

# Forward Kinematics for 2 DOF Planar Robot using Linear Genetic Programming

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.** In the field of robotics, forward kinematics is an activity that allows finding a mathematical model for the resulting position in the final effector based on the robot joints position, a popular alternative for determining this model is defined by the Denavit Hartenberg convention, nevertheless, this method requires knowledge about linear algebra and three-dimensional spatial kinematics. Machine learning uses specific computational methodologies to solving similar problems in several areas, so it could be a viable answer for automatic determining of forward kinematics. In this work we propose the use of genetic programming as a machine learning algorithm for finding the forward kinematics of a 2 degrees of freedom robot, getting a satisfactory outcome obtaining a satisfactory result with blocks that describe the expected solution, validating the capacity of the genetic programming in order to validate this algorithm for later work with more complex robots.

**Keywords:** forward kinematics, automatic robot modeling, linear genetic programming.

## 1 Paper Structure

In the introduction section, a general description is presented where the problem is discussed, the proposed solution, and the scope sought.

The related work presents all the previous investigation that supports the problem as an open issue, the applications that validate the genetic programming as a solution and its characteristics.

In the theoretical framework, the emphasis is placed on the problem, its origin, and the conventional solution and then is described all the related to the proposed solution.

The methodology presents a detailed description of the problem, as well as the solution process used, its behavior and the tests carried out.

The results section shows the data obtained from the algorithm as well as its performance, the validation applied to the results and the discussion about the impact of them with respect to those found in the literature.

Finally, the conclusions describe the contributions, the scope of the results and the future work arising from it.

## 2 Introduction

Robotics remains to be a growing research area, which has become relevant since the presentation of the first industrial robot in 1954 [1].

One of the problems present in robotics is the forward kinematics (FK onwards) model, which relates the positions of the joints to the Cartesian positions of the final effector, for which there are several deterministic methods of solution, but the problem requires previous knowledge in different mathematical areas, like linear algebra and vector spaces [2].

Machine learning allows the solution of complex problems without the intervention of human beings that is why represents an alternative for solving the FK problem. Evolutionary algorithms (EAs onwards) are machine learning optimizing algorithms that can be classified in single or multi-objective algorithms, in general, EAs algorithms follow specific operators which include, randomly initialized population, evaluation, fitness assignment, selection and reproduction (Fig. 1) [3,4].

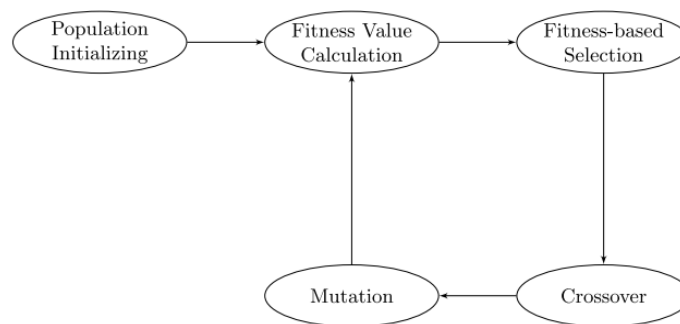


Fig. 1. EA main operators algorithm taken from [3].

EAs have presented important advances in various branches of science [4–8], where they are presented as an alternative method to problems in which the deterministic solution is not successful[9].

The most used EAs are Genetic algorithms (GAs onwards) and Genetic Programming (GP onwards), both based on natural selection principles proposed by Charles Darwin, but applied in different situations. GAs perform numerical optimization on known structures or programs [10], on the other hand, GP performs structural optimization on unknown structures or programs allowing to determine mathematical models, like those required in FK. [3,11].

EAs have been used for various problems in robotics[12], such as tracking people [13], making decisions robots according to their urgency [14], among others, but genetic algorithms are restricted to numerical optimization for known structures [15], Therefore FK cannot be applied directly with GAs, because what that solution the transformation matrix must be determined for each given position which would require high computational power. Nevertheless, if mathematical equations in those matrixes are determined instead of those numbers then the solution model would be found with GP as an optimizing algorithm but only requiring one initial run [16].

## 2.1 Related Works

In recent years FK problem has been studied by different methodologies, some of which use geometric analysis to find a simple solution applied to parallel robots [17], and with this technique, they look for solutions for anthropomorphic mechatronic systems where are used 4x4 matrixes that work as operators for solving FK problem [18].

Other investigations show models that are based on the length of the links for transmitting the movement in a hybrid robot [19] and using this research in [20] models are explored based on flexible robots which are not considered in the aforementioned convention.

Another tool used is the quaternions, which use an extension of the real numbers and for this case allow to describe the movement of the coordinates along a kinematic chain [21].

On the other hand, there is the use of computational tools, which have been explored less frequently, within the highlighted works an hybrid algorithm was used between the search with particle swarming and a differential evolution applied to solve FK problem with remote manipulators [22].

Likewise, the use of neural networks has been recurrent for autonomous FK solution, like when applying complex networks to solve FK with parallel robots in [23], and redundant robots in [24].

In the case of GAs, they have been applied with hybrid processes that take advantage of other algorithms. GAs have been tested working in conjunction with simulated annealing in [25] and have been used with geometric similarities for optimizing its numerical parameters [26].

In this work, it is proposed to find the Denavit Hartenberg (DH onwards) parameters with mathematical expressions obtained with a GP algorithm in a robot with 2 degrees of freedom.

## 3 Theoretical Framework

Kinematics is the branch of physics that analyzes movements without considering the forces that cause them [27]; applied to a robot, the kinematic models describe the relationship between each of the robot joints and its final actuator position [2], where it is possible to determine FK forward or backward depending from if the final actuator position is found or the joints position respectively.

### 3.1 Forward Kinematics

Forward kinematics is the mathematical model that relates the known positions of each of the joints and their relationship with the Cartesian axes [2].

The DH convention describes the behavior of the  $l_i$  link with respect to the  $l_{i-1}$  link, with 4 important transformations described with the below parameters [28].

- $\Theta_i$ , angle of the joint, with which the orthogonal plane is projected with respect to the normal anterior plane.
- $d_i$ , displacement of the joint, length between the links with respect to their joints.

- $a_i$ , length of the link between the common perpendiculars.
- $\alpha_i$ , torsion angle between the orthogonal projections of the Z axis in a perpendicular plane.

The homogeneous matrix that represents those 4 transformations previous parameters is shown in equation (1):

$$A_1^0 = \begin{bmatrix} \cos(\theta) & -\text{sen}(\theta) & \text{sen}(\theta) & a * \cos(\theta) \\ \text{sen}(\theta) & \cos(\theta) & -\cos(\theta) & a * \text{sen}(\theta) \\ 0 & \text{sen}(\alpha) & \cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

Following the DH rules are possible to find a transformation matrix that contains the equations that describe the position and orientation of the robots in its final effector with respect to the values given by its joints [1].

### 3.2 Linear Genetic Programming

This algorithm is considered within the meta-heuristics, since it is an EA that performs a soft search to determine the structure that contains a possible solution [29].

The form of work proposed is that described in Algorithm 1, which specifies the evolutionary cycle that allows a population of possible responses to interact to generate an increasingly better-adapted offspring to the given problem.

```

GP Program (Best Individual)
  Initial values;
  Begin
    Start_population();
    Evaluate_apititude();
    Repeat
      Select();
      Cross();
      Evaluate_apititude();
      Mutation();
      Evaluate_apititude();
    Until aptitude = Expected OR End_generations
  End.
    
```

**Algorithm 1.** Fitness evaluation algorithm.

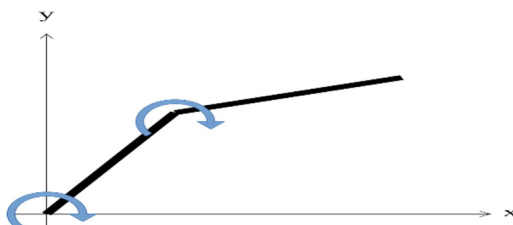
Population randomly initialized generates possible individuals, which are evaluated in the problem to be solved, once the cycle has begun, the individuals who may have children in each generation are selected, the children are created from the parents and in some cases have mutations, finally the cycle is evaluated and repeated until the desired fitness value is reached, or the desired generations are achieved [30].

## 4 Methodology

This section explains how the GP tool was built and the problem of finding DH parameters for a robot automatically and how the proximity to the expected result was evaluated.

### 4.1 Design of Experiment

For this work, a 2-degree freedom planar robot with two rotational joints is used, its schematic is shown in **Fig. 2**.



**Fig. 2.** Planar robot diagram.

It was decided to make use of this robot for its ease and because the search method used has no precedent, it was initialized with a known robot, which, by modifying any of its joints, moves its final effector along the x and y axes.

For simulation MATLAB 2015b was used since the algorithm was worked as a test, for the moment no special command was used, everything was worked with structured code.

### 4.2 Objective Function

In order to evaluate the suitability of the individuals in FK, the results of the GP were taken and placed in a 2x4 matrix and used to assemble the transformation function according to the DH rules, taking the first column in the matrix as the joint angle, the second as the joint displacement, the third as the link size and the last of them as the angle between the z-axis projection and the next plane.

These values were evaluated in a robot of 300 and 400 cm per link in the position of  $\frac{\pi}{3}$  and  $\frac{\pi}{4}$  in each joint respectively, the positions selected here considered that the size of the operation could not exchange its values, in addition to working within the first quadrant of work, thus accumulating the absolute value of the subtraction between the known solution of this robot following the DH convention and those obtained by the GP as is shown at equation (2).

$$\text{Error} = \text{abs}(X_{\text{DH}} - X_{\text{GP}}) + \text{abs}(Y_{\text{DH}} - Y_{\text{GP}}) + \text{abs}(Z_{\text{DH}} - Z_{\text{GP}}). \quad (2)$$

### 4.3 GP Algorithm

This algorithm was executed with the initial conditions described in Table 1, which were selected according to the suggested rules in the literature, where decisions are made for increasing or decreasing each term based on the results obtained and its behavior [30].

**Table 1.** Initial values of GP.

| Initial condition        | Value           |
|--------------------------|-----------------|
| Seed                     | 1               |
| Population size          | 200 individuals |
| Genes number             | 8               |
| Alleles number           | 4               |
| Tournament size          | 3               |
| Generations              | 1500            |
| Mutation percentage      | 10%             |
| Mutation numbers         | 1800            |
| Mutations per generation | 12              |

After this a basic GP code was assembled following algorithm 1, where individuals with 8 blocks were used each block describing a DH parameter in the table, those blocks are functions of  $q_i$  and  $l_i$  having the structure shown in equation (2):

$$B_i = f(q_i, l_i) \cdot \tag{3}$$

These blocks are assembled into a 2 x 4 matrix and subtracted against the previously found DH parameter matrix.

## 5 Results

First of all, the DH parameters were obtained following the rules that this convention requires, in this way the results shown by the algorithm will be confirmed. In this way, the parameters of Table 2 were obtained.

**Table 2.** Planar robot DH parameters.

| Transformation | $\Theta$ | d | a     | $\alpha$ |
|----------------|----------|---|-------|----------|
| ${}^0A_1$      | $q_1$    | 0 | $l_1$ | 0        |
| ${}^1A_2$      | $q_2$    | 0 | $l_2$ | 0        |

The parameters obtained were simulated to visualize the behavior of the robot, and to obtain the graph shown in Fig. 3. Where an angle was used in the first joint of  $\frac{\pi}{3}$  and in the second joint with  $\frac{\pi}{4}$ .

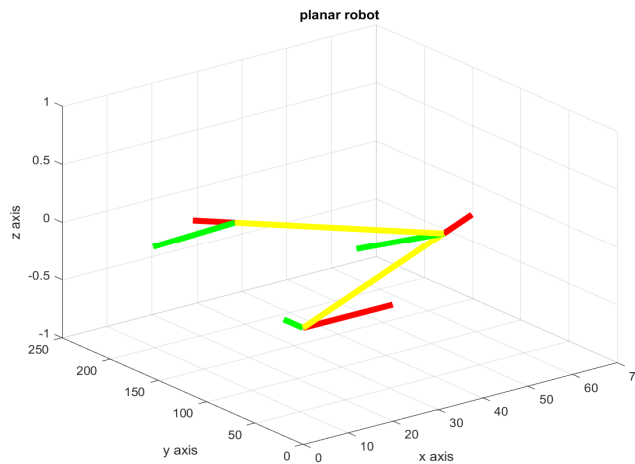


Fig. 3. Planar robot graph.

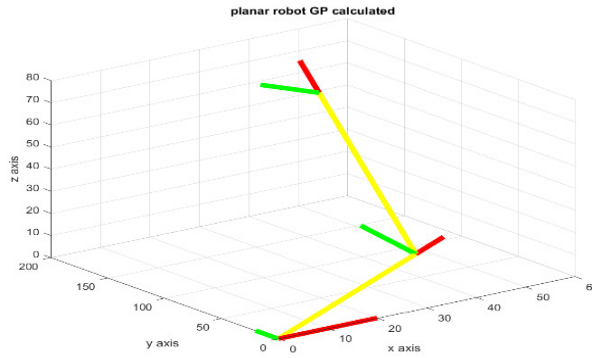
The next step was to take the results of the GP, which formed the matrix shown in Table 3. These have a larger number of parameters, as this algorithm required more space to increase the search diversity.

Table 3. Best element obtained by the GP.

| Transformation | $\Theta$              | d               | a         | $\alpha$            |
|----------------|-----------------------|-----------------|-----------|---------------------|
| ${}^0 A_1$     | $q_1 + \frac{2}{l_2}$ | 0               | $l_1 - 1$ | $1 - \frac{3}{l_2}$ |
| ${}^1 A_2$     | $q_2$                 | $\frac{2}{l_2}$ | $l_2$     | 0                   |

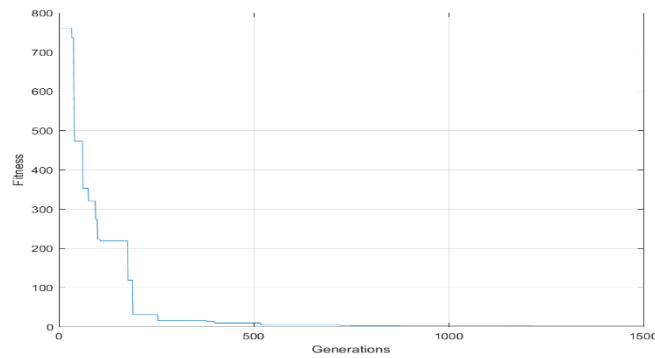
All the solutions as expected have good fitness i.e. equations satisfy FK, but the representation shown by the result requires to be cleaned or that unused extra data be simplified, as was presented at the literature, all the results take the structure to where the correct data is [29].

The graph shown in Fig. 4 shows the position of the robot at the angles mentioned above, where the similarity between the positions obtained can be seen.



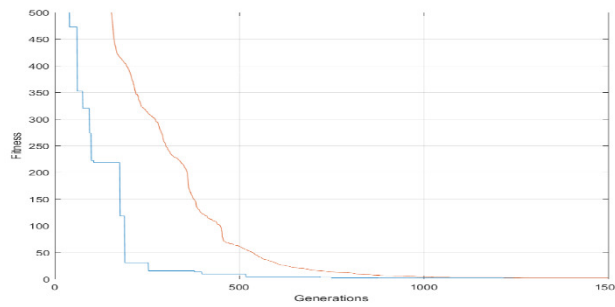
**Fig. 4.** Planar robot graph obtained by the GP.

The evolution of fitness of the GP can be seen in Fig. 5, which shows a desired behavior that favors the search for the correct result.



**Fig. 5.** Population fitness.

At the same time, Fig. 6 shows the best and worst element, where the difference between the two shows the diversity of the population, which disappears when it reaches the best value found.



**Fig. 6.** Behavior of diversity.



The results shown above are only visual, but in order to corroborate mathematically the results, the equation (4) shows the expected evaluated matrix for the planar robot when substituting the links  $l_1 = 300$  and  $l_2 = 400$ , and joints value with  $q_1 = \frac{\pi}{3}$  and  $q_2 = \frac{\pi}{4}$ :

$$DH = \begin{bmatrix} 1.0672 & 0 & 100 & 0 \\ 0.7854 & 0 & 100 & 0 \end{bmatrix}. \quad (4)$$

In comparison with the equation (4), the equation (5) shows the GP algorithm solution in the robot using same values of equation (4) for links and joints, it can be seen that most of them are similar, in addition to the symbolic results are shown:

$$DH_{GP} = \begin{bmatrix} 1.0672 & 0 & 99 & 0.97 \\ 0.7854 & 0.02 & 100 & 0 \end{bmatrix}. \quad (5)$$

## 6 Conclusion

The relative error in the GP result is over 6.57%, this error comparing with the neuronal network develop is higher [23], [24], where the error has a 6% of difference, the same case happened using a PSO algorithm [25], with a 6.5% of error, that gets a lower difference in the absolute error but have an increase in execution time because the PSO use 0.5 milliseconds less by execution, finally the genetic algorithm [26] just have a 4% of error. The performance of GP is lower because the kind of algorithm used optimizes structure but not numerical parameters, i.e. GP is a soft search that just finds how the answer looks like and giving a really close approximation comparing to other three optimization algorithms, however if numerical parameters are optimized using other numerical optimization algorithm the solution could be improved over other works.

In this way it can be concluded that the problem posed by the introduction has a solution by the method selected, and that there is a result achievable by the GP for this problem, but numerical parameters must be optimized after that and structural solution is obtained.

The above mentioned gives rise to continuing to work on the search for the model and to generalize it, as well as to propose a solution to mathematically complex structures as preliminary results, the aim is to be able to add to the algorithm presented an automatic optimization process, this to compensate for the limitations of the GP; the code will be adapted to work parallel to the cycles reducing the execution time; tests will be carried out, which include the use of a redundant parallel robot that is the main problem of the FK; once satisfactory results have been found in the structural solution other numerical optimizing algorithm must be used for optimizing numerical parameters in the obtained structure, after that, the problem of inverse kinematics will be tackled, with the aim of finding a general computational method for the mathematical models of the robot.

## References

1. Baturone, A.O.: *Robótica: manipuladores y robots móviles*. Marcombo (2005)
2. Saha, S.K.: *Introduction to robotics*. Tata McGraw-Hill Education (2014)
3. Montes, M., Padilla, A., Canul, J., Ponce, J., Ochoa, A.: Comparative of Effectiveness When Classifying Colors Using RGB Image Representation with PSO with Time Decreasing Inertial Coefficient and GA Algorithms as Classifiers. In: *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*, pp. 527–546. Springer (2018)
4. Rivera, M.M., Justo, M.O.A., Zezzatti, A.O.: Equations for Describing Behavior Tables in Thermodynamics Using Genetic Programming: Synthesizing the Saturated Water and Steam Table. *Res. Comput. Sci.* 122, 9–23 (2016)
5. Li, M.: Multi-Objective Evolutionary Algorithms with Immunity for SLAM. *Adv. Artif. Intell.* 26, 27–36 (2006)
6. Duque, T., Delbem, A.: An Evolvable Hardware Approach. *Adv. Artif. Intell.* 37 (2006)
7. Ortiz, H.A.T., Montejano, C.A.P., Villegas-Cortez, J., Avilés-Cruz, C.: Evolución de descriptores estadísticos de superficie de imágenes por programación genética para el reconocimiento de imágenes por CBIR: una primera aproximación. *Res. Comput. Sci.* 116, 125–134 (2016)
8. Enríquez, G.B., Sánchez-Partida, D., Morales, S.O.C.: Algoritmo genético para el problema logístico de asignación de la mochila (Knapsack Problem). *Res. Comput. Sci.* 137, 157–168 (2017)
9. Vázquez-Castillo, V., Hernández-Lara, D., Merchán-Cruz, E.A., Rodríguez-Cañizo, R.G., Portilla-Flores, E.A.: Implementación de algoritmos genéticos para el diseño, optimización y selección de vigas. *Res. Comput. Sci.* 137, 121–134 (2017)
10. Hernández, M., Velasco-Arellano, H., Ubach-González, D., Montes-Rivera, M., Aguilar-Justo, M.O.: Steering Wheel Control in Lane Departure Warning System. *Res. Comput. Sci.* 147, 9–21 (2017)
11. Weise, T.: *Global optimization algorithms-theory and application*. Self-Publ. 2 (2009)
12. Ramírez, D., Ortiz, J., Ponce, P., Molina, A.: Sistemas inmunes artificiales aplicados a la robótica: agarre de objetos. *Res. Comput. Sci.* 135, 55–70 (2017)
13. Soriano, F.G., Montero, A.R., López, A.S.: Seguimiento autónomo de personas con un robot aéreo no tripulado. *Res. Comput. Sci.* 135, 71–84 (2017)
14. González, F.M.M., Hernández, A.M., Figueroa, H.R.: An effective robotic model of action selection. In: *Conference of the Spanish Association for Artificial Intelligence* (2005)
15. Coppin, B.: *Artificial intelligence illuminated*. Jones and Bartlett Publishers (2004)
16. Sivanandam, S., Deepa, S.: *Evolutionary Computation*. In: *Introduction to Genetic Algorithms*. pp. 1–13. Springer (2008)
17. Liu, Y., Kong, M., Wan, N., Ben-Tzvi, P.: A Geometric Approach to Obtain the Closed-Form Forward Kinematics of H4 Parallel Robot. *J. Mech. Robot.* 10, 051013 (2018)
18. Petrescu, R.V., Aversa, R., Akash, B., Berto, F., Apicella, A., Petrescu, F.I.: Geometry and direct kinematics to MP3R with  $4 \times 4$  operators (2017)
19. Merlet, J.-P.: Direct kinematics of CDPR with extra cable orientation sensors: the 2 and 3 cables case with perfect measurement and ideal or elastic cables. In: *Cable-Driven Parallel Robots*. pp. 180–191. Springer (2018)
20. Faulkner, J., Dirven, S.: A generalised, modular, approach for the forward kinematics of continuum soft robots with sections of constant curvature. In: *Mechatronics and Machine Vision in Practice (M2VIP), 2017 24th International Conference on* (2017)

21. Yang, X., Wu, H., Li, Y., Chen, B.: A dual quaternion solution to the forward kinematics of a class of six-DOF parallel robots with full or reductant actuation. *Mech. Mach. Theory.* 107, 27–36 (2017)
22. Mao, B., Xie, Z., Wang, Y., Handroos, H., Wu, H., Shi, S.: A hybrid differential evolution and particle swarm optimization algorithm for numerical kinematics solution of remote maintenance manipulators. *Fusion Eng. Des.* 124, 587–590 (2017)
23. Geng, Z., Haynes, L.: Neural network solution for the forward kinematics problem of a Stewart platform. In: *Robotics and Automation 1991. Proceedings., 1991 IEEE International Conference on* (1991)
24. Sadjadian, H., Taghirad, H., Fatehi, A.: Neural networks approaches for computing the forward kinematics of a redundant parallel manipulator. *Int. J. Comput. Intell.* 2, 40–47 (2005)
25. Chandra, R., Rolland, L.: On solving the forward kinematics of 3RPR planar parallel manipulator using hybrid metaheuristics. *Appl. Math. Comput.* 217, 8997–9008 (2011)
26. Boudreau, R., Turkkan, N.: Solving the forward kinematics of parallel manipulators with a genetic algorithm. *J. Robot. Syst.* 13, 111–125 (1996)
27. Hibbeler, R.C.: *Mecánica vectorial para ingenieros: dinámica*. Pearson Educación (2004)
28. Barrientos, A.: *Fundamentos de robótica*. e-libro, Corp. (2007)
29. Howard, L.M., D'Angelo, D.J.: The GA-P: A genetic algorithm and genetic programming hybrid. *IEEE Expert.* 10, 11–15 (1995)
30. McPhee, N.F., Poli, R., Langdon, W.B.: *Field guide to genetic programming*. (2008)