Genetic Algorithm for Walking Robots Motion Optimization

DANUT A. BUCUR, STEFAN A. DUMITRU
Robotics and Mechatronics Department,
Institute of Solid Mechanics of the Romanian Academy
15 C-tin Mille, Sector 1, Bucharest,
ROMANIA
dan.bucur@ymail.com, dumitru_sa@yahoo.com

Abstract: The paper presents a method to determine the optimal solution regarding the movement done by the leg of a hexapod walking robot with three degrees of freedom, with higher positioning precision taking into account the importance of three factors (Precision, Movement and Friction). To find the coefficients that establish the order of these factors we use an Analytical Hierarchy Process. With these coefficients we build the fitness function for a genetic algorithm that is slightly modified (instead of two chromosomes crossed, we crossed three) in order to obtain a diversity in the exchanged information. The obtained results prove that, applying the genetic algorithm in correlation with the analytic hierarchy method of processing, we can optimize the precision in the robot movement with a relative small increase of the consumed energy.

Key-Words: Genetic Algorithm, Analytical Hierarchy Process, Robot Motion Control, Real-time Control

1 Introduction

The movement represents the basic function of mobile robots. They are divided into two main categories: wheeled mobile robots and walking mobile robots. We focus on the second category, and we want to study the movement of the leg during a stepping sequence.

Walking robots have a wide range of applicability due to feet's ability to adapt to land forms. Thus, walking robots can be used successfully in areas where the environment is dangerous for humans - plants, nuclear plants, military, etc. Because of the importance of areas where they are used, researchers have paid attention to this category of robots too [1,2,3].

Among them were H. Deman, D. Lefeber, J. Vermeulen who have developed a control algorithm to drive the dynamics of the robot leg in a cinematic way, in order to obtain a stable dynamic behavior [4]. T. Wadden and Ö. Ekeberg built a neuro-mechanical model to achieve a step leg movement [5]. Tsuchiya, K., Aoi, S., and Tsuji, K. have used motion controllers based on local feedback control to drive the feet's actuators [6].

Their goal was to prove that applying this model of control, the robot will present a considerable adaptability to environmental conditions. Luther R. Palmer, David E. Orin, Duane W. Marhefka, James P. Schmiedeler, Kenneth J. Waldron are using fuzzy control systems to drive leg's motion of a high-speed robot [7,8].

In this paper we intend to find the optimal position of the leg after performing a step, given the influence of three factors (precision, motion and friction) on its motion. Thus we apply the Inverse Kinematics to find the angles of each joint. These values represent the input of a Genetic Algorithm – with these values we generate the algorithm's initial population. The Genetic Algorithm is used to obtain optimal values for the angles of joints, taking into account those three factors. They are introduced in the algorithm using the Fitness Function and their weights are determined using the Analytical Hierarchy Process.

2 Control method for leg movement of a walking robot

Walking robots relies, in turn, on some legs, moving the others into a new position, providing a stable support. Successive occupation of new positions, in a safe and stable travel, involves defining walk. The walking robot's step is defined as the distance to that center of gravity moves in a cycle of locomotion. To achieve and lead a walking robot is necessary to know all the possibilities of walking, because the choice of the number of legs and their structure depends very much on the type of walking selected [10].

A cycle of leg movement of a walking robot consists of two phases, namely: *reliance* or *support* phase and the phase of *return* or *transfer*. At first, the support of the foot is in contact with the ground surface that is moving on; at the *transfer* phase, the walking robot's foot does not touch support surface and is moving so as to achieve the robot's state of stability in overall.

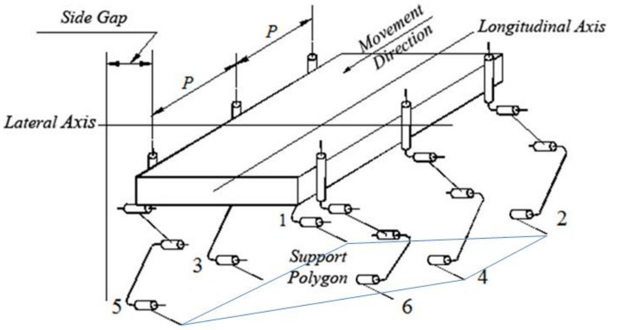

Fig. 1 Mechanical structure of a hexapod walking robot

The control using Direct Kinematics consists of transforming the current robot coordinates, resulting directly from the measuring transducers of each axis, into Cartesian coordinates and comparison with the desired target to be achieved in Cartesian coordinates (reference motion). The resulting error is the difference of position, represented in Cartesian coordinates, needed to be changed. Using the inverse matrix can provide the transformation of position error from robot coordinates in Cartesian coordinates, which allows the generation of angular errors for the direct control of the actuator for each axis [9].

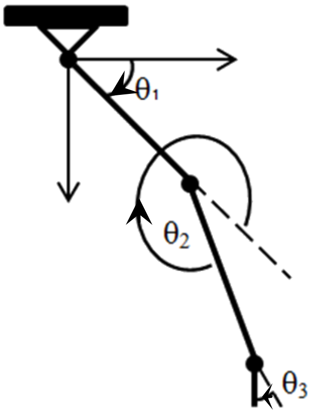

Fig. 2 Robot's foot after performing a step

Let us consider the hexapod walking robot in Fig.1, with a drive mechanism with three degrees of freedom, shown in Fig. 2, which we know the dimensions of leg segments: l_1, l_2 and l_3. Robot's platform is at height h from the surface on which it moves. In a sequence of walking, we want to move the leg, after a given trajectory, from the point where it is at the beginning of sequence - $P_i\ (x_i, y_i)$ - to a point $P_{i+1}\ (x_{i+1}, y_{i+1})$ where to get the leg after performing the step, forming an angle ϕ. The angle ϕ is the angle formed by l_3 and the surface on which the robot moves.

Equations of motion for the mechanism of robot's movement system with the structure in Fig. 2 are the following:

$$\begin{cases} x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos\phi \\ y = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin\phi \\ \phi = \theta_1 + \theta_2 + \theta_3 \end{cases} \qquad (1)$$

In the same time we can write the relation:

$$\phi = \tan^{-1}\left(-\frac{2pas_i + pas_{i-1}}{2h} \right) \qquad (2)$$

where: Pas_i is the current step length, and Pas_{i-1} is the previous step length. Solving this system we obtain two values for each angle θ.

Starting from the motion mechanism of the robot in Fig. 2, with three degrees of freedom and with the equations of motion in relations (1) and (2), we have obtained two solutions for each joint. These solutions are used to generate initial population for the genetic algorithm. Determining the fitness function, we must have regard to the three constraining factors (precision, movement and friction), for which we use the Analytical Hierarchy Process.

The factor Precision refers to positioning precision; the Movement is about the velocity of joint's movement and affects the time in which this movement is realized; the Friction implies the energy loss in the process of movement.

The control method diagram is shown in Fig. 3:

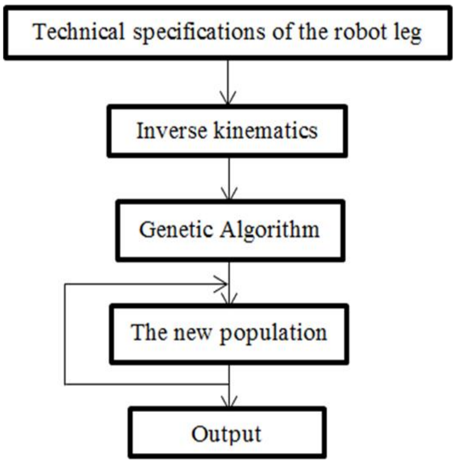

Fig.3 Motion control system diagram of movement of walking robots

The algorithm is repeated until the new population fits with what we want to achieve, in which case the result is stored and the algorithm stops.

3 Genetic algorithm for optimization.

Genetic algorithms are part of the numerical algorithms used in optimization problems solving, inspired from natural genetics and natural selection. The method presented in Fig. 4 is a general one, capable of being applied to an extremely wide field of issues. Unlike some private approaches, genetic algorithms are general techniques and are used to help solve practical problems on a regular basis.

The results from above for θ angles are used to form the initial population of genetic algorithms.

- **Start** – generates the initial population of n chromosomes (solutions using the system equations)

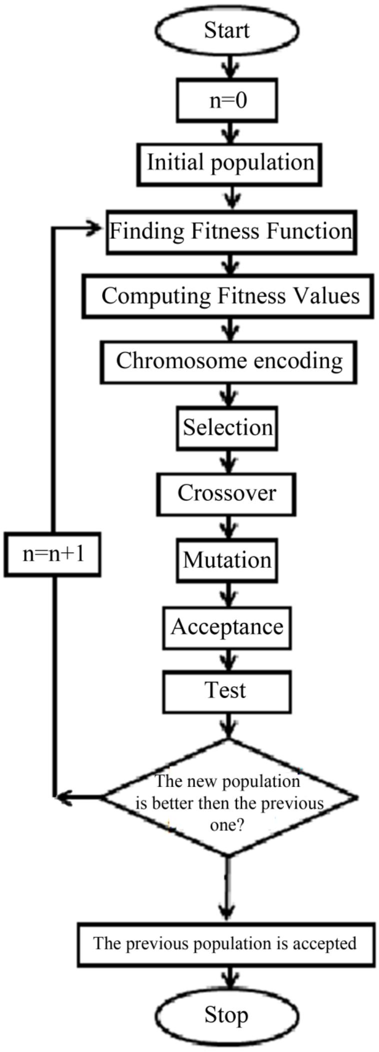

Fig 4. Genetic Algorithm Diagram

- **Fitness** – determine the fitness of each chromosome in the population
- **New population** - this is achieved by following the next steps until the new population is complete:

- selection - based on fitness value, we select the parent chromosomes of the initial population (the higher the fitness value, the greater chance of being selected is)

- **crossover** - is the operation in which individuals of a population exchange information among themselves in a manner similar to that used by natural organisms that reproduce. Using a particular method of crossing, three parent chromosomes are crossed and form new child chromosomes.

- **mutation** - used to change the values of a bit from a bit string of one or several individuals, but not all (using a mutation probability). Mutation operator is therefore a unary transformation, which made some disturbance of chromosomes obtained by crossover.

- **acceptance** - child chromosomes are placed in the new population. This population will be used for subsequent runs of the algorithm.

- **Test** – the condition is checked. If this is fulfilled, will receive the best solution in this population. Otherwise the algorithm is restarted.

The new population's chromosomes are tested using the equations of movement that provides us the coordinates x and y of point P_i, but unlike the first situation, this time we allow the algorithm to have a margin of error. So the new population should satisfy the following condition:

$$\begin{cases} l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) + l_3\cos(\theta_1 + \\ +\theta_2+\theta_3) = x \pm \varepsilon \\ l_1\sin\theta_1 + l_2\sin(\theta_1+\theta_2) + l_3\sin(\theta_1 + \\ +\theta_2+\theta_3) = y \pm \varepsilon \end{cases} \tag{3}$$

4 Analytical Hierarchy Process.

Analytical Hierarchy Process is a structured technique used to make complex decisions. Rather than describing a decision to be "correct", Analytical Hierarchy Process helps you find the solution that best fits to the problem and to understand its purpose - is an organizational process for decisions that people have already made. Based on mathematics and psychology, Analytical Hierarchy Process provides a comprehensive and rational framework for structuring decision problems, for representing and quantifying its elements, to link those elements of the global objectives and for evaluating alternative solutions.

In short, the Analytical Hierarchy Process is a method of obtaining a rational scale from pair comparisons. The input data can be obtained from actual measurements or height and price targets and opinions or feelings of satisfaction. Analytical Hierarchy Process allows a small inconsistency in thinking because human thinking is not always consistent. Rational scale vector is obtained from its main index of consistency and self-worth derived from primary.

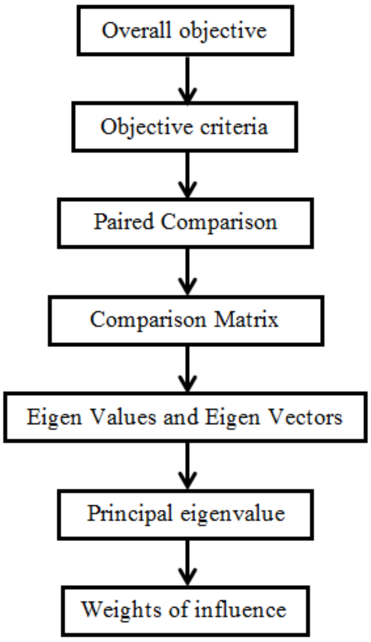

Fig 5. Analytical Hierarchy Process Diagram

Analytical Hierarchy Process diagram is shown in Fig. 5 with the main components explained below.

- **Start** – determine the overall objective
- **Factors** - determine the factors that satisfies the overall goal
- **Paired comparison** - two factors are grouped in pairs and, on a scale of 1-9, each is rated according to the other, for n factors to be compared we have to do $\frac{n(n-1)}{2}$ comparisons.
- **Comparison matrix** - is an n x n matrix that has value 1 on the main diagonal, the upper triangle is filled with the value assigned to the comparison in pairs; if this value is to the left of 1 and the inverse of that value if it is positioned to the right of 1. The lower triangle is filled with the inverse values of the upper triangle of the matrix.
- **Eigen vectors and Eigen values** - there are calculated the eigenvalues and eigenvectors of the comparison matrix
- **Main Eigen value** – from previously obtained eigenvalues is chosen λ_{max}
- **The weights of influence** – are given by the main eigenvector, according to his λ_{max}

A fitness function is a particular type of objective function that establishes the optimality of a solution (in our case, a chromosome) in a genetic algorithm so that a particular chromosome can be placed before the other chromosomes. Optimal chromosomes are used to cross in the idea of producing a better generation.

The Fitness function is given by the following relation:

$$f(x) = \theta_1 \times \lambda(\theta_1) + \theta_2 \times \lambda(\theta_2) + \theta_3 \times \lambda(\theta_3) \quad (4)$$

where θ_1, θ_2, θ_3 are vectors formed by the values of each angle, and λ (θ_1), λ (θ_2), λ (θ_3) represents the weights of influence of the three factors on each angle θ.

5 Results and conclusions.

Robot's foot specifications. We refer to a leg with three degrees of freedom and consider the known dimensions of the robot leg segments: $l_1 = l_2 = 250$ mm and $l_3 = 50$ mm. Note that the robot platform is 440 mm high. In a sequence of walking, we want to move the leg, after a given trajectory, from the point where the stepping sequence begins - P_i (x_i, y_i, z_i) – to a point P_{i+1} (x_{i+1}, y_{i+1}, z_{i+1}) where to get the leg after stepping. A normal step length is 560 mm. We consider as starting point P_0 (0, 0, 0) and end point will be P_1 (280, -440, -32 471).

Equations of motion for the robot with these technical specifications are:

$$\begin{cases} 280 = 250\cos\theta_1 + 250\cos(\theta_1+\theta_2) + \\ +50\cos(-32.471) \\ -440 = 250\sin\theta_1 + 250\sin(\theta_1+\theta_2) + \\ +l_3\sin(-32.471) \\ \quad -32.471 = \theta_1 + \theta_2 + \theta_3 \end{cases} \quad (5)$$

Solving this system we obtain the following solutions:

$$\theta_1 = \begin{matrix} -42.51 \\ -77.63 \end{matrix}, \theta_2 = \begin{matrix} -35.11 \\ 35.11 \end{matrix}, \theta_3 = \begin{matrix} 45.15 \\ 10.04 \end{matrix} \quad (6)$$

Analytical Hierarchy Process. In our case, the problem is finding the optimum solution for foot movement in a walking sequence. Of the many factors influencing this movement we choose three, namely: precision, movement and friction, with the influence angles presented in Table 1, which were determined experimentally on a hexapod walking robot by virtual projection method [11]. Because the influence of these factors is reflected in the final solution, we will use them in determining the genetic algorithm fitness function. Analytical Hierarchy Process introduces these factors as fitness function.

Table 1. Factors influence on angles

Factor	Influence of the factors on angles		
	θ_1	θ_2	θ_3
Precision	0.6584	0.6434	0.0806
Friction	0.0887	0.0738	0.3324
Motion	0.2529	0.2828	0.5870

Using the product between λ_{max} and its corresponding eigenvector we obtain the following results:

$$\lambda(\theta_1) = 3.46, \lambda(\theta_2) = 3.09, \lambda(\theta_3) = 3.32 \qquad (7)$$

Thus, the fitness function is given by:

$$f(x) = \theta_1 \times 3.46 + \theta_2 \times 3.09 + \theta_3 \times 3.32 \qquad (8)$$

Genetic Algorithm. Solving the equations of motion from relations (1) and (2), we obtain two values for each angle. The six values that form eight chromosomes and that will be the initial population for genetic algorithm:

Table 2. Initial population and fitness values

No.	**Chromosome**	**Fitness Value**
1.	42.51, 324.89, 45.15	1.3009
2.	42.51, 324.89, 10.04	1.1843
3.	42.51, 35.11, 45.15	0.4055
4.	42.51, 35.11, 10.04	0.2889
5.	77.63, 324.89, 45.15	1.4224
6.	77.63, 324.89, 10.04	1.3058
7.	77.63, 35.11, 45.15	0.5270
8.	77.63, 35.11, 10.04	0.4104

To obtain the full number of possible chromosomes in the genetic algorithm's initial population, choose to replace one of the values of θ_2 with its complementary value.

We want to have a precision as good as possible, so in order to achieve this we have chosen the margin of error to be of $5mm$, so the test condition is given by:

$$\begin{cases} 250\cos\theta_1 + 250\cos(\theta_1+\theta_2) + 50\cos(\theta_1 + \\ +\theta_2+\theta_3) = 280 \pm 5 \\ 250\sin\theta_1 + 250\sin(\theta_1+\theta_2) + l_3\sin(\theta_1 + \\ +\theta_2+\theta_3) = -440 + 5/-0 \end{cases} \qquad (9)$$

The algorithm runs 3244 times and we get the final population shown below:

Table 3. Final population and fitness values

No.	**Chromosome**	**Fitness Value**
1.	268.82, 201.73, 232.44	2325.2
2.	244.21, 173.90, 41.77	1521.0
3.	249.66, 17.98, 182.44	1525.1
4.	226.25, 175.42, 227.41	2079.9
5.	259.26, 255.42, 308.21	2709.5
6.	216.64, 177.55, 0.80	1300.9
7.	319.46, 148.47, 214.66	2276.8
8.	240.02, 298.38, 226.94	2505.9

And the solution that fits our condition is:

$$\theta_1 = 268.82, \theta_2 = 201.73, \theta_3 = 232.44 \qquad (10)$$

With these values we obtain the coordinates (283.40, -436.45) of the target point P_1, so a difference of $3.40mm$ for *x*, respectively $3.55mm$ for *y*, which means an error of 0.79% that is very small and can be ignored.

To increase the diversity of the exchanged information we have chosen to realize the crossover between three parent chromosomes.

Changing the stop condition and concerning the weights of influence from the Analytical Hierarchy Process on the Precision, we managed to develop and an improved algorithm, with much better positioning, then in our previous research.

To obtain a motion with high positioning precision in moving the leg of a hexapod walking robot in a step sequence, using the values from the equations of motion, we apply a genetic algorithm. We can obtain a much better precision reducing the margin of error, but this implies an increased computing speed, which requires higher computing resources, so bigger costs.

Solving mathematical system we obtained accurate values of the angles required to move the leg in the position specified. After applying the algorithm we reach the values placed somewhere near the foot of the desired point, into the proposed margin of error. The results showed that by applying the genetic algorithm in correlation with the analytical hierarchical process, we achieve a leg movement with high precision but with an increased time of execution.

References

[1] Kumar, R., *Simulation of Robotic Arm using Genetic Algorithm & AHP*, World Academy of Science, Engineering and Technology, Vol.25, 2007, pp.95-101

[2] Hongnian, Y., *Overview of Human Adaptive Mechatronics,* Int. Conf. on Mathematics & Computers in Business and Economics (MCBE '08), Bucharest, Romania, June 24-26, 2008

[3] Pop N, Vladareanu L, Pop P, *Finite Element Analysis of Quasistatic Frictional Contact Problems with an Incremental-Iterative Algorithm,* 8th International Conference on Applications of Electrical Engineering/8th International Conference on Applied Electromagnetics, APR 30-MAY 02, 2009, Univ Houston, Houston, Proceedings of the 8th International Conference on Applications of Electrical Engineering/ 8th International Conference on Applied Electromagnetics, pg. 173-178, 2009, WOS:000266634900027

[4] Deman, H., Lefeber, D., Vermeulen, J., *Design and control of a robot with one articulated leg for locomotion on irregular terrain,* Proceedings of the Twelfth CISM-IFToMM Symposium on Theory and Practice of Robots and manipulators, Springer Verlag, 1998, pp. 417–424, Vienna.

[5] Wadden, T., Ekeberg, Ö., *A neuro-mechanical model of legged locomotion: single leg control,* Biological Cybernetics, Vol.79, 1998, pp. 161-173.

[6] Tsuchiya, K., Aoi, S., Tsujita, K., *Locomotion control of a multi-legged locomotion robot using oscillators*, IEEE intl. conf. SMC, Vol.4, 2002, 6pp.

[7] Vladareanu L, Tont G, Ion I, Velea LM, Gal A, Melinte O, *Fuzzy Dynamic Modeling for Walking Modular Robot Control,* International Conference on Applications of Electrical Engineering (AEE '10), Penang, MALAYSIA, AEE '10: Proceedings of The 9th Wseas International Conference on Applications of Electrical Engineering, pg. 163- 170, 2010, WOS:000278030700024, SN 1790-5117

[8] Palmer, L., Orin, D., Marhefka, D., Schmiedeler, J., Waldron, K., *Intelligent control of an experimental articulated leg for a galloping machine,* IEEE International Conference on RA, Vol.3, 2003, pp. 3824-3827.

[9] Vladareanu, L., Tont, G., Ion, I., Vladareanu, V., Mitroi, D., *Modeling and Hybrid Position-Force Control of Walking Modular Robots*, Proceedings of The American Conference on Applied Mathematics; Pag:510-518; American Conference on Applied Mathematics JAN 27-29, 2010 Harvard Univ, Cambridge, MA; ISBN: 978-960-474-150-2

[10] Dumitru, S. , Bucur, D., Marin, D., *Methods and algorithms for motion control of walking mobile robot with obstacle avoidance,* Proceedings of the European Computing Conference (ECC '11), Paris, France, 2011, pp. 404 – 409, ISBN: 978-960-474-297-4

[11] Vladareanu L., Velea L M, Munteanu R., Curaj A., Cononovici S, Sireteanu T, Capitanu L, Munteanu M S, *Real time control method and device for robot in virtual projection,* EU patent no. EPO-09464001, 2009, EP2105263-A2; EP2105263-A3