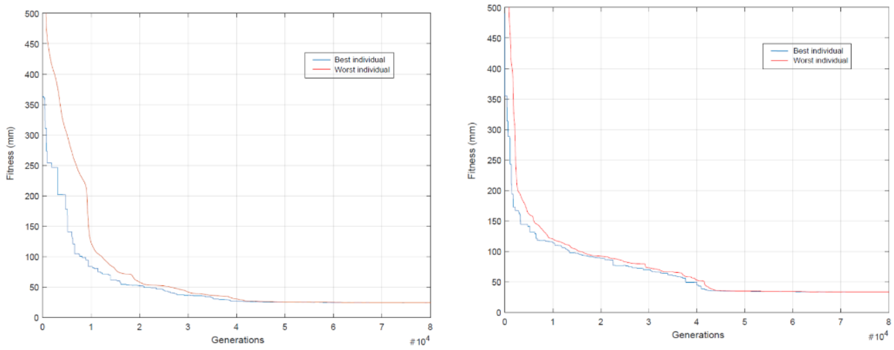

Fig. 6. a) Population adaptation in GP case 1, b) Population adaptation in GP case 2.

Table 1. Initial condition used in the GP algorithm.

Parameter	Value
Population size	2500
Generations	80000
Tournament size	3
Genes	6
Alleles	7
Mutations per generations	28
Seed	1

3.3 GP Algorithm

The selection of initial conditions was according to the rules described in the literature [5], maintaining the population diversity as many generations as possible. Table 1 shows the values.

The individuals contain three independent equations to describe the kinematics, and each equation is dependent on the three cartesian axes x,y, and z (5):

$$\begin{aligned} &\text{Individual}(1,1) = f(x,y,z); \\ &\text{Individual}(1,2) = f(x,y,z); \\ &\text{Individual}(1,3) = f(x,y,z); \end{aligned} \tag{5}$$

4 Results

According to the experiment, both cases give expected adaptation results, decreasing the error as the generations progress. Fig. 6 shows in blue the adaptation of the best

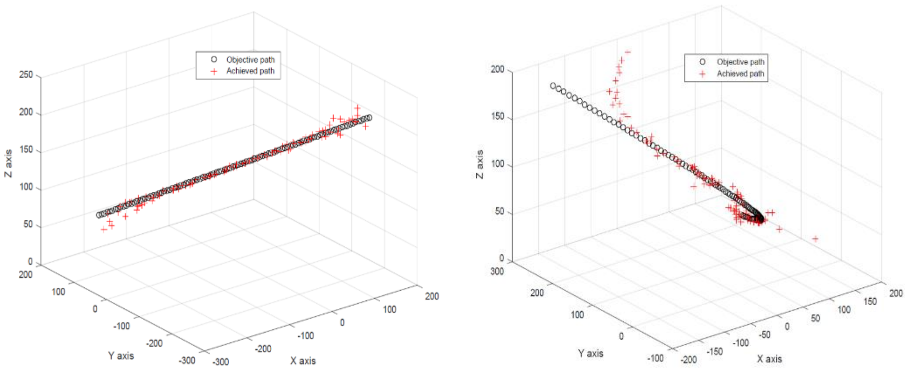

Fig. 7. a) GP model performance for case 1 path, b) GP model performance for case 2 path.

individual and how it decreases. On the other hand, the red line shows the worst individuals, representing the population diversity as the distance between these lines, disappearing at the end of the generations, being Fig. 6a) the linear path for the first case and Fig. 6b) the curve path on the second case.

As can be seen in case 1, with a linear path as the objective, the algorithm has an accelerated adaptation respect case 2, concluding that the complexity and variation in the objective path slow down the adaptation of the algorithm:

$$\begin{cases} q_1 = x + \sin(y) - z - 0.1821x - 0.5\cos(x)\cos(y) - \sin(y)^{\sin(x)} + 9 * 4^{\sin(y)} - (\sin(y)^2 - \sin(y)^{2.7183} + 10.8731)(\sin(y)^4 + \sin(y) + 0.0416z) + 40 \\ q_2 = y + 1.414z + \dfrac{(3^x + \frac{1.812}{\cos(y)})(2.718z + \frac{9}{\cos(y)} + \frac{x\cos(x)}{\cos(z)})}{(0.399\sin(z) - y^4)(\frac{2.718}{y^5} - \cos(y) + 4)(2.718^x - 0.367\cos(y)^y + 1296)} + 3 \\ q3 = z + 0.833y + \dfrac{z - \frac{3\cos(x)\sin(z)}{2.718^{\sin(x)}}}{x^{6z} - 262134} + \dfrac{y}{x} - \dfrac{1}{(\frac{2\sin(x)}{\sin(z)} + 6.2817)^{5z+\sin(x)}} - 49 \end{cases} \tag{6}$$

$$\begin{cases} q_1 = x + \dfrac{\sin(x)^6}{\cos(z)} - z - 0.333\cos(x) - \cos(z) - \dfrac{\frac{13.2436}{\cos(z)} - 2}{403.428\sin(y) + 5} - 0.75x + \dfrac{0.2333xz}{y} - 35.027 \\ q_2 = y + 0.111z - 1.333y - \cos(x) - \cos(z) + \dfrac{4x + z - \cos(x) + \frac{y\sin(x)}{\cos(y)}}{2^{\cos(y)}y - 5\cos(y) + 13} - \dfrac{\sin(x)}{\cos(y)} + 8^{\cos(z)}\cos(x) + 1.953x10^6 \\ q3 = z + 1.111x \end{cases} \tag{7}$$

It is notorious that in both cases, the results were different. To visualize what the GP achieved in each case, Fig. 7a) shows case 1 with the model in (6), and Fig. 7b) shows case 2 with the model in (7). The chart shows the objective paths in circles in both cases, and the achieved paths in crosses.

In both cases, the results obtained by the GP tend to the requested path. However, it is essential to note that this soft search algorithm approximates the structure that meets the requested objective and does not optimize results, which is why it results in a similar path.

Table 2. Initial condition used in the GP algorithm.

Prove	Accumulated error of training	Accumulated error of testing
Case 1	4.599 mm	582.67 mm
Case 2	17.97 mm	154.13 mm

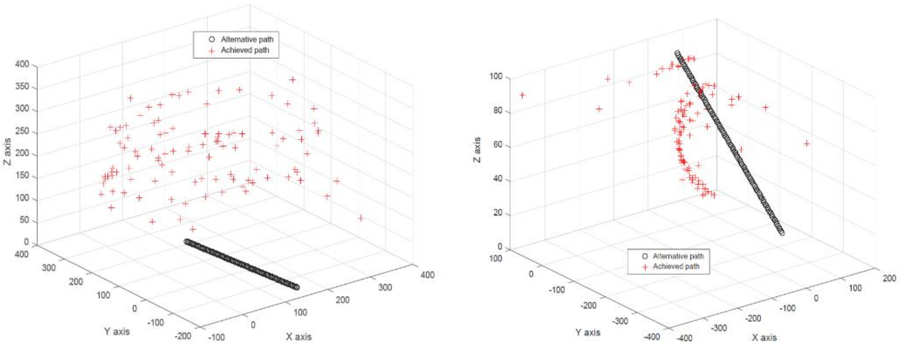

Fig. 8. a) Verification of results on an alternate route for case 1 b) Verification of results on an alternate route for case 2

These results are not conclusive but show that training requires several more points with an alternative path to produce an IK model.

Fig. 8a) shows case 1, and Fig. 8b) shows case 2 performances. It is identified from this analysis that the requested route in circles was not the route obtained by the robot in crosses.

Observing that the results are far away and different from the requested test route is verifiable by comparing the accumulated errors shown in Table 2. This way, the results are limited to the training routes.

It demonstrates the tendency to optimize the local minimum due to the lack of system information.

5 Conclusions

The tests carried out in this work demonstrate that a complex system such as a robotic arm requires several points to generate an IK model. Thus, soft search algorithms cannot consider all possible scenarios if there is no representative dataset of the search space. On the other hand, feeding the algorithm as poor information may reach a local minimum that follows the training path in the dataset used, i.e., the lack of information does not allow it to see the cases in which it fails.

In any case, the GP can solve this complex problem for specifically requested routes but fail in other scenarios.

These results are not an error and provide valuable information. Based on them, it is possible to identify the following failures extrapolated to any application. Moreover, it is vital to restrict the variables that affect the model and how they do it considering the third joint should not be a function of the x, y axes. On the other hand, is suggested a trying meshes of points with the expected resolution in the robot as a training dataset for future works using GP to obtain the IK model.

In the same way, the fitness function must contemplate a combinatorial of possible solutions to avoid that the GP disregards important points of work.

References

1. Barrientos, A., Peñil, L., Balaguer, C.: Fundamentos de Robótica (2007)
2. Baturone, A.O.: Robótica: Manipuladores y Robots Móviles. Marcombo (2005)
3. Bertram, D., Kuffner, J., Dillmann, R.: An Integrated Approach to Inverse Kinematics and Path Planning for Redundant Manipulators. In: International Conference on Robotics and Automation. ICRA, pp. 1874–1879 (2006) doi: 10.1109/ROBOT.2006.1641979.
4. Saha, S.K.: Introducción a la robótica. McGraw-Hill (2008)
5. Poli, R., Langdon, W.B., McPhee, N.F.: A Field Guide to Genetic Programming (With contributions by JR Koza) (2008)
6. Howard, L.M., D'Angelo, D.J.: The GA-P: A Genetic Algorithm and Genetic Programming Hybrid. IEEE Expert, 10(3), pp. 11–15 (1995) doi: 10.1109/64.393137.
7. Alfred, R.: Short-Term Time Series Modelling Forecasting Using Genetic Algorithm. In: Proceedings of the International Conference on Data Engineering (DaEng-2015), pp. 163–171 (2019) doi: 10.1007/978-981-13-1799-6_18.
8. Rivera, M.M., Ramos, M.A.P., Mora, J.L.O.: An Alternative Structure to Replace the Tree-Based Individuals in a Genetic Programming Algorithm for Decoupling MIMO Systems. In: 6th MICAI Workshop on Hybrid Intelligent Systems (2013)
9. Gao, Z., Wei, G., Dai, J.S.: Inverse Kinematics and Workspace Analysis of the Metamorphic Hand. Proceedings of the Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science, 229(5), pp. 965–975 (2015) doi: 10.1177/0954406214541429.
10. Nguyen, H.V., Le, T.D., Huynh, D.D.: Forward Kinematics of a Human-Arm System and Inverse Kinematics Using Vector Calculus. In: 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), pp. 1–6 (2016) doi: 10.1109/ICARCV.2016.7838641.
11. Alibeigi, M., Rabiee, S., Ahmadabadi, M.N.: Inverse Kinematics Based Human Mimicking System Using Skeletal Tracking Technology. Journal of Intelligent and Robotic Systems, 85(1), pp. 27–45 (2017) doi: 10.1007/s10846-016-0384-6.
12. Duka, A.V.: ANFIS Based Solution to the Inverse Kinematics of a 3DOF Planar Manipulator. Procedia Technol, 19, pp. 526–533 (2015) doi: 10.1016/j.protcy.2015.02.075.
13. Raj, D.R., Raglend, I.J., Anand, M.D.: Inverse Kinematics Solution of a Five Joint Robot Using Feed Forward and Radial Basis Function Neural Network. In: International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC), pp. 117–122 (2015) doi: 10.1109/ICCPCT.2015.7159376.
14. Olaru, A., Olaru, S., Mihai, N.: Proper Smart Method of the Inverse Kinematic Problem. In:

Applied Mechanics and Materials, 772, pp. 455–460 (2015) doi: 10.4028/www.scientific.net/AMM.772.455.

15. Momani, S., Abo-Hammour, Z.S., Alsmadi, O.M.K.: Solution of Inverse Kinematics Problem Using Genetic Algorithms. Applied Mathematics Information Sciences, 10(1), pp. 225 (2016) doi: 10.18576/amis/100122.
16. Gupta, R.A., Chow, M.Y.: Overview of Networked Control Systems. In: Wang, F.Y., Liu, D.: Networked Control Systems. pp. 1–23 (2008) doi: 10.1007/978-1-84800-215-9_1.
17. Chapelle, F., Bidaud, P.: Closed Form Solutions for Inverse Kinematics Approximation of General 6R Manipulators. Mechanism and Machine Theory, 39(3), pp. 323–338 (2004) doi: 10.1016/j.mechmachtheory.2003.09.003.
18. Zhai, R., Wen, S., Zhu, J.: Trajectory Planning of NAO Robot Arm Based on Target Recognition. In: International Conference on Advanced Mechatronic Systems (ICAMechS), pp. 403–407 (2017)
19. Sadiq, A.T., Raheem, F.A.: Robot Arm Path Planning Using Modified Particle Swarm Optimization Based on D* Algorithm. Al-Khwarizmi Engineering Journal, 13(3), pp. 27–37 (2017) doi: 10.22153/kej.2017.02.001.
20. Becerra, Y., Arbulu, M., Soto, S.: A Comparison Among the Denavit-Hartenberg, the Screw Theory, and the Iterative Methods to Solve Inverse Kinematics for Assistant Robot Arm. In: International Conference on Swarm Intelligence, pp. 447–457 (2019) doi: 10.1007/978-3-030-26369-0_42.
21. Krastev, E.: Kinematic Path Control of a Redundant Robot Arm in Sliding Mode. In: International Conference on Robotics in Alpe-Adria Danube Region, pp. 281–288 (2017) doi: 10.1007/978-3-319-61276-8_31.
22. Praveena, P., Rakita, D., Mutlu, B.: User-Guided Offline Synthesis of Robot Arm Motion from 6-DoF Paths. In: International Conference on Robotics and Automation (ICRA), pp. 8825–8831 (2019) doi: 10.1109/ICRA.2019.8793483.
23. Michelin, M., Hervé, P.E., Tempier, O.: Path Following Demonstration of a Hybrid Cable-Driven Parallel Robot. In: International Conference on Cable-Driven Parallel Robots, pp. 323–335 (2021) doi: 10.1007/978-3-030-75789-2_26.
24. Boehm, J.R., Fey, N.P., Majewicz, A.: Inherent Kinematic Features of Dynamic Bimanual Path Following Tasks. IEEE Trans Human-Machine Syst., 50(6), pp. 613–622 (2020) doi: 10.1109/thms.2020.3016084.
25. Diaz, E., López, A.S., Serna, M.: Planificación reactiva de movimientos en tiempo real para robots móviles. Research in Computing Science, 147(7), pp. 115–128 (2018) doi: 10.13053/rcs-147-7-10
26. Gibbs, J.: Easy Inverse Kinematics Using Genetic Programming. In: Proceedings of the 1st Annual Conference on Genetic Programming, 422 (1996)
27. Chapelle, F., Bidaud, P.: A Closed Form for Inverse Kinematics Approximation of General 6R Manipulators Using Genetic Programming. In: International Conference on Robotics and Automation, ICRA, 4, pp. 3364–3369 (2001) doi: 10.1109/ROBOT.2001.933137.
28. Low, E.S., Ong, P., Cheah, K.C.: Solving the Optimal Path Planning of a Mobile Robot Using Improved Q-Learning. Robotics and Autonomous Systems, 115, pp. 143–161 (2019) doi: 10.1016/j.robot.2019.02.013.
29. Lavín-Delgado, J.E., Solís-Pérez, J.E., Gómez-Aguilar, J.F.: Trajectory Tracking Control Based on Non-Singular Fractional Derivatives for the PUMA 560 Robot Arm. Multibody

System Dynamics, 50(3), pp. 259–303 (2020) doi: 10.1007/s11044-020-09752-y.

30. Kalra, P., Prakash, N.R.: A Neuro-Genetic Algorithm Approach for Solving the Inverse Kinematics of Robotic Manipulators. In: International Conference on Systems, Man and Cybernetics, 2, pp. 1979–1984 (2003) doi: 10.1109/ICSMC.2003.1244702.

31. Rivera, M.M., Ochoa-Zezzatti, A., Ramírez, C.A.M.: Automatic Generation of Programs for Data Tables with Batch Least Square Mamdani Inference Systems: Application in the AWG Table. Research in Computing Science, 148(6), pp. 39–49 (2019) doi: 10.13053/rcs-148-6-3.

32. Sivanandam, S.N., Deepa, S.N.: Introduction to Genetic Algorithms. Science and Business Media (2007) doi: 10.1007/978-3-540-73190-0.

33. Torres, M.A.G., Ponomaryov, V.I.: Selección óptima de cartera de inversión de stocks de S&P 500: Uso de algoritmos genéticos. Research in Computing Science, 148(10), pp. 271–278 (2019) doi: 10.13053/rcs-148-10-23.