SIMULATION OF ROBOTS' INVERSE KINEMATICS IN ENGINEERING EDUCATION: AN APPROACH BASED ON GENETIC ALGORITHMS

José Tarcísio Franco de Camargo[1], Eliana Anunciato Franco de Camargo[2], Estéfano Visconde Veraszto[3], Gilmar Barreto[4], Jorge Cândido[5]

ABSTRACT

The study of articulated robots necessarily goes through the development of their kinematic models. In turn, the kinematics of a robot can be described through its direct and inverse models. The inverse kinematic model, through which the state of the joints is obtained as a function of the desired position for the free end of the robot, is usually described algebraically. However, this representation is often difficult to obtain. Thus, while the exact determination of the inverse kinematic model is unquestionable, the use of genetic algorithms in the design stage can be very attractive because it allows predicting the behavior of the robot before the formal development of its model. In this sense, the results of this work present a relatively fast way to simulate the inverse kinematic model, which can be useful in teaching robotics in engineering, allowing the student to have a broader view of the model, coming to identify points that must be corrected or that can be optimized in the structure of a robot. It can be concluded that the use of genetic algorithms in robotics is feasible, having as main advantages its easy computational implementation and its precision in the representation of kinematic models.

Keywords: robotics; evolutionary algorithms; process optimization; computer simulation.

[1] Professor, Ph.D., Regional University Center of Espírito Santo do Pinhal (UNIPINHAL), Espírito Santo do Pinhal, SP; jtfc@bol.com.br.

[2] Professor, Ph.D., Regional University Center of Espírito Santo do Pinhal (UNIPINHAL), Espírito Santo do Pinhal, SP; eafcamargo@yahoo.com.br.

[3] Professor, Ph.D., Federal University of São Carlos (UFSCar), São Carlos, SP; estefanovv@ufscar.br.

[4] Professor, Ph.D., State University of Campinas (UNICAMP), Campinas, SP; gbarreto@dsif.fee.unicamp.br.

[5] Professor, Ph.D., Federal University of Technology (UTFPR), Campo Mourão, PR; jocandido@utfpr.edu.br.

INTRODUCTION

The motion described by a manipulator robot can be represented by its direct and inverse kinematic models, as described in Craig (2017). Through the direct kinematic model it is possible to determine the position and orientation of the robot's end effector in terms of the state of its joints or degrees of freedom (DOF). Obtaining the direct kinematic model is relatively simple, which is defined by a set of transformations among the coordinate systems of each DOF.

The inverse kinematic model, in turn, allows the determination of the state of the joints of a robot according to the desired position for its tool. In this way, when a trajectory for the tool is defined, it is possible to determine the set of joint positions that will allow the robot to describe the desired movement (Miller 2017).

Obtaining the inverse kinematic model, however, tends to be more complex than obtaining the direct kinematic model, since it involves the solution of a system of non-linear equations that can admit more than one solution. Even in relatively simple cases, the definition of the inverse kinematic model is not trivial.

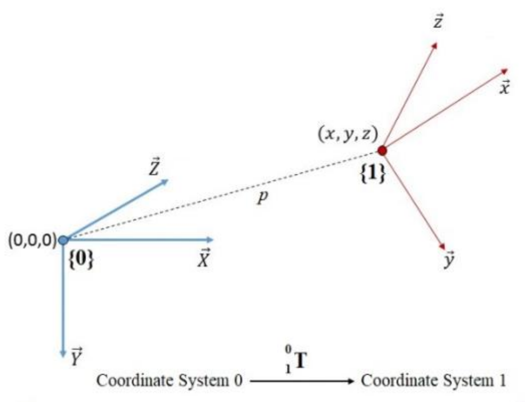
In this way, being able to predict the behavior of a robot in a relatively simple way, before the formal development of its inverse kinematic model, can become a relevant factor for learning robotics in an engineering course. Through the use of genetic algorithms (GAs) it is possible to simulate the behavior of a robot, determining with relative precision the state of its joints in function of the desired position for its free end, allowing design failures to be detected, as well as the identification of possible points for optimization.

Thus, this paper aims to present the theme of GAs in the context of simulation in robotics, trying to present a generic solution capable of representing the behavior of inverse kinematic models of articulated robots in a practical, efficient and relatively simple way, with a view to explore optimization opportunities in robotic projects, during an engineering course.

KINEMATICS FOR A GENERIC ROBOT

The study of the direct and inverse kinematic models in robotics can be introduced by the concept of "transformations" among coordinate systems, which will be attached to the DOF of a robot. Figure 1 represents an example where a transformation can be defined between frames **{0}** and **{1}**.

Figure 1 – Two coordinate systems that can be associate by a transformation.



This transformation can be represented by a rotation matrix, that represents the orientation between the frames, and a vector, that represents the distance between the systems. The matrix form that defines the transformation between the coordinate systems in Figure 1 is described in Equation (1).

$$^0_1\boldsymbol{T} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

In this transformation, the orientation between these systems can be stated as the rotation matrix presented in Equation (2).

$$^0_1\boldsymbol{R} = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \tag{2}$$

In its turn, Equation (3) represents the distance vector between these systems.

$$\boldsymbol{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \qquad (3)$$

In general, for a robot with *n* DOF, there will be a reference coordinate system attached to each DOF. This way, the transformation matrix between the inertial reference frame **{0}** and the last frame **{n}** attached to the robot's end effector will determine the position and orientation of the tool regarding the inertial coordinate system. This transformation matrix can be described by Equation (4).

$${}^{0}_{n}\boldsymbol{T} = {}^{0}_{1}\boldsymbol{T} . {}^{1}_{2}\boldsymbol{T} . \cdots . {}^{n-1}_{n}\boldsymbol{T} \qquad (4)$$

In Equation (4), ${}^{0}_{1}\boldsymbol{T}$ represents the transformation between frames **{0}** and **{1}** and so on. This way, for a robot with only rotational joints, Equation (4) can be rewritten as shown in Equation (5).

$${}^{0}_{n}\boldsymbol{T}(\theta_1, \cdots, \theta_n) = {}^{0}_{1}\boldsymbol{T}(\theta_1) . \cdots . {}^{n-1}_{n}\boldsymbol{T}(\theta_n) \qquad (5)$$

Equation (5) means that, for a robot built only with rotational joints, the transformation between frames **{i-1}** and **{i}** depends only on its respective rotational angle θ_i. In turn, the transformation among frames **{0}** and **{n}** will depend on all rotational angles: $\theta_1, \theta_2, \cdots, \theta_n$.

Thus, for a *n* DOF robot, the direct kinematic model can be achieved as described in Frame 1. Details about how to assign a transformation matrix to a robot's DOF can be found in Craig (2017).

Frame 1 – Steps to evaluate de direct kinematic model for a *n* DOF robot.

1. For each DOF **{i}** of the robot, using Eq. (1), determine its respective transformation matrix regarding to the previous coordinate system **{i-1}**. This transformation matrix will be dependent on θ_i.
2. Determine the transformation matrix between the inertial frame **{0}** and the end effector frame **{n}** using Eq. (5). This transformation will be dependent on $\theta_1, \theta_2, \cdots, \theta_n$.
3. For specific values of the rotational joints, determine the orientation and position of the robot's tool.

Inverse kinematics will require a more complex procedure to determine the angles $\theta_1, \theta_2, \cdots, \theta_n$ for a desired position/orientation of the robot's end effector. Thus, simulate the inverse kinematic model may be an interesting strategy to acquire knowledge about the robot's behavior.

One way to simulate the inverse kinematics of a robot without the explicit definition of this model is the use of GAs. These algorithms have great vocation for the solution of optimization problems, as this model can be treated. In this way, a GA can be built in a way that, given an initial estimate for the values of $\theta_1, \theta_2, \cdots, \theta_n$ and a target function (get as close as possible to the position/orientation desired for the robot's tool), this estimate can be refined towards an optimal solution. By means of this strategy, in a very simple way, the GA can be initially fed by a random estimate for the values of $\theta_1, \theta_2, \cdots, \theta_n$, evolving this solution until a certain condition of minimum position/orientation error is achieved.

This way, through the use of GAs, it is possible to determinate the inverse kinematics of a robot without the formal specification of its model, allowing certain behaviors to be identified by the genetic algorithm. Thus, the robotics' study proposed here is driven to the use of GAs, as an alternative to conventional methods for the determination of the inverse kinematic model of a manipulator robot.

FUNDAMENTALS OF GENETIC ALGORITHMS

The use of genetic algorithms in optimization problems was initially proposed in Holland (1975), being popularized through Goldenberg (1989) and Haupt (2004). Briefly, it can be said that GAs are an analogy to Charles Darwin's Theory of Evolution of Species (Darwin 2009), which, in turn, began with the integration of concepts between natural selection and genetics carried out by Gregor Mendel (Miller 2009). In summary, in a computational environment, we aim to search for the evolution of a given solution to a problem, from an initial estimate, possibly rough, to an optimal one. To do so, the optimization process requires a search space, formed by "individuals" of a "population", where the optimal solution is sought for the studied case, as well as an objective function, which leads to the pursuit towards the best possible solution (Bing 2016; Gupta 2016; Kramer 2017).

In this context, the use of GAs implies a stochastic process, where possible solutions are grouped into a population, being all of which evaluated simultaneously, with higher scores attributed to the best individuals, i.e., to the best solutions. Thus, possible solutions to the problem are treated as individuals within a population of solutions.

In turn, the evolution of the population towards optimized solutions passes through events where individuals combine with each other, in "crossover" processes, or suffer "mutations", similarly to what occurs in biological populations. Such evolution gives rise to new generations that should represent better solutions to the problem addressed.

The binary genetic algorithm

The computational implementation of a GA is relatively simple, but it is important to code the individuals of a population in a binary representation, for the proper application of the algorithm proposed here. Thus, from the initial population of individuals that constitutes a space of search towards the best solution of the problem, as long as a certain evolutionary criterion is not reached, the steps presented in Frame 2 must be repeated.

Frame 2 – Basic steps of a GA.

1. Each of the individuals of the population is evaluated, assigning to the same grades that represent their respective "fitness" for the solution of the treated problem. Such grades are obtained from the objective function, which represents the north for the best solution. The higher the score of an individual, the closer it is to the optimal solution.
2. The best individuals of the population are selected, so that they can be combined in pairs determined by sortition, in a process called "crossover". Through these crossings, individuals, in pairs, exchange part of their bits, giving rise to a new generation for the population.
3. Each individual of the new generation is subjected to an eventual "mutation". In this process, bits of a particular individual can change their value, upon occurrence of a low probability event.
4. "Elitism" is applied in the new generation. This implies bringing the best individuals of the current generation to the new one, thus preserving the best solution obtained so far.
5. This procedure is repeated again from the initial step until the expected evolutionary criterion is met, that is, the population or one of its individuals reaches the limits of the optimal solution.

The fitness of an individual *x_i* of the population can be represented by a function *f(x_i)*, which indicates how close this individual is to the optimal solution to the studied problem. Thus, in a population composed by *N* individuals, each of them will have its own fitness defined through *f(x)*.

The analysis and comparison of the fitness of the individuals from a population will establish the probability $p(x_i)$ that an individual *i* will have to generate descendants, through the crossover process. In the case where this probability is directly proportional to the numerical value of *f(x)*, then it can be calculated by equation (6).

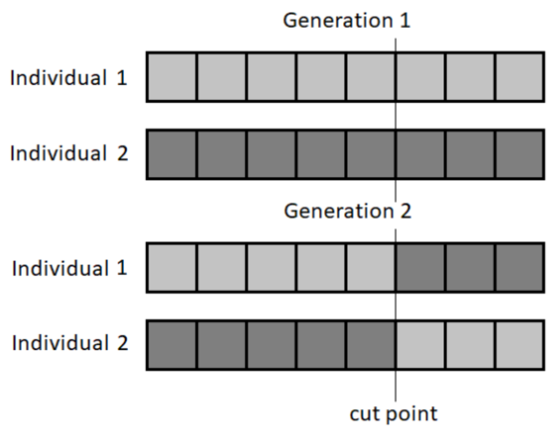
$$p(x_i) = \frac{f(x_i)}{\sum_{k=1}^{N} f(x_k)} \qquad (6)$$

If the probability increases as the numerical value of the objective function tends to zero, as is the case of the model discussed in this paper, then the probability of selecting an individual shall be calculated as shown in equation (7).

$$p(x_i) = \frac{1-\left(\frac{f(x_i)}{\sum_{k=1}^{N} f(x_k)}\right)}{(N-1)} \qquad (7)$$

The crossover procedure, to which individuals with better fitness will be subjected, can be understood from Figure 2.

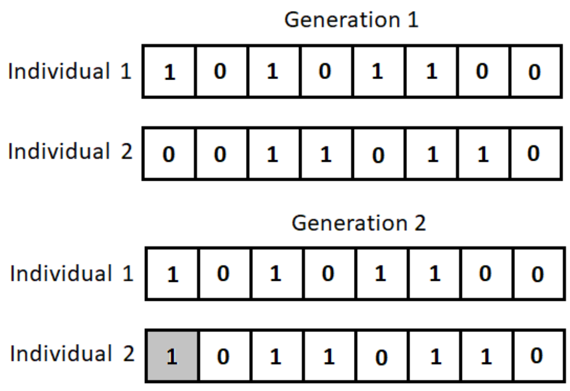
Figure 2 – Crossover procedure between two individuals of a population.



Prior to the crossover, a "cut point" must be defined at random, which will indicate the region to be exchanged between the two individuals. Through Figure 2 it can be observed that, from the cut point, there is the exchange of information between the pair.

In turn, the mutation procedure is represented by Figure 3.

Figure 3 – Mutation occurred in one of the individuals of the population.



Mutation is a random event of low probability, which may occur to reverse the value of one or more bits of individuals in a given population. When applying the mutation procedure in a GA, care must be taken to do not make this process an event with high frequency, which could cause degeneration of the solution represented by the group.

In an GA, "elitism" aims to preserve the best characteristics of the current generation, transporting it to the next generation. Specifically, the fittest individual (or those who are most fit) passes directly from the current generation to the next generation, without undergoing any modifications.

GENETIC ALGORITHMS AND THE KINEMATICS OF A ROBOT

This section describes the adopted procedure regarding the use of a GA to solve the inverse kinematic problem of a generic robot. The solution to the model described at the beginning of this paper requires the determination of joint angles, $\theta_1, \theta_2, \ldots, \theta_n$ which satisfy the positioning of the free end of the robot at a certain point and orientation in 3D space.

At first, consider a desired position, $\boldsymbol{p_d}$, and orientation, ${}^0_n\boldsymbol{R_d}$, for the end effector of the robot. According to equations (2) and (3), these can be respectively described as the 3D vector in equation (8) and the rotation matrix in equation (9).

$$\boldsymbol{p_d}(\theta_1, \ldots, \theta_n) = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (8)$$

$${}^0_n\boldsymbol{R_d}(\theta_1, \ldots, \theta_n) = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \quad (9)$$

Calculation through a GA leads to an approximation for desired position and orientation. Consider, in equations (10) and (11) respectively, the calculated approximations for position and orientation.

$$\boldsymbol{p'_d}(\theta_1, \ldots, \theta_n) = \begin{bmatrix} p'_x \\ p'_y \\ p'_z \end{bmatrix} \quad (10)$$

$${}^0_n\boldsymbol{R'_d}(\theta_1, \ldots, \theta_n) = \begin{bmatrix} n'_x & o'_x & a'_x \\ n'_y & o'_y & a'_y \\ n'_z & o'_z & a'_z \end{bmatrix} \quad (11)$$

From equations (8) and (10), the Euclidean distance between the desired and the calculated position can be presented as in (12).

$$d_p = \sqrt{(p_x - p'_x)^2 + (p_y - p'_y)^2 + (p_z - p'_z)^2} \quad (12)$$

In turn, considering equations (9) and (11), it is possible to determine the Frobenius norm for the difference between ${}^0_n\boldsymbol{R}_d$ and ${}^0_n\boldsymbol{R}'_d$. This norm is presented in Equation (13).

$$d_R = \sqrt{trace\ (({}^0_n\boldsymbol{R}_d - {}^0_n\boldsymbol{R}'_d) * ({}^0_n\boldsymbol{R}_d - {}^0_n\boldsymbol{R}'_d)^T} \quad (13)$$

Finally, equations (12) and (13) will be used to determine the objective function, which will be applied in the optimization process, as presented in equation (14).

$$d = \max{(d_R, d_p)} \quad (14)$$

The value of d calculated in (14) represents a measure of "distance" between desired and calculated position/orientation. Specifically, the considered distance will be the higher value between d_R and d_p. According to the purpose of the algorithm, the lower the value of d, the closer the calculated solution will be to the desired position/orientation.

For the implementation of the GA, as previously described, it is necessary to code the variables θ_1 to θ_n in a binary format. Considering that the values of these angles will be constrained to the interval between 0 and 2π, we opted for a codification where the three most significant bits are reserved to the integer part of the angle, and the other bits are reserved to the fractional part.

The computational implementation of the GA also requires that an individual be represented by the following data structure:

```
Structure Individual
{
Real θ1;
Real θ2;
…
Real θn;
Real fitness;
Real selection probability;
};
```

In this structure, $\theta_1 \dots \theta_n$ define the solution represented by the individual; "fitness" synthesizes the grade assigned to this solution and "selection probability" represents the probability of the individual being selected for the crossover procedure.

Fitness will be calculated by equation (14), being remembered that, because it represents a distance, the less its numerical value, the greater will be the fitness of the individual. This implies the use of equation (7) for the calculation of the selection probability of an individual.

In turn, a population with *N* individuals will describe the space for the search and evolution towards the optimal solution. This population is described as a vector with *N* individuals in the computational implementation of this algorithm.

```
Individual Current_Generation[N];
```

Thus, from an initial generation with *N* individuals, the algorithm in Frame 3 can be used to determine future generations, until an optimal solution is reached.

Frame 3 – GA for the solution of the inverse kinematics of the planar robot.

```
1. Select, at random, N individuals for the first generation of the population.
   If there are individuals whose values of $\Theta_1 \dots \Theta_n$ are outside the permitted limits ($0 \leq \Theta < 2\pi$), replace these individuals for others.
2. Calculate the fitness of each individual, through equations (8) to (14).
3. Calculate the selection probability of each individual, through Equation (7).
4. Find the individual with higher fitness in this generation.
5. While the higher fitness does not meet the criterion of stop:
6. To start a new generation, repeat N/2 times:
7. Select, by sortition, based on the selection probability, two individuals of the current generation.
8. Perform the crossover of the two selected individuals.
9. Store the two individuals generated by crossover in the new generation.
   End of repeat for step 6.
10. Submit all individuals of the new generation to an eventual mutation process.
12. If in the new generation there are individuals whose values of $\Theta_1 \dots \Theta_n$ are outside the
```

```
permitted limits ($0 \leq \theta < 2\pi$), replace these individuals for the fittest of the current generation.
13. Select the less fit individual of the new generation and replace it with the fittest individual of the current generation.
14. Calculate the fitness of each individual of the new generation.
15. Calculate the selection probability of each individual of the new generation.
16. Select the fittest individual of the new generation.
17. Make the new generation be the current generation.
    End of while for step 5.
    End of the algorithm.
```

Some points of this algorithm are highlighted at next. A considerable advantage of this is the fact that it does not require a predetermined solution so that it can evolve towards an optimal solution. In this way, Step (1) of the algorithm allows us to create a random initial generation as a starting point. As the only constraint, consider that the values of θ_1 to θ_n of this first generation must be within the allowed range ($0 \leq \theta < 2\pi$).

Step (2) calculates the individual fitness from the values of θ_1 to θ_n. Through equations (8) to (14) the distance between the desired and the obtained position/orientation will be evaluated. This distance represents the fitness of the individual, being better the lower is its numerical value.

The probability of selecting a particular individual for crossover, pointed in Step (3), derives from its fitness, that is, from the distance that its respective solution represents. This probability, calculated through Equation (7), takes into account the fitness of the other individuals, having a greater probability of selection the one which is at a shorter distance from the desired position/orientation. Given its relative character, the sum of the probabilities of each individual should be equal to 1.

Finding the individual with the highest fitness, as provided in Step (4), allows us to verify how far the simulation is from its stopping criterion. This criterion, pointed out in Step (5), can be adjusted in several ways. In the simulation developed in this work, it was chosen to establish as a criterion of stopping for the evolution of the population the highest individual fitness, with numerical value inferior to a pre-established limit. Other stop criteria to be considered may be a given number of generations, loss of diversity of a population or convergence to a given solution after a certain number of generations (Holland 1975).

The selection of individuals for crossover by sortition, indicated in Step (7), can be interpreted through the format of a "lottery", based on the fitness and on the probability of an individual being selected. In this model of sortition, a set of "lottery tickets" is distributed to each individual of the population, which is proportional to its probability of selection. Thus, through the draw of the "winning ticket", it is defined the individual ("owner of the ticket") that will be selected for the crossover.

The crossover, indicated in Step (8), is the event that will trigger the birth of a new generation, different from the current. Moreover, this new generation may pass through a mutation process, as provided in step (10), increasing the diversity of the population. As previously indicated, mutation is an event that should be used with caution in GAs, since high mutation rates may lead to degeneration of the population and, therefore, the loss of the solution that it represents.

Crossover and/or mutations are procedures that may eventually give rise to "degenerate" individuals, that is, whose values of θ_1 to θ_n are outside the permitted limits. This kind of occurrence may be circumvented through the replacement of the degenerate individuals by the fittest individuals from the current generation (Step 12). In this same sense, with or without degenerate individuals, it is convenient to preserve the fittest individual of the current generation (elitism). To do so, Step (13) proposes to replace the less fit individual of the new generation with the fittest of the present generation.

Once the new generation is defined, the fitness and the selection probability of each individual must be recalculated. Finally, the new generation is made the current generation and, if the stop criterion has not been reached, the procedures are repeated for the creation of another generation.

RESULTS AND DISCUSSION

The model presented throughout this text was simulated for a PUMA 560 robot, with 6 DOF, using the structural parameters described in Frame 4.

Frame 4 – PUMA 560 parameters for the simulation.

Param-eter	Value (mm)	Description
A_2	431.80	Length of the arm
A_3	20.32	Offset between elbow and wrist
D_3	149.09	Offset between shoulder and elbow
D_4	433.07	Length of the forearm

According to Craig (2017), the link transformations for this robot are described in equations (15.a) to (15.f).

$$ {}^{0}_{1}\boldsymbol{T} = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15.a)$$

$$ {}^{1}_{2}\boldsymbol{T} = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15.b)$$

$$ {}^{2}_{3}\boldsymbol{T} = \begin{bmatrix} c_3 & -s_3 & 0 & A_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & D_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15.c)$$

$$ {}^{3}_{4}\boldsymbol{T} = \begin{bmatrix} c_4 & -s_4 & 0 & A_3 \\ 0 & 0 & 1 & D_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15.d)$$

$$ {}^{4}_{5}\boldsymbol{T} = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15.e)$$

$$ {}^{5}_{6}\boldsymbol{T} = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15.f)$$

Where s_1 stands for $\sin(\theta_1)$, c_1 stands for $\cos(\theta_1)$ and so on.

Thus, the transformation that relates the rotational joint in the end effector and the inertial reference frame, ${}^{0}_{6}\boldsymbol{T}$, can be calculate through equation (15.g).

$$ {}^{0}_{6}\boldsymbol{T} = {}^{0}_{1}\boldsymbol{T}.{}^{1}_{2}\boldsymbol{T}.{}^{2}_{3}\boldsymbol{T}.{}^{3}_{4}\boldsymbol{T}.{}^{4}_{5}\boldsymbol{T}.{}^{5}_{6}\boldsymbol{T} \quad (15.g)$$

The executed simulation consists in a translation/rotation starting from ${}^{0}_{6}\boldsymbol{T_s}$ to ${}^{0}_{6}\boldsymbol{T_f}$ as described in equations (16) and (17).

$$ {}^{0}_{6}\boldsymbol{T_s} = \begin{bmatrix} 1 & 0 & 0 & 300 \\ 0 & 1 & 0 & 300 \\ 0 & 0 & 1 & 100 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

$$ {}^{0}_{6}\boldsymbol{T_f} = \begin{bmatrix} 0.6428 & -0.7660 & 0 & 250 \\ 0.7660 & 0.6428 & 0 & 250 \\ 0.00 & 0.00 & 1 & 50 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

And for each step in this trajectory the GA must calculate the values for the joint angles $\theta_1,\ \theta_2, \theta_3, \theta_4, \theta_5\ and\ \theta_6$.

The angles θ_1 to θ_6, which define the solution represented by an individual, were encoded in 24 bits, being 3 bits dedicated to the representation of the integer part of the angle and 21 bits destined to the representation of the fractional part. It should be stressed that the angles are restricted to the interval $0 \leq \theta < 2\pi$.

In the developed implementation, a population of 40 individuals was used. The cut point for the crossover is set at random, each time this operation is performed, being limited to a maximum of 87,5% of the bits, counted from the least significant bit. In turn, the probability of mutation was limited to 2%, in order to avoid population degeneration.

The stop criterion used to finish the calculation of the angles θ_1 to θ_6 associated with a position/orientation consists in obtaining a distance "*d*" of less than 0.001 units (Equation 14). It must be stated that there are eight possible solutions available for the PUMA 560. Figures 4 to 6 present three solutions obtained with GA simulation.

Figure 4 – Comparation between algebraic solution and GA simulation for PUMA 560 (first solution).

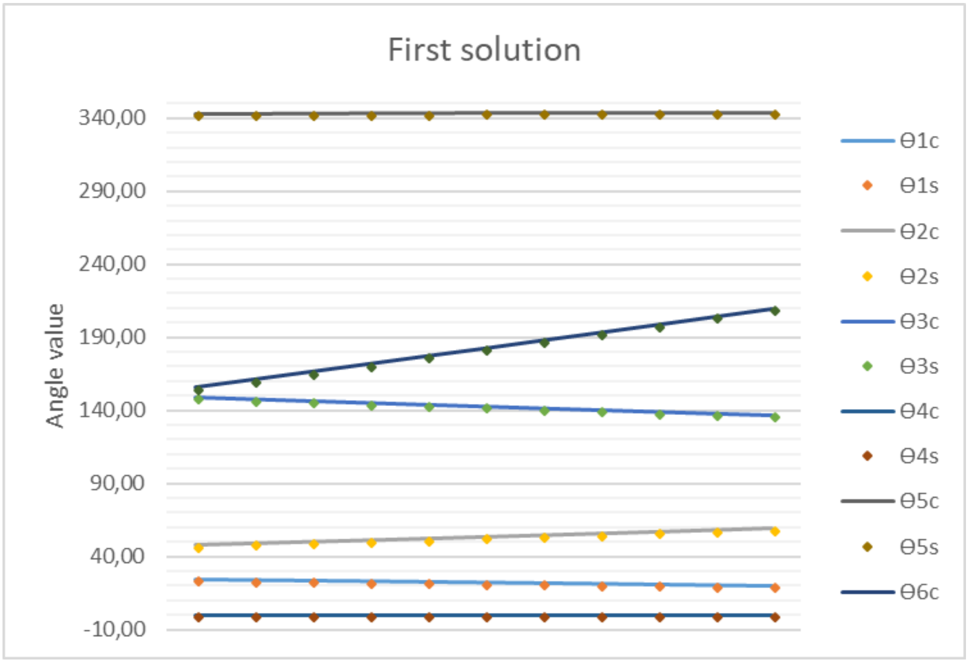


Figure 5 – Comparation between algebraic solution and GA simulation for PUMA 560 (second solution).

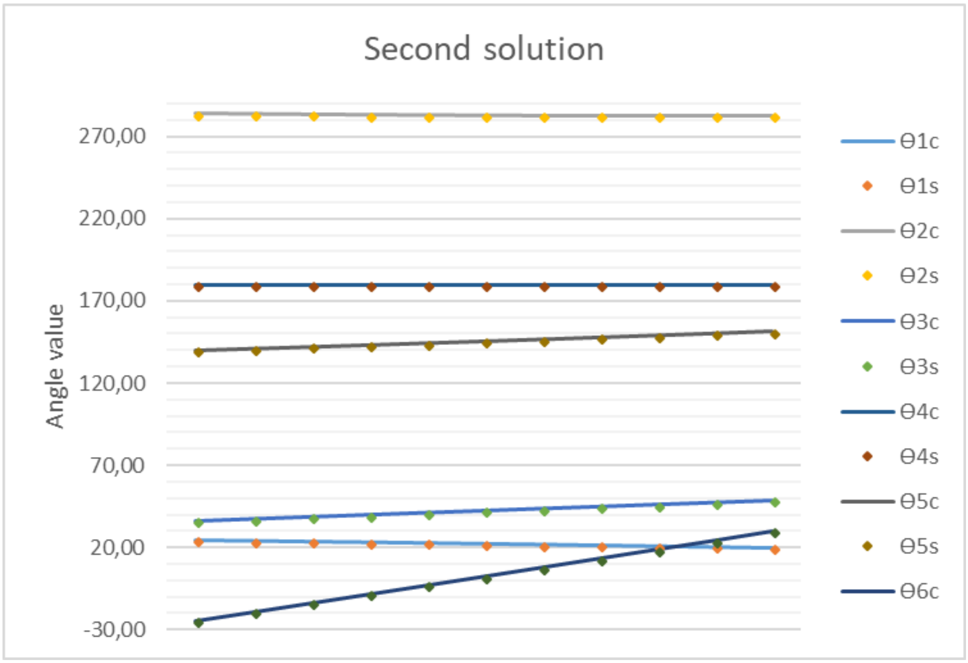
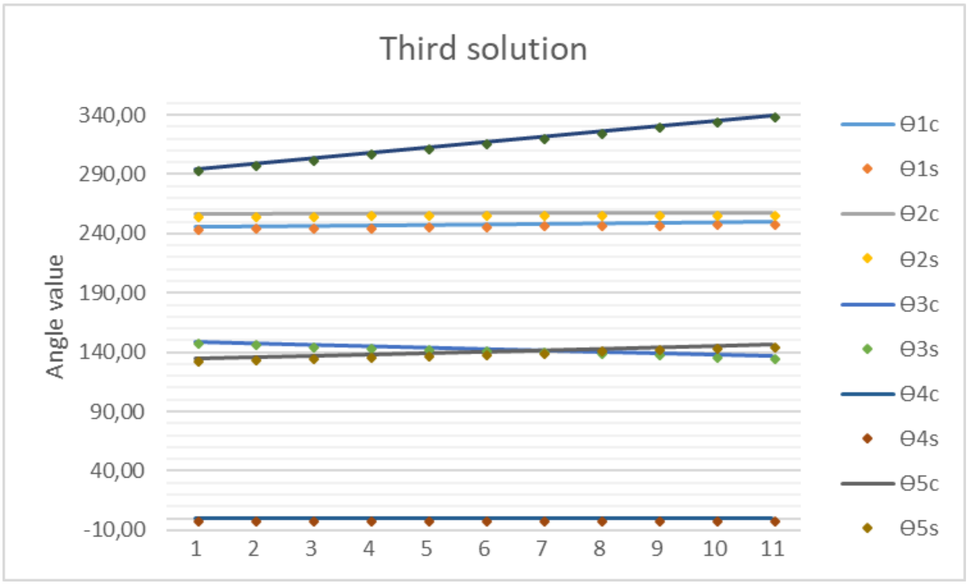


Figure 6 – Comparation between algebraic solution and GA simulation for PUMA 560 (third solution).



In figures 4 to 6, solid lines ($\theta_{1c}, \theta_{2c}, \theta_{3c}, \theta_{4c}, \theta_{5c}\ and\ \theta_{6c}$) represent the algebraic solution for the inverse kinematic model, as presented in Craig (2017). In turn, dots sequences ($\theta_{1s}, \theta_{2s}, \theta_{3s}, \theta_{4s}, \theta_{5s}\ and\ \theta_{6s}$) represent the simulation obtained through GA.

Tables 1 to 3 present the numerical values for figures 4 to 6.

Table 1 – Comparation between algebraic solution and GA simulation for PUMA 560 (first solution).

First solution											
Θ_{1c} (˚)	Θ_{1s} (˚)	Θ_{2c} (˚)	Θ_{2s} (˚)	Θ_{3c} (˚)	Θ_{3s} (˚)	Θ_{4c} (˚)	Θ_{4s} (˚)	Θ_{5c} (˚)	Θ_{5s} (˚)	Θ_{6c} (˚)	Θ_{6s} (˚)
24,42	24,42	47,83	47,83	149,19	149,19	0,00	0,04	342,98	343,00	155,57	155,50
24,06	24,06	48,91	48,91	147,91	147,91	0,00	0,02	343,18	343,18	160,94	160,92
23,68	23,68	50,00	50,00	146,64	146,64	0,00	0,02	343,37	343,38	166,31	166,29
23,29	23,29	51,10	51,10	145,37	145,37	0,00	0,05	343,53	343,55	171,71	171,66
22,88	22,88	52,22	52,22	144,11	144,11	0,00	0,03	343,66	343,67	177,11	177,11
22,45	22,45	53,36	53,36	142,87	142,87	0,00	0,07	343,77	343,77	182,40	182,44
22,01	22,01	54,52	54,52	141,62	141,62	0,00	0,00	343,85	343,89	187,95	188,00
21,55	21,55	55,70	55,70	140,39	140,39	0,00	0,08	343,90	343,88	193,32	193,36
21,07	21,07	56,91	56,91	139,16	139,16	0,00	0,06	343,93	343,94	198,78	198,80
20,58	20,58	58,14	58,14	137,94	137,94	0,00	0,00	343,92	343,92	204,33	204,34
20,05	20,05	59,40	59,40	136,73	136,73	0,00	0,08	343,87	343,88	209,79	209,82

Table 2 – Comparation between algebraic solution and GA simulation for PUMA 560 (second solution).

Θ_{1c} (˚)	Θ_{1s} (˚)	Θ_{2c} (˚)	Θ_{2s} (˚)	Θ_{3c} (˚)	Θ_{3s} (˚)	Θ_{4c} (˚)	Θ_{4s} (˚)	Θ_{5c} (˚)	Θ_{5s} (˚)	Θ_{6c} (˚)	Θ_{6s} (˚)
24,43	24,43	283,90	283,90	36,18	36,18	180,00	179,94	140,09	140,08	-24,43	-24,48
24,06	24,06	283,69	283,69	37,47	37,46	180,00	179,94	141,15	141,17	-19,06	-19,10
23,68	23,68	283,49	283,49	38,74	38,74	180,00	179,98	142,23	142,26	-13,69	-13,68
23,29	23,29	283,32	283,32	40,00	40,00	180,00	179,99	143,32	143,33	-8,29	-8,31
22,88	22,88	283,17	283,17	41,26	41,26	180,00	180,03	144,43	144,43	-2,89	-2,84
22,46	22,46	283,05	283,05	42,51	42,51	180,00	180,02	145,56	145,59	2,54	2,56
22,02	22,02	282,95	282,95	43,75	43,75	180,00	179,99	146,70	146,72	7,98	7,97
21,56	21,56	282,89	282,88	44,98	44,98	180,00	180,00	147,87	147,86	13,44	13,47
21,08	21,08	282,85	282,85	46,21	46,21	180,00	179,93	149,06	149,06	18,92	18,86
20,58	20,58	282,84	282,84	47,43	47,43	180,00	180,00	150,27	150,27	24,32	24,29
20,06	20,06	282,87	282,87	48,65	48,65	180,00	180,00	151,52	151,54	29,94	29,91

Table 3 – Comparation between algebraic solution and GA simulation for PUMA 560 (third solution).

Third solution											
Θ_{1c} (°)	Θ_{1s} (°)	Θ_{2c} (°)	Θ_{2s} (°)	Θ_{3c} (°)	Θ_{3s} (°)	Θ_{4c} (°)	Θ_{4s} (°)	Θ_{5c} (°)	Θ_{5s} (°)	Θ_{6c} (°)	Θ_{6s} (°)
245,57	245,57	256,10	256,10	149,19	149,19	0,00	0,05	134,72	134,72	294,43	294,45
245,94	245,94	256,31	256,31	147,91	147,91	0,00	0,05	135,78	135,77	299,06	299,08
246,32	246,32	256,51	256,51	146,64	146,64	0,00	0,06	136,86	136,85	303,68	303,72
246,71	246,71	256,68	256,68	145,37	145,37	0,00	0,02	137,95	137,98	308,29	308,31
247,12	247,12	256,83	256,83	144,11	144,11	0,00	0,04	139,06	139,03	312,88	312,90
247,54	247,54	256,95	256,95	142,87	142,87	0,00	0,03	140,18	140,15	317,46	317,49
247,98	247,98	257,05	257,05	141,62	141,62	0,00	0,00	141,33	141,24	322,02	321,99
248,44	248,44	257,11	257,12	140,39	140,39	0,00	0,03	142,50	142,49	326,56	326,57
248,92	248,92	257,15	257,15	139,16	139,16	0,00	0,01	143,69	143,69	331,08	331,10
249,42	249,42	257,16	257,16	137,94	137,94	0,00	0,01	144,90	144,90	335,48	335,50
249,94	249,94	257,13	257,13	136,73	136,73	0,00	0,02	146,14	146,13	340,06	340,06

Regarding the simulation itself, the presented results were obtained by predicting, in the algorithm, the presence of crossover, mutations and elitism. However, for testing purposes, the simulation was also performed using only crossover and elitism and only mutations and elitism. These variations led to some empirical observations, which are presented bellow.

- When applied only crossover and elitism, without mutations, it was noted that the algorithm hardly converged toward the stopping criterion. Such an occurrence can be explained by the fact that the crossover used reached only the least significant part of the bits and did not allow the exchange of these in a ratio higher than 87,5%. Thus, with the three most significant bits preserved, the algorithm tends not to converge. Such a situation can be circumvented if a crossover with a higher rate and more than one cut point is used.
- When used mutation and elitism, without crossover, it was noticed that the algorithm presented very slow convergence, but the stopping criterion was achieved most of the time. This occurrence can be explained by the fact that mutation, although a phenomenon of low probability, can occur in all bits of the individual. Thus, the occurrence of mutations in the most significant bits of the individual, associated with elitism, allowed the algorithm to show convergence.
- The combination of crossover, mutation and elitism causes the algorithm to present its best results, with convergence occurring, most of the times, after a few number of iterations.

CONCLUSIONS

The results of this study demonstrate that the use of genetic algorithms for the solution of the inverse kinematic problem of robotics is feasible, especially in situations where the explicit determination of the inverse model is costly. A considerable advantage for the use of a GA in this type of problem is its relatively simple computational implementation. The presented algorithm is also parallelizable, being able to be fragmented in a cluster of computers, reducing the calculation time to obtain more precise solutions. Regarding the precision of the algorithm, it depends heavily on the number of digits used for the binary encoding of an individual. Another significant advantage in the use of GAs in optimization problems is their high flexibility, which makes them easily adaptable to other robot models or even other types of optimization problems.

REFERENCES

Bing, X., Mengjie, Z., Browne, W. N. (2016). A Survey on Evolutionary Computation Approaches to Feature Selection. **IEEE Transactions on Evolutionary Computation**, (20)4, 606-626.

Craig, J. J. (2017). **Introduction to Robotics***: Mechanics and Control*. 4th. Ed. New York: Pearson.

Darwin, C. (2009). **On the Origin of Species***:* By Means of Natural Selection. New York: Cambridge University Press.

Goldber, D. E. (1989). **Genetic Algorithms in Search, Optimization and Machine Learning**. Boston: Addison-Wesley Longman Publishing Co.

Gupta, A., Yew-Soon, O., Liang, F. (2016). Multifactorial Evolution: Toward Evolutionary Multitasking. **IEEE Transactions on Evolutionary Computation**, (20)3, 343-357.

Haupt, R. L. & Haupt, S. E. (2004). **Practical Genetic Algorithms.** 2nd. Ed. New Jersey: John Wiley & Sons.

Holland, J. H. (1975) **Adaptation in natural and artificial systems:** An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. Ann Arbor: The University of Michigan Press.

Kramer, O. (2017) **Genetic Algorithm Essentials**. New York: Springer.

Miller, F. P., Vandome, A. F. & McBrewster, J. (2009). **Gregor Mendel.** London: Alphascript Publishing.

Miller, R. M. & Miller, R. (2017). **Robots and Robotics:** Principles, Systems, and Industrial

Applications. USA: Mc Graw Hill Education.

AUTHORS' INFORMATION

Jose Tarcisio Frvanco de Camargo has degrees in Electrical Engineering (State University of Campinas, UNICAMP, Brazil, 1989) and Education (UNINTER, Brazil, 2015), M.Sc. Degree in Electrical Engineering (UNICAMP, 1992) and Ph.D. in Electrical Engineering (UNICAMP, 1995). He works as university professor since 1990, currently being professor and researcher at the Regional University Center of Espirito Santo do Pinhal (UNIPINHAL, Brazil). His research interests cover the areas of computer graphics, robotics and new technologies applied to education. Contact information: Av. Helio Vergueiro Leite, s/n, E. S. do Pinhal, SP, 13.990-000, Brazil; phone: +55(19)3651-9600, e-mail: jtfc@bol.com.br .

Eliana Anunciato Franco de Camargo has Degree in Biology (FIMI, Brazil, 1992) and Pedagogy (FIA, Brazil, 2000), Marketing (UNOPAR, Brazil, 2018), Specialization degree in Health Teaching (Federal University of Rio Grande do Sul, UFRGS, Brazil, 2015), Specialization Degree in Distance Education (UNOPAR, Brazil, 2016), M.Sc. Degree in Parasitology (UNICAMP, 1997) and Ph.D. degree in Animal Biology in the area of Anthropogenic Relations, Environment and Parasitology (UNICAMP, 2016). She works as teacher since 1992 and currently teaches and researches at the UNIPINHAL. Her interests are in Parasitology, Entomology and Malacology with emphasis in Parasites and Vectors, and in the area of Learning Education and Distance Education. E-mail: eafcamargo@yahoo.com.br.

Estefano Vizconde Veraszto has Degree in Physics (UNICAMP, 2001) and Ph.D. in Education, Science and Technology (UNICAMP, 2009). He also made a supervised Ph.D. internship at Complutense University of Madrid, Spain, 2009. He is currently a professor at the Federal University of Sao Carlos, UFSCar (since 2013), coordinator of postgraduate studies in Science and Mathematics Education (since 2016) and researcher at Technological Innovation Laboratory in Applied Education, UNICAMP (since 2003), and at the Research Group in Natural Sciences Education, UFSCar (since 2013). E-mail: estefanovv@gmail.com .

Gilmar Barreto received B.S. degree in Chemical Engineering from UNICAMP, in 1982; received M.S. and Doctorate degrees in Electrical Engineering, from UNICAMP, respectively in 1986 and 2002. He has experiences in Electrical Engineering with emphasis on Education Engineering, Industrial Electronics, Systems and Electronic Control: Fuzzy systems, multiple variables systems, control, multiobjective optimization, electrochemistry and Electrical engineering teaching. He is author of the book "Electrical Vehicles" within Professor Doctor Celso Pascoli Bottura. Currently he is a Professor at UNICAMP, School of Electrical and Computing Engineering - FEEC, Electrical Machines, Components and Intelligent Systems Department- DMCSI. E-mail: gbarreto@dsif.fee.unicamp.br

Jorge Candido is graduated in Electrical Industrial Engineering (1989) by the Federal Center for Technological Education of Paraná (CEFET-PR). Graduated in the Superior course of training of teachers of specialized disciplines (1992) by the Federal Center of Technological Education of Paraná (CEFET-PR), Master in Production Engineering (2001) by the Federal University of Santa Catarina (UFSC). PhD in electrical Engineering from the State University of Campinas (UNICAMP). Professor of installations and electrical machines in the technical course in electro-technical of CEFET-PR, currently professor of the disciplines the management, production and administration of UTFPR-CM. E-mail: jocandido@utfpr.edu.br .