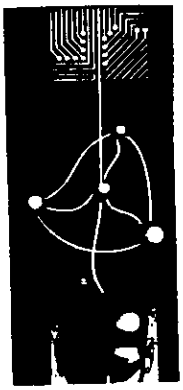

1° Congresso Brasileiro de Redes Neurais

Escola Federal de Engenharia de Itajubá
Itajubá. 24 a 27 de outubro de 1994

The Use of Genetic Algorithms for the Evaluation of Inverse Kinematics of Manipulators

Carlos Henrique da Silveira
(silveira@dcc.ufmg.br)

Lúcio de Souza Coelho
(omni@dcc.ufmg.br)

Mário Fernando Montenegro Campos
(mario@dcc.ufmg.br)

May, 1994

Departamento de Ciência da Computação
Instituto de Ciências Exatas
Universidade Federal de Minas Gerais
Caixa Postal 702 – Belo Horizonte – MG – Brazil – CEP 30.161-970

Abstract

This paper describes the implementation of a genetic algorithm system able to compute the inverse kinematics of a generic manipulator (robot). Given its geometric description and an homogeneous transformation describing the goal to be reached by its end-effector, the system computes the joint angle/displacement values with the desired precision. The method presented good performance in the case of a 6 degree of freedom manipulator, with a fast convergence to the specified values. An application to the case of redundant manipulators is also presented, where analytical solution by traditional methods would be of high complexity.

1 Introduction

The inverse kinematics problem can be stated as, given a position and an orientation in the workspace of a manipulator, find the set of joint values that will accomplish them. This problem is complex due to the multiplicity of solutions and to the difficulty to obtain, in the generic case, analytical solutions to the non-linear equation systems for six or more degrees of freedom manipulators [1].

The approach to non-linear equation system is, in general, a high-complexity problem. For this reason alternative methodologies have been widely researched in the field of Artificial Intelligence. Particularly Genetic Algorithms (GA) [2-7] have show to be an useful tool for some complex applications such as optimization and machine learning.

It is described here a system which has developed to generate the joint values, given the description of a manipulator in terms of its geometrical parameters and a final position and orientation of the end-effector. The values to be assumed for the joint variable parameters are computed so that the manipulator reaches the goal (or a transformation) specified by the user, with

a given precision

2 The Genetic Algorithm

2.1 Mapping Between Individuals and Solutions

In the GA solution for the inverse kinematics problem, individuals are sets of joint values. As strings of 1's an 0's, each of those sets is represented by the concatenation of substrings of equal length, each of them corresponding to a joint value; each substring is interpreted as a Gray Code number [1]; the value V of a given joint is obtained from

$$V = n\frac{M}{N} + I$$

where n is the integer number represented by the substring, M is the length of the range where the joint value can vary, N is the greatest number representable by a substring and I is the lower joint limit. For example, imagine a very simple manipulator composed of only three revolute joints, each of them able to rotate from 0^o to 180^o. Suppose that four bits are used to represent each joint and we want to find out to which configuration corresponds the individual

0010 1110 0101.

The first substring (from left to right) of four bits (0010) corresponds to the first joint, the second substring to the second and so on. Making the conversion for the second joint using the formula above and taking as parameters $n = 3$ (0010 in Gray Code), $M = 180^o$, $N = 15$ (the largest number represented with four bits) and $I = 0^o$, we find that $V = 36^o$. Computing V for the other joints and using the geometrical notation suggested by Denavit-Hartenberg [1], we find that individual corresponds to

$$\theta_1 = 36^o, \theta_2 = 132^o \text{ e } \theta_3 = 72^o.$$

[1] Gray code was selected to avoid disruptive mutations, as suggested in [2].

2.2 Fitness Evaluation

A fitness function is chosen such that the precision criterion selected is met. It is the following: let be A the homogeneous transformation describing the end-effector of a manipulator M under its frame 0 when the joints assume the values with respect to an individual x: let B be the homogeneous transformation that is desired to be assumed by the end-effector under the frame 0; making $C = A - B$ and calling i and j the coordinates of the C element with the greatest absolute value among all, its is said that x satisfy a precision p if $|c_{i,j}| \leq p$. Keeping in mind this definition, we use the fitness function:

$$f(x) = \frac{1}{|c_{i,j}|}$$

2.3 About the Reproduction and the Population

Individuals with the best fitness have a greater probability to reproduce in GA. In order to satisfy this paradigm, we choose the following distribution. Imagine a population with N individuals in decreasing order of fitness, that is, $f(n-1) \geq f(n) \geq f(n+1)$. To select an individual to reproduce among the population members, we use the function

$$s(r) = r \bmod (r \bmod N + 1) + 1$$

where $r \in \mathbb{N}$ is a random number. Let S_n be the event of the individual n to be selected to reproduce (that is, $s(r)$ returns n). It follows that

$$P(S_n) = \frac{1}{N}\sum_{i=n}^{N}\frac{1}{i}$$

The probability distribution is shown in Figure 1 for $N = 65$.

It should be pointed out that two individuals are always selected to mate. The first individual is straightforwardly chosen by $s(r)$. But, when selecting the second,

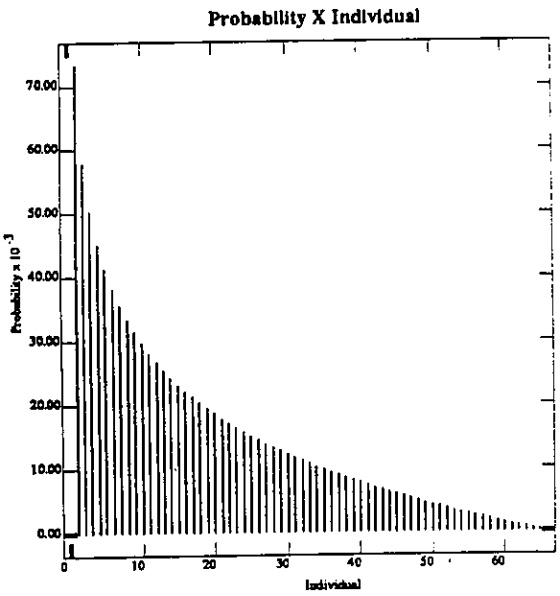

Figure 1: Probability distribution (vertical axis) *versus* individuals in decreasing order of fitness (horizontal axis).

it must avoided selecting the first individual. So, the first individual is skipped during the second selection, which implies that $s(r)$ works with a population smaller by one unit.

When mating two individuals, the two offspring generated by a crossover are placed in the next generation. Each offspring still experiences a mutation, which is implemented by the complement of a randomly selected bit.

The new generation grows until its size becomes equal to the size of the current generation, hence, the population held constant.

Finally, we choose to follow the elitist model as in [5]: the best fitness individual is cloned; a copy of it is always placed in the next generation. Thus, it is ensured that, in the running the GA, a "retreat" never occurs in the search for a satisfactory solution.

2.4 The Crossover Operator

When the character subsets, that are interchanged among two individuals by crossover, are equal, the offspring generated are only copies of their parents. To avoid this, the crossover operator used here first determines what can be called an *useful crossover region*: which delimited by the leftmost and rightmost different bits comparing two strings selected for reproduction. For example, consider the pair of individuals below, where the useful crossover region is shown underlined:

100110010010
and
100110101010.

Then, a crossover is made by selecting a point between the bits belonging to the useful crossover region, ensuring that the offspring will be different from its parents, as in

100110101**1010**
e
100110010010.

The useful crossover concept helped us to think what to do when the two individuals that are about to reproduce are equal or different only by one bit, in which case there is no useful crossover point. To avoid this situation one of the individuals will be forced to undergo 1 or 2 mutations, depending if they differ by 1 or 0 bits, respectively. Then, a useful crossover is made.

3 Results

3.1 Case Study I

In this section the speed of the current GA is analyzed statistically for the problem of the inverse kinematics of a manipulator.

A single manipulator is used, composed by three prismatic joints (perpendicular among them and responsible for the positioning of the manipulator) and three final revolute joints (responsible for the orientation of the end effector, with coincident origins and perpendicular rotation axes).The Denavit-Hartenberg parameters

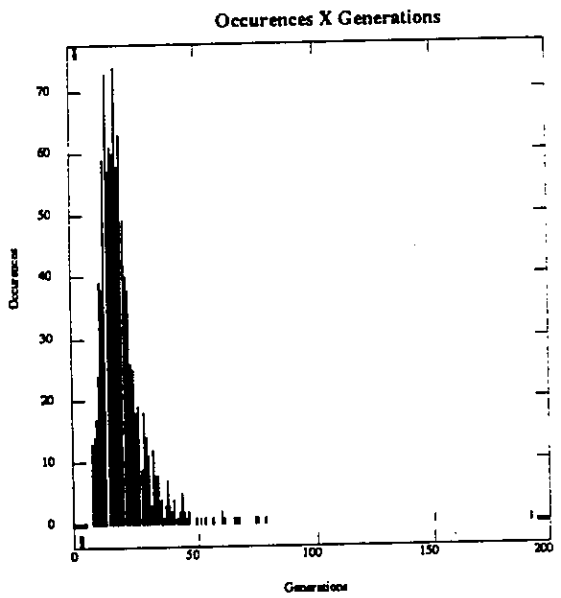

Figure 2: Number of occurrences (vertical axis) *versus* generations number (horizontal axis).

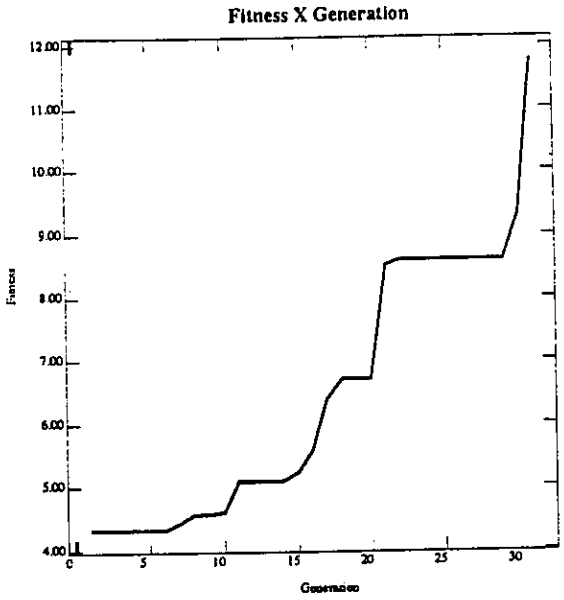

Figure 3: Best individual fitness (vertical axis) *versus* generation (horizontal axis) for the discovery of a solution to a six degree of freedom manipulator.

for the mentioned manipulator are shown in the table below:

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0°	0	d_1	0
2	90°	0	d_2	90°
3	−90°	0	d_3	0
4	0°	0	0	θ_4
5	−90°	0	0	θ_5
6	90°	0	0	θ_6

For simplicity, it was assumed that prismatic joints can vary their values continuously from 0 to 1 and revolute joints have no rotation limits. Finally, the transformation that describes the end-effector with reference to the frame of the last link is the identity matrix.

Wishing only to evaluate the speed of the algorithm, we work with a little precision of 0.1, and 6 bits were designated to represent each joint value. Therefore, our individuals were strings with length 36 over the alphabet $\{0, 1\}$. Population size was arbitrarily assigned to 65 individuals (1 individual cloned and 32 pairs generated by reproduction).

Each test was made executing the following steps:

1. Random values, within the variation bounds specified in manipulator description, are generated to the joints.

2. Using the direct kinematics [1], a transformation A, describing the end-effector with reference to the frame 0 when joints assume the values generated in Step 1, is obtained. Therefore, A surely is in the manipulator workspace.

3. It is asked to the algorithm to supply the joint values that reach A with the specified precision.

4. The result of the test is the number g of generations spent by the algorithm to reach the solution.

Data from executing 1000 tests is obtained. Figure 2 shows the histogram with the frequency distribution for each test result. The average number of generations was 19, with 5 and 193 as the minimum and the maximum, respectively. In order to give a notion of the real time spent.

each test took, in average, about one hundredth of a second of CPU time running on a SPARC2 workstation.

Figure 3 shows the fitness evolution through generations in one test.

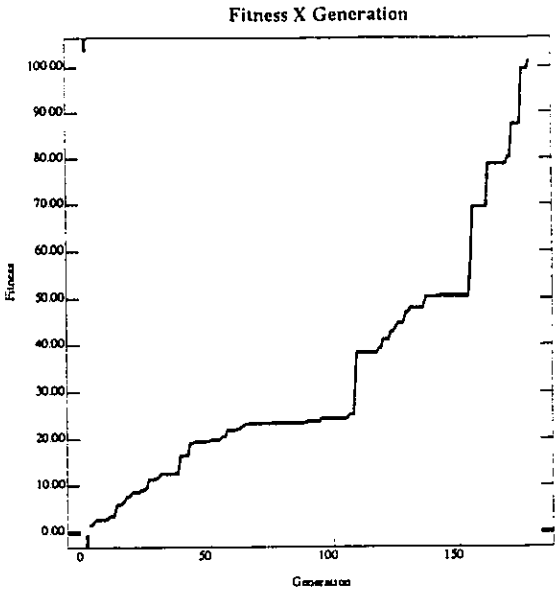

Figure 4: Best individual fitness (vertical axis) *versus* generation (horizontal axis) for the redundant manipulator.

3.2 Case Study II: a Manipulator with Redundant Degrees of Freedom

The greater the number of joints of a manipulator, the more difficult and laborious it is to obtain its inverse kinematics equations. Manipulators normally have only six joints, each one corresponding to a degree of freedom, which is the minimum needed to ensure a non-zero dextrous workspace. When a manipulator has more than six degrees of freedom, it is called redundant, because there are redundant degrees of freedom.

Even though it is hard to obtain analytically, the inverse kinematics of a redundant manipulator can be obtained automatically by the method presented here. That gives manipulator independent solutions for this problem.

In order to test this generality, it was defined a manipulator much more complex than the one of the previous section, having 10 degrees of freedom and only revolute joints. We assume that the odd link joints do not have bounds restricting its rotations and the even link joints can rotate in the range of 0° to 180°. The transformation that describes the end effector in relation to last link frame is the identity matrix. The manipulator is described below according to Denavit-Hartenberg notation:

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0.4	θ_1
2	-90^o	0	0	θ_2
3	-90^o	0	0.4	θ_3
4	-90^o	0	0	θ_4
5	-90^o	0	0.4	θ_5
6	-90^o	0	0	θ_6
7	-90^o	0	0.4	θ_7
8	-90^o	0	0	θ_8
9	-90^o	0	0.4	θ_9
10	-90^o	0	0	θ_{10}

The GA worked with a precision of 0.01 and used 10 bits to represent each joint value. A population of 65 individuals was used. The manipulator reaches the identity matrix, which means that the end-effector frame reached the position and orientation of the base frame within the required precision. The matrix below was produced as the solution

$$\begin{bmatrix} 1.000 & -0.004 & 0.005 & 0.005 \\ 0.004 & 1.000 & 0.008 & 0.010 \\ -0.005 & -0.008 & 1.000 & 0.009 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$$

in 179 generations. The fitness evolution of the best individual during generations is shown in Figure 4.

4 Conclusions

The method presented here worked for the tested cases, functioning as a generic evaluator of inverse kinematics. Although, we consider the tests very simple and, hoping to know more about the behavior of this GA, it would be convenient pursue the following directions: verification of bit convergence, variation of population size, variation of crossover rate, variation of mutation rate, a study of the relations between the bit number per joint and precision and the use of other probability distributions related to individual selection for reproduction.

5 References

1 Craig, John J., **Introduction to robotics: mechanics and control.** 2nd edition, Addison-Wesley Publishing Company, Inc.

2 Forrest, Stephanie. **Genetic Algorithms: Principles of Natural Selection Applied to Computation.** in *Science*. Vol. 261, 13 Aug. 1993. pp. 872-8.

3 Goldberg, David E.. **The Genetic Algorithm Approach: Why, How, and What Next?**, in *Adaptative and Learning Systems - Theory and Applications*, Plenum Press, New York and London, pp. 247-53.

4 Holland, John H.. **Genetic Algorithms**, in *Scientific American*. July 1992, pp. 44-50.

5 McGarrah, D.B. e Judson, R.S.. **Analysis of the Genetic Algorithm Method of Molecular Conformation Determination.** in *Journal of Computational Chemistry*, Vol. 14, 11, 1993, pp. 1385-95.

6 Unger, Ron e Moult, John. **Genetic Algorithms for Protein Folding Simulations.** in *Journal of Molecular Biology*, 231, pp. 75-81.

7 Wayner, Peter. **Genetic Algorithms.** in *Byte*. January 1991. pp. 361-8.