

# An Evolutionary Gait Generator with Online Parameter Adjustment for Humanoid Robots

Amin Zamiri<sup>†</sup>, Amir Farzad<sup>†</sup>, Ehsan Saboori<sup>†</sup>, Mojtaba Rouhani<sup>‡</sup>, Mahmoud Naghibzadeh<sup>†</sup>,  
Amin Milani Fard<sup>†</sup>

<sup>†</sup>*Department of Computer Engineering, Ferdowsi University of Mashhad  
Mashhad, Iran*

<sup>‡</sup>*Engineering Department, Islamic Azad University of Gonabad  
Gonabad, Iran*

*Amin\_Zamiri@ACM.org*

## Abstract

*This article proposes a new hybrid methodology, together with an associated series of experiments employing this methodology, for an evolutionary gait generator that uses trigonometric truncated Fourier series formulations with coefficients optimized by a Genetic Algorithm. The Fourier series is used to model joint angle trajectories of a simulated humanoid robot with 25 degrees of freedom. The humanoid robot in this study learns to imitate the human walking behavior on flat terrains in a dynamically simulated environment. The simulation result shows the robustness of the developed walking behaviors even in extremely high and low speeds providing appropriate frequency. Number of range limitations were applied to the genetic algorithm used in this research to improve the learning period to less than 48 hours. The research seeks to improve upon the previous works on evolutionary gait generation, in robots with lower degrees of freedom. In addition, the proposed solution adapts a hybrid approach, thereby avoiding the long learning curves and unstable and slow gaits associated with evolutionary approaches.*

## 1. Introduction

Building robots that imitate human behavior to perform their actions like walking and running is amongst the most popular, and at the same time most complex tasks, in autonomous system design. This complexity is mainly due to the difficulty to cope with the many Degrees of Freedom (DOF) of a humanoid robot. This high number of degrees of freedom in a biped mobile robot creates new problem spaces in control and navigation where conventional methods

often fall short [1]. To reduce the complexity of the analysis, some researchers adopted a simplified dynamic model such as the inverted pendulum with certain assumptions on the robot's motion and structure [2]. While these simplifications come handy in designing initial trajectories, there still exist significant differences between the dynamics of a simple bipedal robot and a genuine humanoid robot with a high DOF.

A popular approach used for joint trajectory planning for bipedal locomotion is based on the Zero Moment Point (ZMP) stability indicator. In many ZMP-based trajectory planning approaches, motion planning is presupposed and performed in the Cartesian space [3, 4].

Hence, evolving control systems for robot locomotion is becoming a standard approach for the generation of improved or newer control systems for robots [5]. Various learning approaches for bipedal locomotion have been proposed by several researchers. Lin Yang et. al presented the Genetic Algorithm Optimized Fourier Series Formulation (GAOFSF) method for stable gait generation in bipedal locomotion in [6]. They use Truncated Fourier Series (TFS) formulations together with a ZMP stability indicator to generate the feasible gaits for a simple seven-link planar robot. A genetic algorithm is then utilized to search for optimal gaits according to the objective functions considering the specified constraints.

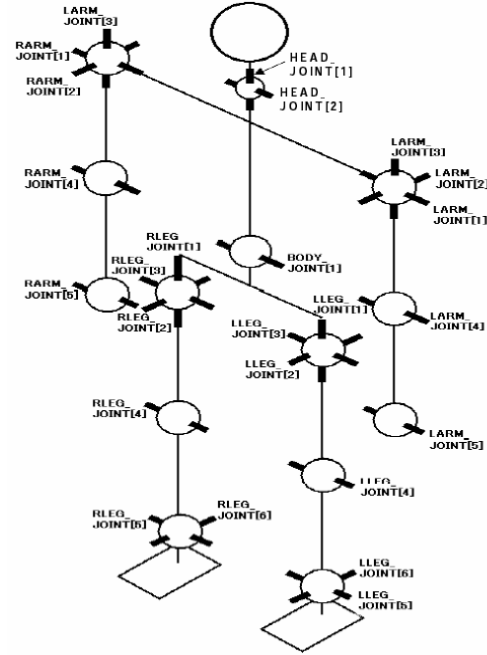
Some researches have used genetic algorithm to directly generate joint trajectories for each step [7, 8]. These trajectories are then applied to joints repeatedly while walking. Although this method is successfully utilized for biped robots, the generated gaits can not be changed to achieve desirable real-time motion adjustment.

In this paper, automatic evolution of walking behavior in a simulated humanoid with online adjustable speed is discussed. The robot in this study is a simulated model of Fujitsu's HOAP-2 that is a genuine humanoid with two arms and two legs and 25 DOF (Figure. 1). The simulation is performed by Spark, a generic three-dimensional physics simulator based on Open Dynamics Engine (ODE). Spark is capable of carrying out scientific distributed multi-agent calculations as well as various physical simulations ranging from articulated bodies to complex robot environments [9]. Robot models simulated in Spark can be easily controlled using programming languages like c++ and java. A simple PD controller was implemented to control the joint motors, but due to the adaptive nature of evolutionary methods, any other type of controller can be used.

Since the movements of the robot are known to be periodic while walking on flat plains, the motion of every joint can be expressed in terms of a trigonometric truncated Fourier series. Coefficients of the Fourier series are determined by using a genetic algorithm. Each individual in the genetic algorithm contains a set of coefficients for every joint's Fourier series, and thus defines a gait. These gaits are then tested in the simulation environment until the robot falls down over the ground or a sufficient amount of time passes by. The fitness is calculated based on the forward movement of the robot and the total time of the test. Once all individuals of the current generation are tested, the next generation is generated by applying GA operators over the best fit individuals. Using this method, a relatively fast and stable walking gait is evolved within a three-day simulation time on a Pentium IV 2.8GHz machine with 1GB of physical memory.

## 2. Truncated Fourier series

Evolutionary approaches include trying to optimize the parameters of a given type of motion model [10]. Common to these methods is the fact that a certain amount of knowledge of how locomotion is performed is implicitly present in the model. This narrows down the search space thus reducing the time needed for the optimization process. For a deeper look into biped motion properties we recorded and processed HOAP-2 (Figure. 2) walking gait which is included in the Webots simulation software [11].



**Figure 1 Fujitsu HOAP-2 humanoid robot [12]**

Figure. 2(a) and Figure. 2(b) give the knee and hip trajectories for the HOAP-2. These trajectories are identical in shape for both legs, but are shifted in time relative to each other by half of the walking period. The gait period is given by  $2\pi/\omega$  where  $\omega$  is defined as the gait frequency in radians per second (rad/s). Because of the periodic nature of the motion we can formulate joint angles using Fourier series. The Fourier Series of a periodic function of time  $f(t)$  can be written as:

$$f(t) = a_0 + \sum_{n=1}^{\infty} \left( a_n \cos \frac{2n\pi}{T} + b_n \sin \frac{2n\pi}{T} t \right)$$

Where  $a_n$  and  $b_n$  are constant coefficients and  $t$  is the period. The period can be calculated by desired fundamental frequency  $\omega$  by  $t = 2\pi/\omega$ .

Since the servo motors used in the robot act as low pass filters, we expected to be able to omit higher order frequencies without decreasing performance because these frequencies cannot contribute to the motion that is actually being executed by the robot. Thus we use Truncated Fourier Series (TFS) which have only the 3 first terms of trigonometric form of Fourier series.

$$f(t) = A \left[ a_0 + \sum_{n=1}^m \left( a_n \cos \frac{2n\pi}{T} + b_n \sin \frac{2n\pi}{T} t \right) \right]$$

Where  $a_n$ ,  $b_n$  and  $t$  are same as Equation 1, and  $a$  is an amplitude scaling parameter used for changing the step length. The parameter  $m$  determines the number of terms in the Fourier series. Using this formula significantly reduces the subsequent computational load in the search for feasible and optimal solutions using GA which is discussed in next section.

One of the advantages of this approach is that trajectory generation is done directly in the joint space. As such, inverse kinematics computation is not required thus avoiding the singularity problem. The walking rhythm, speed, and walking pattern can also be adjusted online through tuning either a single or two parameters.

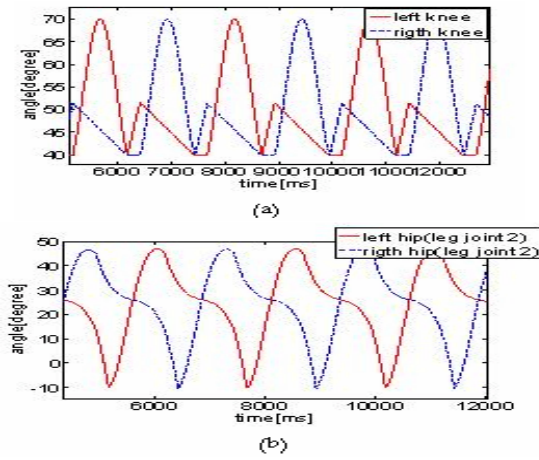


Figure 2 Knee and hip trajectories for HOAP-2

### 3. The genetic algorithm

As discussed in the previous section of this study, each joint trajectory can be modeled by a truncated Fourier series. A genetic algorithm was used to search for the optimal values of the coefficients so as to achieve stable walking behavior with desirable characteristics for the robot model. The first 3 terms of the general trigonometric Fourier series were used in the formulations. This is due to the fact that very high frequencies are normally filtered by the joint motors. The frequency of the Fourier series is considered as a constant value and can be changed after the offline learning is finished. This way every Fourier series could be represented by 7 real numbers and as for HOAP-2 model, 25 series are required to define a gait (One for each joint). This makes the size of the individuals relatively large and as a result the learning time gets very long. Therefore, some improvements

should be taken into consideration to make the learning process more effective.

Joints of the left part of the robot's body get the same periodic values as the right ones while walking straight, with a delay equal to a period time and there's no need to consider a separate set of Fourier series for the right part in the chromosome structure. This makes the chromosome almost half in size. Besides, the joints in charge of controlling the foot can get their values from the knee and hip joints as they could always be kept parallel to the ground line. (Figure. 3) Such automatic control of the foot is extremely beneficial in terms of keeping robot's balance, especially in the early stages of the learning phase. Head joints can be also ignored in the gait for the sake of simplicity.

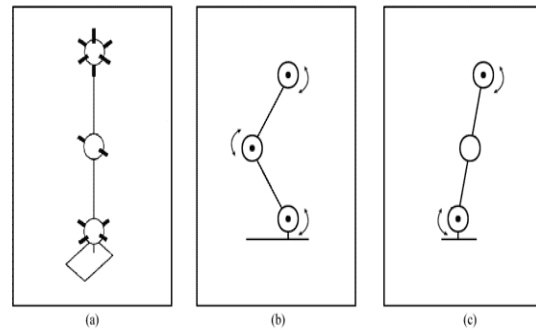


Figure 3 The Foot joints could be kept parallel to the ground line (a) The HOAP-2 leg joints configuration, (b) the leg front view and (c) the leg side view

With omitting excessive joints, the final chromosome will contain 7 real numbers for every one of the 12 primary joints. Considering this relatively large chromosome size, there are 300 individuals per generation, and 900 generations.

The basic GA algorithm is as follows:

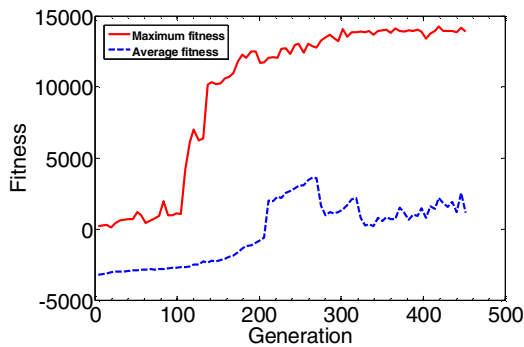
- A random number generator is used to give random real values to Fourier series coefficients in chromosomes.
- The robot resets and locates in the initial position in the simulation environment.
- Chromosomes are tested one by one. In every simulation step, the motor value for each joint is calculated using a PD controller, based on the corresponding Fourier series that is defined by the current chromosome.
- Each test continues until the robot falls over or 60 seconds has passed by. After the test is finished the fitness is calculated on the basis of the time passed multiplied by the distance from the start point and the next test starts.

- Once all chromosomes have been tested, the roulette-wheel selection decides which individuals are allowed to reproduce using mutation and crossover with specified probability.

A custom crossover was implemented that consists of a simple two-point crossover together with a creep operator. The creep operator randomly increases or decreases some real values of the chromosome by a very small value between 0 and 5. This crossover was applied with the probability of 0.6 together with a mutation with probability of 0.05. Valid range for the coefficients was also limited to -50 to 50. These values were arrived at after some experimentation.

#### 4. Initial results

The learning process takes about 7 days to complete. However some decent walking behaviors begin to emerge within the second day of the evolution. These walking gaits are somewhat unstable and do not follow a forward straight line. After the 4th day of the process, stable gaits evolve gradually. Figure. 4 shows the average and maximum fitness values for the robot over 460 generations. These figures are averaged over 3 runs. Note that although elitism was employed, because of the accurate physics simulation with noise, etc., this allows for the maximum fitness to fall as well as rise from generation to generation.



**Figure 4 Maximum and average fitness over 460 generations**

Studying the initial simulation results with different frequencies reveals that almost all of the generated gaits during the learning phase convergence to a human-like walking behavior with similar movement patterns. These similarities include human-like movements of the hand and sinusoidal movements of the waist.

#### 5. The improved algorithm

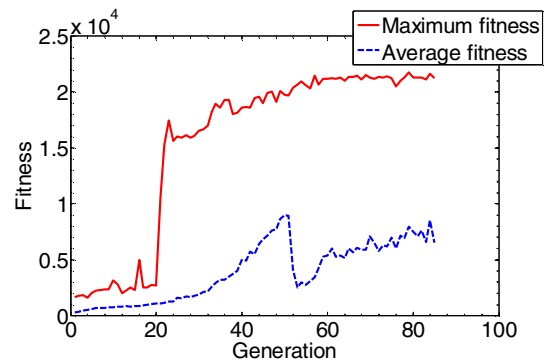
In the initial experiments it was found that some joint trajectories are similar in all of the generated gaits. These common trajectories share the same characteristics and differ in terms of their speed and frequency. For example, the elbow joint bends a little at the start of the walking and stays almost the same during the rest of the movement. Studying these similarities helps to further improve the evolutionary learning process by removing the unnecessary parameters. Experiments show that stable straight walking gaits can be generated by using only two joints of the leg: the hip joint 3 and the knee joint. This simplification affects the learning process dramatically and decreases the learning time to less than 48 hours (Figure. 5).

Based on the initial experiments, the fitness function was also changed in order to achieve better results. In this stage the time factor was ignored for the gaits that keep the robot walking for more than half of the test time. This helps faster gaits to get more fitness value over the stable but slow ones. Furthermore, the average amount of deviation was taken into account so that straight walks have more chance to be selected for regeneration. The simplified model of fitness function is as follows:

```

if (CurrentTestTime < TotalTestTime / 2)
    Fitness := Time * Distance
else
    Fitness := Distance - AverageDeviation
fi

```



**Figure 5 Maximum and average fitness over 86 generations**

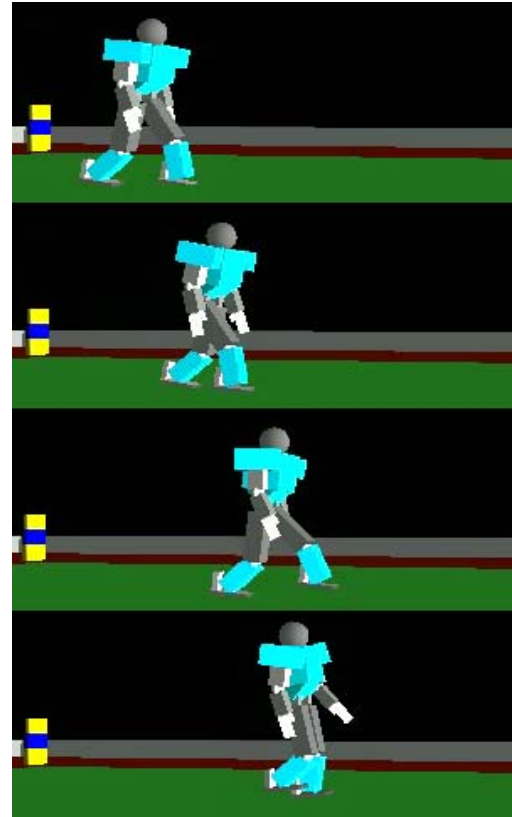
## 6. Online parameter adjustment

Walking behavior can be adjusted through changing one or two parameters. By changing the Fourier series' frequency and joint movement gain, desired walking speed can be achieved dynamically. In the final experiments we were able to stop the robot after some periods of walking by gradually increasing frequency and decreasing controller's gain.

The movement direction can also be determined by the leg joint 1. This joint is responsible for rotating thigh around Z axis. The value of this joint was kept unchanged during the learning process, but it can be slightly modified while walking to make the robot change its direction a few degrees. Figure. 6 shows the simulated robot in Spark simulation environment while walking with maximum speed.

## 7. Summary and future works

Using a set of Fourier series with parameters adjusted by a genetic algorithm, a simulated robot has been able to teach itself how to walk. We have described how to use an evolutionary algorithm to search for the optimal values for series coefficients to develop quite realistic walking sequences through the manipulation of up to 24 of the 25 motors used in the robot in relatively short time spans (certainly compared to human walking). The different evolutionary techniques used, such as custom cross-over operator and roulette wheel selection, were briefly described. The development of the algorithm used to determine the best walk sequences has been described in some detail. Work is continuing on using this technique for the development of controllers for humanoid robots, by again extending the number of degrees of freedom, and by the application of the techniques described here to different robot morphologies and environments. We believe that as well as aiding in the future development of humanoid robot locomotion generation, further research in this area may help in understanding certain aspects of different human behaviors like walking and running.



**Figure 6 Walking behavior of the simulated humanoid in Spark 3D simulation environment**

## 8. References

- [1] P. Nordin and M. Nordahl, "An Evolutionary Architecture for a Humanoid Robot," *In Proceeding of Symposium on Artificial Intelligence (CIMA99)*, Havana Cuba, 1999.
- [2] O. Kajita Matsumoto and M. Saigo, "Real-Time 3D Walking Pattern Generation for a Biped Robot With Telescopic Legs," *In Proceedings of the IEEE International Conference on Robotics and Automation*, May 21–26, 2001, Seoul, Korea, pp. 2299–2308.
- [3] M. Vukobratović and B. Borovac, "Zero-moment point, Thirty five years of its life," *International Journal of Humanoid Robotics*, Vol. 1, No. 1, 2004, pp. 157-173.
- [4] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi and K. Tanie, "Planning walking patterns for a biped robot," *IEEE Transactions on Robotics and Automation*, 2001, pp. 280-289.
- [5] A. Boeing, S. Hanham and T. Braunl, "Evolving autonomous biped control from simulation to reality," *In Proceedings of the 2nd International Conference on Autonomous Robots and Agents*, December 13–15, 2004, ACM press, New Zealand, pp 440–445.

- [6] L. Yang, C. Chew, T. Zielinska, and A. Poo “A uniform biped gait generator with off-line optimization and online adjustable parameters,” *Robotica*, Vol. 25, Issue 2, Cambridge Press, March 2007, pp. 1-17.
- [7] M. Eaton and T. J. Davitt, “Evolutionary control of bipedal locomotion in a high degree-of-freedom humanoid robot: first steps,” *Artificial Life and Robotics*, Vol. 11, No. 1, Springer, pp. 112-115.
- [8] A. Boeing and T. Braunl, “Evolving splines: an alternative locomotion controller for a bipedal robot,” *In Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision (ICARCV02)*, December 2002, Singapore, pp 798–802.
- [9] J. Boedecker, “Humanoid Robot Simulation and Walking Behaviour Development in the Spark Simulator Framework,” *Artificial Intelligence Research University of Koblenz*, April 2005.
- [10] J. Hoffmann, U. Düffert, “Frequency Space Representation and Transitions of Quadruped Robot Gaits,” *In Proceedings of the 27th Australasian Computer Science Conference (ACSC 2004)*, Vol. 26, University of Otago, New Zealand, 2004, pp 275 – 278.
- [11] cyberbotics.com.
- [12] Fujitsu Automation, "Humanoid Robot HOAP-2 Specification." *Fujitsu Corporation Web site*, <http://jp.fujitsu.com/group/automation/downloads/en/services/humanoid-robot/hoap2/spec.pdf>.