Evolutionary approach for developing fast and stable offline humanoid walk

Hafez Farazi[#*1], Farzad Ahmadinejad[*2], Farhad Maleki[#3], M.E Shiri [#4]

[#] *Mathematics and Computer Science Department, Amirkabir (polytechnic Tehran) University*
Tehran, Iran

[*] Humanoid Robotics Lab, Amirkabir (polytechnic Tehran) University
Tehran, Iran

{h.farazi, f.ahmadinejad, farhad.maleki, shiri}@aut.ac.ir

***Abstract*—To make stable and fast walking in humanoid robot possible, lots of works has to be done including attitude estimation, dynamic stability controls, path planning and online position tracking. These processes need an exact mechanic modelling of the robot's body structure. In this paper, an evolutionary approach is used to model the human walk for a standard humanoid robot named NAO. This method makes use of inverse kinematics for trajectory generations. Additionally a novel method is proposed for mapping from gait design to genetic algorithm, in order to optimize generated trajectory parameters. The proposed method leads us to forward velocity of 64 cm/s, which is a superior result compared to the other common methods.**

***Index Terms*—Robotics, Humanoid, NAO, Genetic algorithm, Inverse kinematics**

I. Introduction

In the past three decades, implementing walking for humanoid robots has been studied by many researchers including experts in the fields of mechanic, physics, electronics, and computer science. Biped locomotion, due to the several Degrees Of Freedom (DOF) and a little support stability polygon is a challenging subject in robotics. Although many approaches have been presented to generate fast, stable, and more controllable walking motion, the results are still too far from human movements.

One of the classic solutions to make and model human walking was represented by Vocobratovic in 1972, called Zero Moment Point (ZMP) [1]. ZMP is a mechanical oriented model based on the idea of locating a point on the ground in a way that, the sum of the moments of all forces equals to zero. The gait is dynamically stable if ZMP is located in supported polygon. Although ZMP can be used for any human-like motions, it requires many expensive computations including complex dynamic equations solving and depends on the kinematic and the kinetics of the system.

Another common method is Central Pattern Generator (CPG), which works on nonlinear equations to model rhythmic walking controller. This method is adapted from neural activities and it is based on a complex mathematical model. Although this mathematical model plays an important role in this method, there is no certain way to find it.

In this paper an evolutionary approach is used to generate and model a fast and robust human walk.

The rest of this paper is organized as follows. Section 2 describes the human gait generator. Section 3 briefly explains robot's and the simulation's specifications. Section 4 focuses on inverse kinematics equations. Section 5 demonstrates the genetic algorithm. The proposed method is presented in section 6. Experimental results are presented in section 7. Finally, the paper ends with a brief conclusion and suggestions for future works.

II. The Gait Generator

Many human motions can be considered periodic, like walking and turn motion. One of the essential aspects of analysing human-like motion is walking trajectory, which is widely studied by two different approaches, namely angular and positional trajectory.

To follow angular or positional trajectory by all joints, a good controller is needed, and in this paper Proportional Derivate (PD) controllers are used for each individual joint, to drive all joint positions toward its goal position.

In order to propose a stable and robust human-like motion, it is a reasonable approach to study a human walk process.

Figure 1 Illustrates human hip and knee captured trajectory.

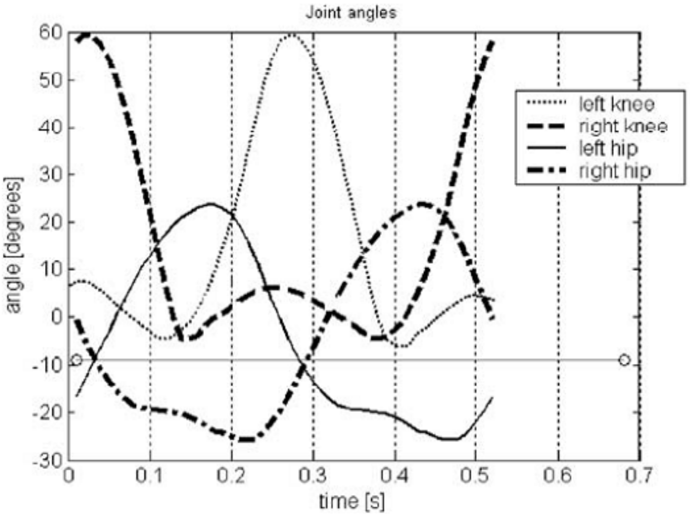

Fig. 1. Human walking angular trajectory [2]

978-1-4673-6315-0/13/$31.00 ©2013 IEEE

Assuming a walk sagittal symmetry, which applies the same movements and trajectory for left and right joints with a half, phase shift, is a common assumption in analysing human-like motions. This assumption is depicted in hip and knee trajectory in the Fig .2.

By analyzing main features of Fig .2 we will find that in [t2, t5] the right leg is support leg and while a leg is support, the other one is the swing leg therefor t2 and t5 are switch times and

In t3, left and right tights are crossing each other.

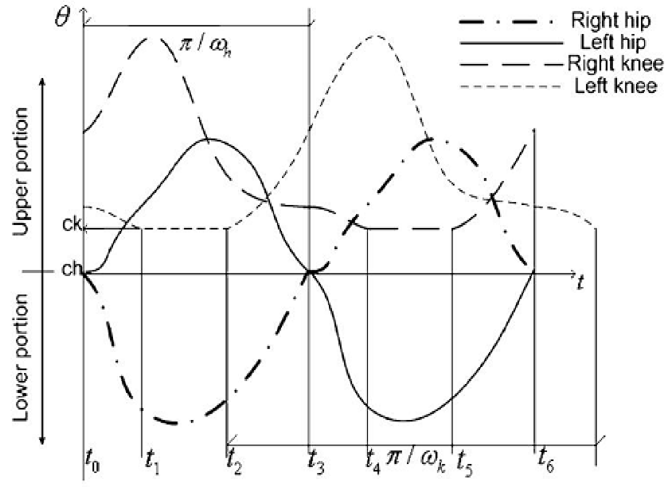

Fig. 2. Gait adapted from human gaits features [2]

A common way to make a robot gait is generating an oscillator function and placing each appropriate oscillator behind every joint. In order to keep all joints synchronous in a single frequency clock, each oscillator period should be the same. Since each single oscillator has at least 4 parameters and there are 6 joints in each leg, 48 oscillators are needed. It is obvious that optimizing 48 continuous parameters is a computationally expensive task [3]. In this paper, instead of using oscillator equations, a different approach, which has 13 parameters, is used for trajectory generation. This method reduces the complexity of the search space.

III. Simulator and Robot Model

Standard Robocup leagues like SPL and 3D simulation using a standard type of robot named NAO provide an appropriate platform to make and test humanoid locomotion and motion control, regardless of mechanical and hardware aspects of robots. NAO is a humanoid standard robot and has 21 DOF as follows [4]:

- Two in each Ankle
- One in each Knee
- Two in each Thigh
- One in Hip
- Four in each Arm
- Two in Head

The Robot weights 4.5kg stands 57cm high. The anatomy of the NAO is depicted in Fig. 3.

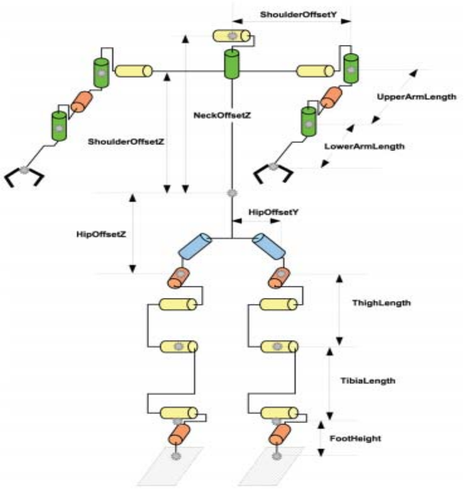

Fig. 3. NAO Anatomy and referential axis model [5]

As a three-dimensional standard simulator, RCSSSERVER3D [6] which is based on Open Dynamic Engine (ODE) and SimSpark, has been chosen. SimSpark can simulate robot's body, environment, and collisions. In this simulator, many physical and mechanical rules are obeyed to represent the true state of the robot.

IV. Inverse Kinematics

In order to reduce the search space we use inverse kinematics to generate each joint angle. We include 12 DOF in inverse kinematics formula, 6 for each leg. In each leg, the involved joints are as follows:

1. Hip $_{\text{Yaw}}$
2. Hip $_{\text{Roll}}$
3. Hip $_{\text{Pitch}}$
4. Knee
5. Ankle $_{\text{Pitch}}$
6. Ankle $_{\text{Roll}}$

Inputs for our inverse kinematic function, shown in Eq. 1, are x_1, y_1, z_1, yaw, and outputs are12 joint parameters as mentioned above.

$$\text{Joints degree} = \text{Inverse\_Kinematics}(x1, y1, z1, yaw) \tag{1}$$

As it illustrated in Fig .4, circulating around Yaw angle for each foot is calculated using Eq. 2 [7].

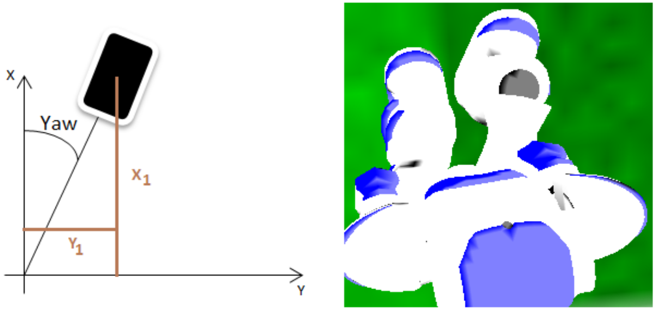

Fig. 4. Circulating foot around z axis

$$
\begin{aligned}
Hip_{Yaw} &= -yaw \\
x_2 &= x_1\cos(yaw) + y_1\sin(yaw) \\
z_2 &= z_1
\end{aligned}
\tag{2}
$$

Movements in Y-axis are calculated using Eq. 3 [7] and Eq. 2. A movement in Y-axis is depicted in Fig .5.

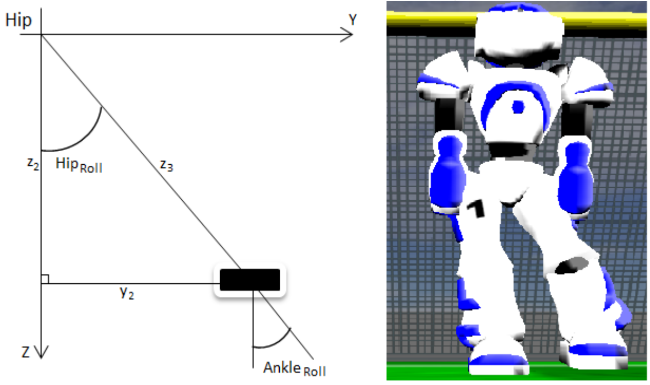

Fig. 5. Moving foot in Y axis

$$
\begin{aligned}
Hip_{Roll} &= \arctan\left(\frac{y_2}{z_2}\right) \\
Ankle_{Roll} &= -Hip_{Roll} \\
x_3 &= x_2 \\
z_3 &= \sqrt{{y_2}^2 + {z_2}^2}
\end{aligned}
\tag{3}
$$

Equation 4 [7] and Eq. 3 is used to calculate the joint angles in X and Z axis. Figure 6 is used to explain Eq. 4. In Eq. 4 we assume that upper leg and lower leg value have the same length. This assumption, due to the real value of NAO hardware manual [4] , is an almost correct assumption:

Thigh length=upper leg=100mm.

Tibia length= lower leg=102mm.

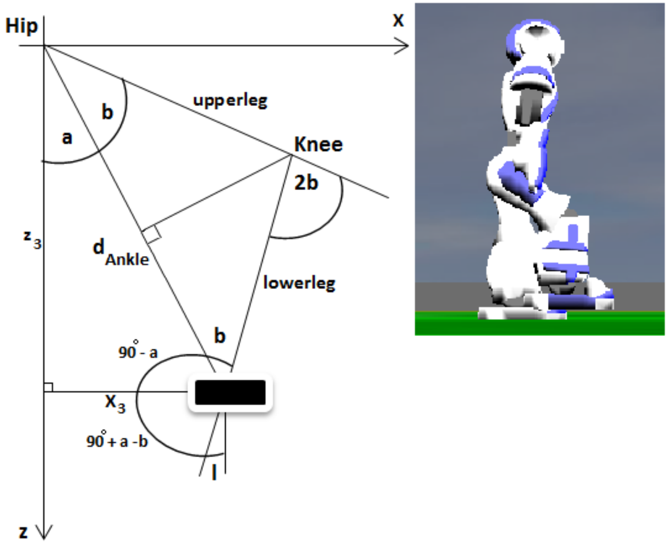

Fig. 6. Moving leg in Z and X axis

$$
\begin{aligned}
a &= \arctan\left(\frac{x_3}{z_3}\right) \\
b &= arccos\left(\frac{d_{Ankle}}{2\ upperleg}\right) \\
Ankle_{Pitch} &= l = -a + b \\
Knee_{Pitch} &= -2b \\
Hip_{Pitch} &= a + b \\
d_{Ankle} &= \sqrt{{z_3}^2 + {x_3}^2}
\end{aligned}
\tag{4}
$$

Each 4 inputs (x_1, y_1, z_1, and yaw) leads to 12 joint angles, and each joint angle apply to each joint, the result make a position in the robot. In fact making a motion is a result of position changing during the time.

Since there is a complicated and nonlinear relation between stable walking and gait trajectory, in this paper Genetic algorithm as an optimization method is used to find optimum trajectory parameter.

V. Genetic Algorithm

Genetic Algorithm (GA) is an evolutionary algorithm which is based on natural selection of Darwin's theory and genetics[8]. In GA, each chromosome codes a possible solution for the problem. GA begins with an initial population, which is usually generated at random. Then the fitness of each individual is calculated using the fitness function.

Regarding to the fitness value, some chromosomes have higher priority to the others. This priority is the foundation for natural Selection, which leads the process of the next generations.

The algorithm continues for several of iterations, in each iteration using selection function, mutation function and crossover function a new generation is obtained.

Definitions of three main GA functions are:

a. Selection function selects a group of the most fitted chromosomes. These chromosomes are used in the process of making a new generation.

b. Crossover function is used to make offspring from parents who have been selected by the selection function.

c. Mutation function is used to improve diversity of offspring. This function helps the algorithm to avoid trapping in local optima.

There are many selection functions proposed in the literatures. Roulette-wheel is one the most widely used selection functions. Suppose there are N individuals in the population, and each individual characterized by its fitness wi $> 0\ (i = 1, 2, 3, \dots, N)$. Based on roulette-wheel selection function, the i-th individual takes part in the process of producing offsprings with a probability of P_i which is calculated using Eq. 5 [9].

$$
p_i = \frac{w_i}{\sum_{i=1}^{N} w_i}\ (i = 1,2, \dots, N)
\tag{5}
$$

The iterations of GA continue until the stopping criteria are satisfied.

Figure 7 illustrates the general schema of GA.

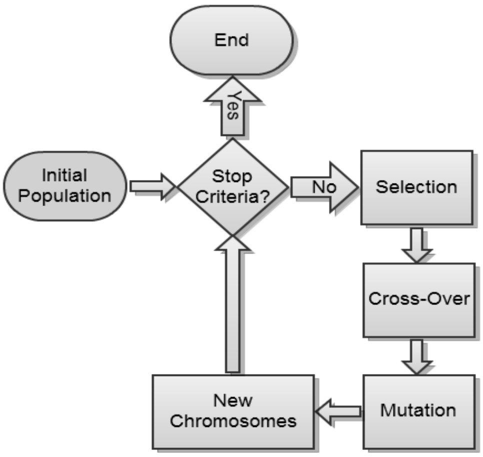

Fig. 7. Genetic Algorithm flow chart

VI. GA CONFIGURATION

Following parameters are selected to model gait trajectory and GA is used to optimize these parameters:

- 2-Dimentional vector chain locations in x-z plane
- Sitting parameter
- Hip movements
- Swinging arms degree

Each chromosome consists of ten 2-dimensional vector chain locations in x-z plane which determines robot foot trajectory in x-z plane, a sitting parameter which shows how much a robot sit in centimeter during the walk process, a hip movement parameter that specifies how much the heap moves in y axis, and a swinging arms degree parameter that determines how much an individual arm moves in degree.
We illustrate individual chromosome parameters in Fig. 8.

(X_i, Y_i) { i=1,2,...,10 }	**Sitting**	**Hip movement**	**Swinging arm degree**

Fig. 8. Individual Chromosome Parameters

In this paper, initial populations of 20 chromosomes are generated randomly. In each iteration, 5 chromosomes are selected using roulette-wheel and these chromosomes undergo mutation and crossover functions.
Briefly, the next generation is composed of the following chromosomes:

1) The 5 chromosomes selected using roulette-wheel
2) The 5 selected chromosomes from 1 undergo mutation function to produce 5 new chromosomes
3) The 5 selected chromosomes from 1 undergo crossover function to produce 5 new chromosomes
4) 5 chromosomes are generated randomly

The GA process will continue until stopping criteria are satisfied. In the following subsections random generator function, mutation function, fitness function and normalization function are described. In addition stopping criteria are explained.

A. Random generator function

This function generates 10 random points in pre-defined intervals based on the following conditions:

- In order to make a half-circle trajectory followed by robot’s foot, sum of all movements along z-axis should be equal to zero and also sum of all movements along x-axis should be equal to gait step length.
- Each gait step length should be at least more or equal to minimum movement of epsilon.

Sitting parameter, hip movements and swinging arm degree are generated randomly at predefined intervals.

B. Mutation function

In this function one of the following actions is applied on the input chromosomes randomly:

- Some of the 10 points are selected randomly, and their corresponding x and z components will randomly be changed by±.
- Sitting ratio parameter will randomly be changed by±.
- The hip movements parameter will randomly be changed by±.
- Swinging arms degree parameter will randomly be changed by±.

C. Crossover function

This function unlike usual crossover functions produce only one child and in order to gain this child, randomly name of the parents as main parent and the other as support parent, then copy the main parent to the child.
Finally apply one of the following actions to the child:

- Some points were randomly selected and set their values equal to the corresponding point in support parent.
- Set each of the 10 points, as the average of their corresponding parent points.
- Set sitting parameter, as the average of their corresponding parent parameter.
- Set hip movements parameter, as the average of their corresponding parent parameter.
- Set Swinging arm degree parameter, as the average of their corresponding parent parameter.

D. Fitness function

In all GA applications, choosing a proper fitness function is an important task, which should be done in order to achieve good results. For forward walking, it is a good idea to calculate fitness by measuring distance between the reached position and start position. Although this idea leads to a fast and often stable result, it does not guarantee a straight walk. To solve this issue, we obtain the amount of deviation from expected walking and subtract this value from the distance between the reached position and the start position.

In addition, to accelerate the convergence property of the proposed method, we assigned a penalty to each unstable walk according to Eq. 6.

$$s = \frac{|max\_angle|}{|\max\ \_allowed\_angle|}$$

$$fitness = \begin{cases} -100 & , |\max\ \_allowed\_angle| < |max\_angle| \\ s * (x - \alpha y\,) & , |\max\ \_allowed\_angle| \geq |max\_angle| \end{cases} \quad (6)$$

In this equation, s is a number between 0 and 1 and shows the deviation from standing angle. X is the difference between x-coordinates of the start and reached positions. Y is defined in a similar manner . α is a constant value which shows the deviation from straight walk penalty. Our experiments show that an appropriate value for α is 3.

In this formula “max_angle” determines the maximum vertical angle during a chromosome run time. “max_allowed _angle” refers to a maximum vertical angle that a stable walking robot can have, this parameter calculated by trial and error before starting the algorithm. In our experiments, it has been found that the value of 25° is the proper value of this parameter.

The locations of the robot at first and final positions are fetched from SimSpark’s location output and angles are read from gyro module in torso part of the robot.

E. Normalization function

After calling each of the previous functions, normalization function checks the following conditions. If this function returns false, the generated chromosome would not a valid one and it should be substituted by a new valid chromosome.

- If sum of all x parameters in each trajectory vector is not equals to zero, return false. This condition aims at preventing collision between foot and ground.
- If sitting parameter is greater than Thigh length of the robot, return false.
- If swinging arm parameter is greater than 90°, return false.
- If there is at least one y component less than an epsilon, return false.

F. Stopping Criteria

The algorithm will stop if one the following conditions are satisfied.

- Running duration of the algorithm exceeded a predefined time limit.
- There is no improvement in the last 30 populations.

VII. Experimental Results

After several hours running GA on a Core i5 2.53 GHz computer with 4 GB of physical memory, number of generations exceeded to 133. Figure 9 shows the average and maximum velocity during 133 generations.

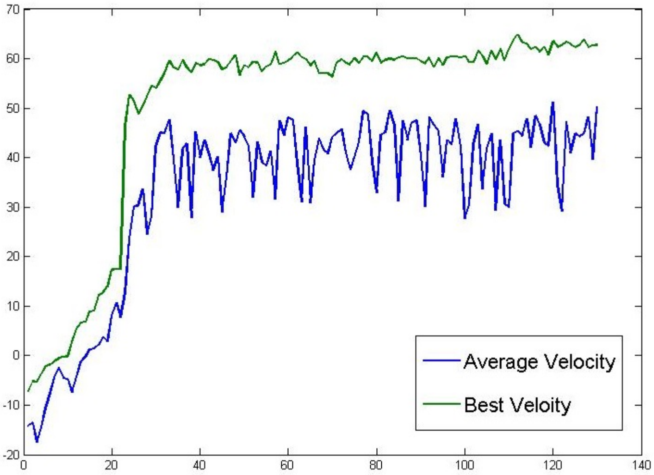

Fig. 9. Average velocity and maximum velocity during 133 generations

Non smooth plot in Fig. 9 for average velocity is a result of concluding 5 random chromosomes into each population. Although this approach causes a non-smooth velocity plot, it helps to prevent the proposed method from trapping in local optima.

The best and average population velocity in the first 10 populations is negative because of either walking backward or falling. The robot start to walk for the first time in the 11th generation with a velocity of 2.67 cm/s.

As a result of the proposed method, the simulated NAO robot achieved 64 cm/s in less than 5 hours of learning. This result shows the superiority of the proposed method over three well known methods [10] [11] [12] which respectively gain 45 cm/s, 50 cm/s, 17 cm/s of maximum velocity with the same simulator and robot in flat conditions. Two papers [10] [11] use learning approach and exceed 9 hours and 5 days of learning, respectively.

In Table I. the best result values are shown.

TABLE I. Best Generated individual

X_1,Y_1	0.26,0.93 cm	X_6,Y_6	0.64,-0.69 cm
X_2,Y_2	0.47,0.58 cm	X_7,Y_7	1.24,-0.65 cm
X_3,Y_3	2.71,0.79 cm	X_8,Y_8	2.71,-0.79 cm
X_4,Y_4	1.24,0.65 cm	X_9,Y_9	0.47,-0.58 cm
X_5,Y_5	0.62,0.69 cm	X_{10},Y_{10}	0.26,-0.93 cm
Sitting ratio	3.81 cm	Hip Move	0.26 cm
Swing arm		39□	

The generated walk screenshots are illustrated in Fig .10.

A video clip including the walk generating proccess and sideward motion generating based on this method is avalible [13].

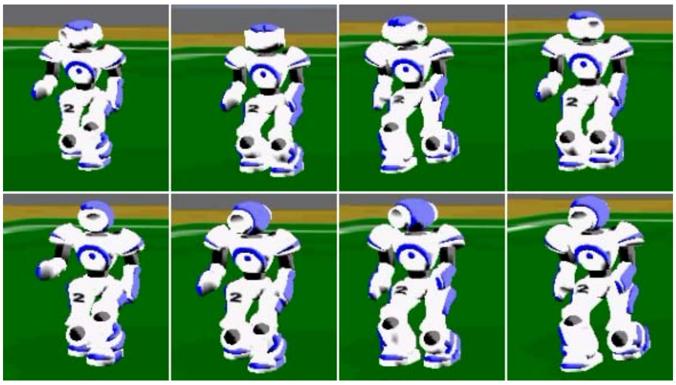

Fig. 10. Walking gait screenshots

VIII. CONCLUSION AND FUTURE WORK

This paper presented an evolutionary approach for developing and optimizing a fast, stable and straight walking motion with minimum sensor inputs. The main advantages of this method are; it's few parameters and it's time efficiency beside its high value of maximum velocity. The proposed method was tested on SimSpark as a simulating environment.

This method could apply for other periodic motions like sideward[13] and turning motions. Moreover developing Omni-directional locomotion for humanoid robot based on this approach is suggested as future research topics. Beside of Omni-directional locomotion, this novel method for mapping from gait design to GA could use for generating human-like walking on inclined surfaces.

Although simulation is not always efficient, especially in simulating friction between feet and ground, we believe that our best result can easily port on the real NAO and we are going to port it soon.

REFERENCES

[1] M. Vukobratović and J. Stepanenko, "On the stability of anthropomorphic systems," *Mathematical Biosciences,* vol. 15, pp. 1-37, 1972.

[2] L. Yang, C.-M. Chew, T. Zielinska, and A.-N. Poo, "A uniform biped gait generator with offline optimization and online adjustable parameters," *Robotica,* vol. 25, pp. 549-565, 2007.

[3] J. D. a. B. Weingarten, M. and Groff, R.E. and Koditschek, D.E., "Gait generation and optimization for legged robots," presented at the The IEEE International Conference on Robotics and Automation, 2002.

[4] *Aldebaran Robotics Co.* Available: http://www.aldebaran-robotics.com/documentation/nao/hardware

[5] D. G. and, V. H. and, P. B. and, C. K. and, J. e. r. o. m. M. and, P. L. and, B. M. and, J. S. and, and B. Maisonnier, "The NAO humanoid: a combination of performance and affordability," *CoRR,* vol. abs/0807.3223, p. http://arxiv.org/abs/0807.3223, 2008.

[6] J. Boedecker, "Humanoid Robot Simulation and Walking Behaviour Development in the Spark Simulator Framework," Artificial Intelligence Research University of Koblen, 2005.

[7] D. Scholz, "Modulare und plattformunabhängige dynamische Bewegungserzeugung für autonome mobile humanoide Roboter," Master, TU Darmstadt, 2008.

[8] D. E. Goldberg, "Genetic algorithms in search, optimization, and machine learning," *Addison-Wesley Professional,* 2009-12-14T10:56:13.000+0100 1989.

[9] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Physica A: Statistical Mechanics and its Applications,* vol. 391, pp. 2193-2196, 2012.

[10] N. Shafii, Seyed Javadi, M.H. & Kimiaghalam, B., "A truncated fourier series with genetic algorithm for the control of biped locomotion.," *IEEEASME International Conference on Advanced Intelligent Mechatronics,* 2009.

[11] H. Picado, M. Gestal, N. Lau, L. P. Reis, A. M. Tom\, and \#233, "Automatic Generation of Biped Walk Behavior Using Genetic Algorithms," presented at the Proceedings of the 10th International Work-Conference on Artificial Neural Networks: Part I: Bio-Inspired Systems: Computational and Ambient Intelligence, Salamanca, Spain, 2009.

[12] A. M. B, A. S. K, Y. Salehinia, and F. Najafi, "An Open Loop Walking on Different Slopes for NAO Humanoid Robot," *Procedia Engineering,* vol. 41, pp. 296-304, 2012.

[13] Video clip Avalible: http://youtu.be/S0MMWFI0gpk