Evolution of Biped Walking Using Truncated Fourier Series and Particle Swarm Optimization

Nima Shafii[1], Siavash Aslani[1], Omid Mohamad Nezami[1], Saeed Shiry[2]

[1] Mechatronics Research Laboratoy (MRL), Department of Computer and Electerical Engineering, Qazvin Islamic Azad University,
Qazvin, Iran
{shafii, saslani, mohamadnezami}@mrl.ir
[2] Computer Engineering Department, Amirkabir University,
Tehran, Iran
shiry@ce.aut.ac.ir

Abstract. Controlling a biped robot with a high degree of freedom to achieve stable and straight movement patterns is a complex problem. With growing computational power of computer hardware, high resolution real time simulation of such robot models has become more and more applicable. This paper presents a novel approach to generate bipedal gait for humanoid locomotion. This approach is based on modified Truncated Fourier Series (TFS) for generating angular trajectories. It is also the first time that Particle Swarm Optimization (PSO) is used to find the best angular trajectory and optimize TFS. This method has been implemented on Simulated NAO robot in Robocup 3D soccer simulation environment (rcssserver3d). To overcome inherent noise of the simulator we applied a Resampling algorithm which could lead the robustness in nondeterministic environments. Experimental results show that PSO optimizes TFS faster and better than GA to generate straighter and faster humanoid locomotion.

Keywords: Bipedal Locomotion; Particle Swarm Optimization; Truncated Fourier series

1 Introduction

In recent years, bipedal locomotion, especially "bipedal walking" has been one of the interesting research topics in multi disciplinary topic. Bipedal walking as a very complex motion, involves most of humanoid joints including its sensors and actuators. Many researchers have focused on this topic and a lot of approaches have been presented. But so far no method exists that can walk a robot as stable as human's do. There are two major approaches in bipedal walking researches; model-based and model free approaches. In model-based approach the designer first derives model of the robot and then builds a controller for the model. Two well known methods in this approach are "Zero Moment Point"[1] (ZMP) and "Inverted Pendulum"[2].

In model-free approach, which is also called "Dynamics Based", it is common to make use of the sensory information and associate it with motions. No physical model is used in this method that eases the implementation of the skills. There are three important studies done in this field; Passive Dynamic Walking (PDW) [3], Central

Pattern Generator (CPG) [4] and Ballistic Walking [5]. In PDW approach, the robot does not have any actuators model and walks just by utilizing the gravity force. The Ballistic walking is originated from the simple human walking model based on the observation of human walking in which the muscles of the swing leg are activated only at the beginning and the end of the swing phase. In CPG approach, special neural circuits take the role of rhythmic walking controller using the non-linear equations to model the neural activities. Researchers usually focus on complex mathematical models like Hopf [6] or Matsuoka [7] to model these neural activities and generate rhythmic walk patterns (Gaits).

In 2006, Truncated Fourier Series (TFS) formulation is used for gait generation in bipedal locomotion [8]. TFS together with a ZMP stability indicator are used to prove that TFS can generate suitable angular trajectories for controlling bipedal locomotion. It does not require inverse kinematics and stable gaits with different step lengths and stride frequencies can be readily generated by changing the value of only one parameter in the TFS.

Taking the advantages of TFS as a model-free approach, we implemented a TFS in a simulated humanoid robot to generate gait trajectories in three dimensions. In this novel approach, the Particle Swarm Optimization (PSO) technique with constraint handling on angles and time is used to find optimum parameters of TFS and train the robot to achieve fast bipedal walking for the first time.

To overcome inherent noise of the simulator, Resampling algorithm is implied which could lead to robustness in nondeterministic environments. The Genetic Algorithm (GA) is also implemented in the same manner. Learning results of GA and PSO are compared with each other which indicate PSO as a better learning method for this complex problem in non-deterministic environment.

2 Simulator and Biped Model

In this paper, a new approach for walking behavior in a simulated humanoid robot is discussed. However simulation is not always efficient, due to difficulty of the modeling collision between feet and the ground, we still believe that numerical simulation is sufficient to explore and test bipedal locomotion methods.

The simulation is performed by Rcssserver3d simulator which is a generic three-dimensional simulator based on Spark and 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]. The robot in this study is a simulated model of NAO that is a real humanoid Robot with two arms, two legs and a head. This robot weighs 4.5kg, stands 57cm high and has 22 degrees of freedom (DOF). There are six DOFs in each leg; two in the hip, two in the ankle and one at the knee. An additional DOF that exists at each leg's hip for yaw causes the legs to rotate outward and inward.

As an appropriate test-bed, in our soccer simulation team MRL we have implemented and tested our new bipedal locomotion approach on simulated NAO robot how the generated software based on this simulator is developed by MRL team from scratch. According to our studies, we found 6 DOFs (three for each leg) more

effective than other DOFs to make the robot capable of fast walking. The DOFs of hip, knee and ankle which move on the same plane of forward-backward are the major ones. Although other DOFs are effective in walking behavior, but in fact, their role smoothes the robots walking motion. So here, it's preferred to ignore them to decrease learning search space. Like in [10], Foot was kept parallel to the ground by using ankle joint in order to avoid collision. Therefore ankle trajectory can be calculated by hip and, knee trajectories and its DOF parameters are eliminated.

3 TFS gait Generator

Bipedal walking as a complex motion, involves most of humanoid robot's joints. Researchers attempt to imitate the human walking style as well as its speed. Therefore analyzing human walk pattern has been used for acquiring beneficial information about this motion. Human walk has been investigated from many angles; walking trajectory is one of them. The walking trajectory is divided into several types. Positional trajectory and angular trajectory are two of them. In angular trajectory, the angle of each joint is plotted at a certain time slice. Therefore the angular trajectory is obtained by angular variation of each joint. Biped angular trajectory of two joints; hip and knee captured from human walking are shown in Fig 1.a [11].

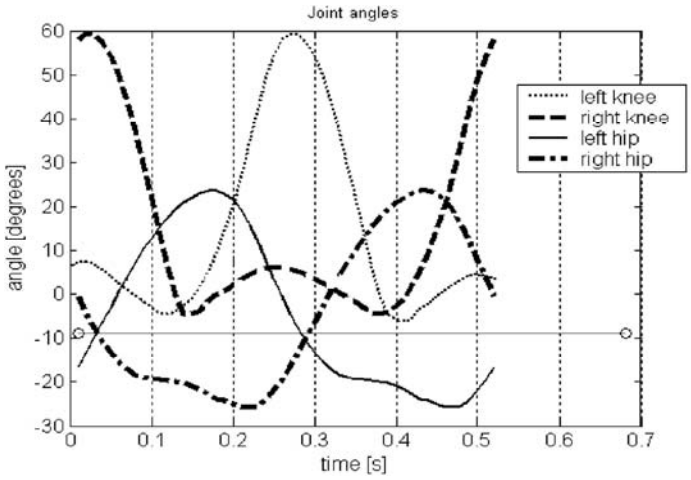

Fig. 1.a Human walking angular trajectory [11]

The angle of each joint in one period of walking signal from t_0 to t_6 is represented in fig 1.b [11] by capturing the main features of fig 1.a and gives a general form to make it applicable to robots. In time range $[t_0, t_2]$ and $[t_5, t_6]$ the left leg is support leg and the right one is swing leg but in range of time $[t_2, t_5]$ the left and right legs play the role of support and swing legs respectively. In another word, in

two times of t_2 and t_5 the roles of two legs are switched with each other. At time t_3 where two hip trajectories intersect, two thighs cross each other.

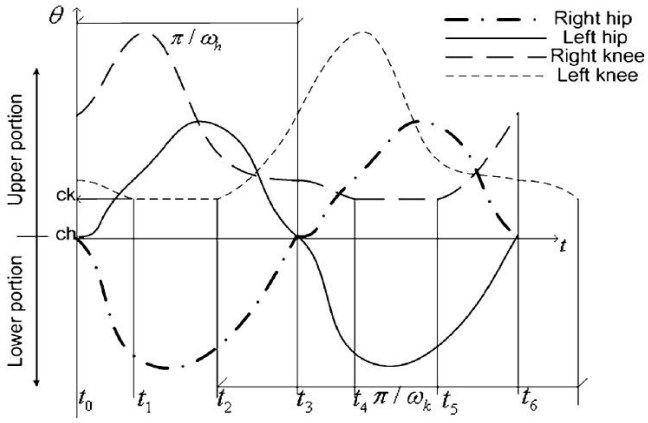

Fig. 1.b gaits elaborated from human gaits features [11]

3.1 Angular trajectory generation

Regarding the fact that all joint trajectories of human walking are periodic and similar to sine or cosine signals [12], the generation of these angular signals can be done by Fourier series.

3.2 Basic Fourier series

The original definition of Fourier series is described by following formula:

$$F(t) = \frac{a_0}{2} + \sum_{i=1} \left(a_i \cos \frac{i\pi t}{L} + b_i \sin \frac{i\pi t}{L} \right) \quad \textbf{(1)}$$

The first term $(\frac{a_0}{2})$ of equation 1 represents the DC bias a_0 of the signal and the L represents half of the largest period that exists in the signal. By $\omega = \frac{\pi}{L}$ then the frequency form of Fourier series is achieved as follows:

$$F(t) = \frac{a_0}{2} + \sum_{i=1} (a_i \, cos(i\omega t) + b_i \, sin(i\omega t)) \quad \textbf{(2)}$$

Where ω is frequency of periodic signal, any complicated signal can be produced by this formula when i is considered infinite. But when the value of i is limited to a definite number, precision of generating signal is reduced and this type of Fourier series is called partial sum of the Fourier series. According to fig. 1.a, Human Walking angular trajectories are too complicated to be produced by a definite Fourier series band limited to the second harmonic. Therefore a modified definite Fourier series as a Truncated Fourier series (TFS) is used in this study.

3.3 Trajectory generation by using TFS

According to Fig. 1.b., the signals are divided in two parts; upper portion and lower portion. Whereby each portion can be assumed as an odd function, the cosine part of TFS is eliminated. So the TFS is reduced to equation 3 to generate each portion of trajectory.

$$F(t) = a + \sum_{i=1}^{n} b_i \sin(i\omega t) \tag{3}$$

Where ω is fundamental frequency of signal and a is signal offset. Separate production for each portion, caused to generate complex signals with different upper and lower portions. The number of parameters for generating these complex signals is also less than the parameters used in Fourier series. As shown in Fig. 1.b., each signal has an offset. C_h and C_k are hip trajectory and knee trajectory offsets respectively. From t_0 to t_2 the left leg is considered as supporting leg and the variation of its knee angle is so minute that can be assumed fixed. This duration of walking is named lock phase. In addition, the amount of shift phase of the two leg trajectories signal is half of the period of each signal. The trajectories for both legs are identical in shape but are shifted half of the walking period in time. Therefore by figuring out walking angular trajectory of one leg the other leg trajectory is obtained. Using (3) and considering curves of Fig. 1.b., the TFS for generating each portion of hip and knee trajectories are formulated as follow (4):

$$\begin{gathered}
\theta_k^+ = \sum_{i=1}^{n} C_i.\sin(i\omega_k t_2) + C_k\,,\omega_k = \frac{2\pi}{T_k} \\
\theta_k^- = C_k \geq 0 \\
\theta_h^+ = \sum_{i=1}^{n} A_i.\sin(i\omega_h t_3) + C_h\,,\omega_h = \omega_k \\
\theta_h^- = \sum_{i=1}^{n} B_i.\sin(i\omega_h t_6) + C_h\,,\omega_h = \omega_k
\end{gathered} \tag{4}$$

In these equations, the plus (+) sign represents the upper portion of walking trajectory and the minus (-) shows the lower portion. A_i, B_i and C_i are constant

coefficients for generating signals. The h and k index stands for hip and knee respectively. C_h *and* C_k are signal offsets and T_k is assumed as period of knee trajectory. Considering the fact that all joints in walking motion have equal movement frequency [12], the equation $\omega_k = \omega_h = \frac{2\pi}{T_k}$ can be concluded. Parameter t_3 shows the end time of hip trajectory in upper portion and starts its down portion, t_6 shows the end time in down portion. These parameters are not significant since they can be obtained when the hip trajectory intersects the C_h line. But parameter t_2 represents the end time of knee lock phase and must be considered to produce knee trajectory. Therefore Truncated Fourier series parameters to produce trajectories are; C_h, C_k, A_i, B_i, C_i, t_2, *and* W_k. In this essay there are some constraints to be dealt with as shown in the following equation:

$$0 < t_2 < T_k \, , \omega_k = \frac{2\pi}{T_k} \Rightarrow 0 < t_2 < \frac{2\pi}{\omega_k} \qquad C_k \geq 0 \tag{5}$$

Finally an optimization algorithm is needed to optimize a 7_dimension Problem for finding the best gait generator.

4 PSO algorithm

The PSO algorithm consists of three steps; generating primitive particle's positions and velocities, velocity update and position update [13]. These parts will be described in sections 4.1, 4.2 and 4.3 respectively.

4.1 Initializing particles' positions and velocities

Equations (6) and (7) are used to initialize particles which Δt are the constant time increment. Using upper and lower bounds on the design variables values, X_{min} and X_{max}, the positions, X_k^i and velocities, V_k^i of the initial swarm of particles can be first generated randomly. The swarm size will be denoted by N. The positions and velocities are given in a vector format where the superscript and subscript denote the i^{th} particle at time k.

$$X_0^i = X_{min} + Rand(X_{max} - X_{min}) \tag{6}$$

$$V_0^i = \frac{X_{min} + Rand(X_{max} - X_{min})}{\Delta t} = \frac{Position}{time} \tag{7}$$

4.3 Updating Velocities

The fitness function value of a particle is used to determine the particle which has the best global value in the current swarm (P_k^g), and to determine the best position of each particle over time (P^i).

The three values that affect the new search direction, namely, current motion, particle own memory, and swarm influence, are incorporated via a summation approach as shown in Equation below with three weight factors, namely, inertia factor, w , self confidence factor,C_1 , and swarm confidence factor,C_2 ,respectively.

$$\underbrace{V_{k+1}^{\ i}}_{\substack{Velocity\ of\ Particle \\ i\ at\ time\ k+1}} = \overbrace{w}^{[0.4,1.4]} \underbrace{V_k^{\ i}}_{\substack{Current \\ Motion}} + \overbrace{C}^{[1,2]}_1 \underbrace{Rand\ \frac{(P^i - X_k^{\ i})}{\Delta t}}_{Particle\ Memory\ Influence} + \overbrace{C}^{[1.5,2]}_2 \underbrace{Rand\ \frac{(P_k^{\ g} - X_k^{\ i})}{\Delta t}}_{Swarm\ Influence} \quad (8)$$

The inertia weight w controls how much of the previous velocity should be retained from the previous step. A larger inertia weight facilitates a global search, while a smaller inertia weight facilitates a local search [14]. A balance can be achieved between global and local exploration to speed up search results using a dynamically adjustable inertia weight formulation. There have been different strategies for determining the value dynamic inertia weight. Introducing a nonlinear decreasing inertia weight as a dynamic inertia weight into the original PSO significantly improves its performance through the parameter study of inertia weight [14]. This nonlinear distribution of inertia weight is expressed as follow:

$$w = w_{init} + U^{-k} \quad (9)$$

Where w_{init} is the initial inertia weight value selected in the range [0, 1] and U is a constant value in the range [1.0001, 1.005], and k is the iteration number.

4.4 Updating the Position

Position update is the final step of each iteration and it is done by using the current particle position and its own updated velocity vector shown in the Equation below.

$$X_{k+1}^i = X_k^i + V_{k+1}^i \Delta t \quad (10)$$

In summary, the PSO technique will be:

Let initialization iterative number k = 0, initialization population size.(6),(7) Calculate each particle's fitness value of initialization population, and let first generation Pi be initialization particles, and choose the particle with the best fitness value of all particles as the $P_1^{\ g}$.

Repeat

For each particle

Calculate inertia Weight according to equation (9).

Update the velocities according to equation (8).
Update the positions according to equation (10).
Evaluate its fitness value according to the objective function.
Update P_k^g and P^i if necessary.

End for

Until a sufficient good criterion is met, either good fitness or a maximum number of iterations (As in genetic algorithm).

5 Implementation

Bipedal walking is known as a complicated motion since many factors affect Walking style and stability such as robot's Kinematics and dynamics, and collision between feet and the ground. In such a complex motion, relation between Gait trajectory and walking characteristic is nonlinear. In this approach the best parameters to generate angular trajectories for bipedal locomotion must be found. This kind of optimization problem is usually difficult; therefore particle swarm optimization (PSO) seems to be appropriate solution.

In PSO, the parameters of the problems are coded into a finite length of string as a particle. According to section 2, TFS has 7 parameters to generate all joints angular trajectories; there is a 7-dimension search space for the PSO to find the optimum solution.

Fitness function has a critical rule in PSO that is used to judge whether a solution represented by a particle is good or bad. Angular trajectory produced by each particle is used for walking by simulated robot. To use angular trajectory for walking, all individual robot's joints attempt to drive towards their target angles using proportional derivative (PD) controllers. To equip the robot with a fast walking skill a fitness function based on robot's straight movement with limited action time is considered. First the robot is initialized in x=y=0 (0, 0) to walk for 15 seconds then fitness function is calculated whenever robot falls or time duration for walking is over. Fitness function formulation is expressed as follow; The *Current Time* in the formula determines time passed since robot has started walking:

```
If ((Current Time >= time duration for walking) or
(robot is fallen))
Fitness := 10*x ;
End if
```

Due to the fact that there is noise in simulated robot's actuators and sensors, training walking task in this approach can be viewed as an optimization problem in a noisy and nondeterministic environment. Resampling is one of the techniques to improve the performance of evolutionary algorithms (EAs) in noisy environment [15]. In Resampling, the individual set of parameters (particle)y_i, the fitness $F(y_i)$ is measured m times and averaged yielding fitness. According to (11) the strength of noise $\overline{F}$ is reduced by the factor$\sqrt{m}$.

$$\overline{F(y_i)} = \frac{1}{m}\sum_{i=1}^{m} F(y_i), \quad y(i) = const \implies \overline{\sigma_e} = \sqrt{Var\left[\overline{F(y_i)}\right]} = \frac{\sigma_e}{\sqrt{m}} \tag{11}$$

In this study, for comparing GA and PSO performance as an optimizer, we implemented them by the same mentioned model, fitness function and Resampling factor of *m* as 3.

5.1 PSO and GA implementation

Since particles may not be satisfied in constraints during updating position procedure constraint handling is a vital step in PSO algorithm. There are many constraints on parameters in this study (i.e time parameters in TFS must be positive). Therefore Pareto [16] with multi-objective modeling is used for handling constraints.

In Pareto, a solution, *x(2)*, is dominated by solution, *x(1)*, if *x(1)* is not worse than *x(2)* in all objectives, and for at least one of the objectives, *x(1)* is strictly better than *x(2)*. Without loss of generality, these conditions can be expressed as follows for the case where all of the objective functions are going to be minimized:

$fm\left(x\left(1\right)\right) \le fm\left(x\left(2\right)\right) for \forall m = 1, 2, ..., M$ and
$fm\left(x\left(1\right)\right) \prec fm\left(x\left(2\right)\right)$ for some *m*.

Each constraint is assumed as an object in which parameters must be satisfied .So according to Pareto method, a particle can be considered to find P_i , P_k^g when it satisfies objects and constraints. So calculating fitness for particles that cannot satisfy constraint is not necessary.

Salman et.al [17] used the values of 0.9, 2 and 2 for *w*, C_1 and C_2 respectively. But it is possible that much combination of values lead to much slower convergence or even non-convergence. The tuning of the PSO algorithm values is an issue that warrants proper investigation but is outside the scope of this work. We considered various values for each parameter of the algorithm and tried all possible combinations. Finally we chose the best combination of the parameters regarding the dynamic inertia weight and test results that C_1 and C_2 are assumed as 1, 1.5, w_{init} as 0.8, U as 1.0002 and Δt as 1, respectively. We have also implied a swarm consisted of 100 particles (N = 100) and maximum iteration of 10.

In GA implementation, the crossover rate and mutation rate are set to 0.8 and 0.06 respectively and roulette wheel is assumed as selection method. Population for each generation is 100, termination condition is to have a generation counter greater than 10 and Resampling *m* factor is 3. In another world 3000 trials are needed to find appropriate TFS parameters.

6 Results

4 hours after starting GA on a Pentium IV 3 GHz Core 2 Duo machine with 2 GB of physical memory, 3000 trials were performed. The robot could walk 6.7m in 15s with average body speed of 0.45m/s and the period of 0.41s for each step. Fig. 3 shows the average and best fitness values during these 10 generations.

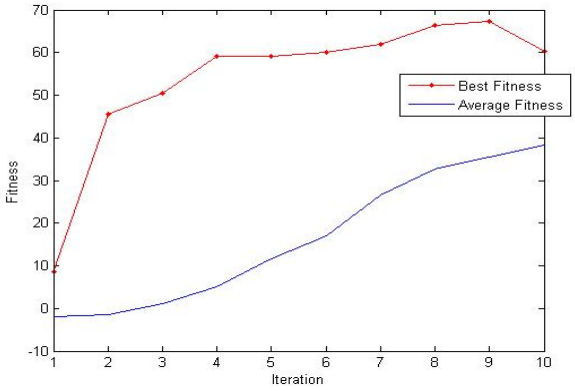

Fig 3. GA Convergence

Running the PSO on the same system with the same parameters of iteration and population, more satisfactory results are achieved. Implying constraint handling, some of the particles that did not satisfy constraint were not tested. So through PSO after 1782 training tests instead of 3000 by GA, the robot could walk 8.7 m in 15 s with average body speed of 5.8 m/s, that's significantly better than GA result. This outcome also proves that PSO has bypassed a local minimum that GA was caught in and it can optimize faster. Fig. 4 illustrates PSO algorithm convergence results.

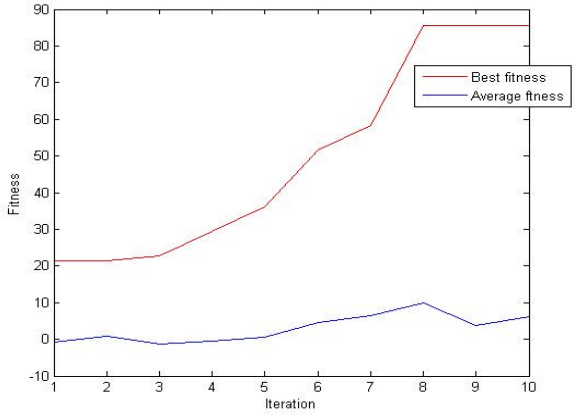

Fig 4. PSO Convergence

7 Conclusion

In this study for the first time TFS with PSO is implemented in a simulated robot that can walk fast and stable. The technique has some advantages. First, it can be implemented on many humanoid robots as simulated NAO robot to walk based on its walking performance without considering any mathematical modeling. Second, the modified PSO converges sooner than GA to find the best TFS parameters. Since each individual or particle needs a long time to be tested, the higher speed of PSO convergence becomes more significant. On the other hand by using PSO the robot has achieved a faster walk that means PSO performs better than GA in such problems. Resampling technique is also used to overcome uncertainty and noise of the environment.

References

1. Vukobratovic, M., Borovac, B., Surdilovic. D. :Zero-moment point proper interpretation and new applications. In: Proceedings of the 2nd IEEE-RAS International Conference on Humanoid Robots, pp. 237--244 (2001).
2. Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., Hirukawa. H.: The 3D linear inverted pendulum mode A simple modeling for a biped walking pattern generation. In: Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 239–246 (2001)
3. McGeer, T. :Passive dynamic walking. International Journal of Robotics Research, Vol. 9(2), pp. 62--82 (1990).
4. M.A,Pinto C. ,Golubitsky, M.:Central Pattern Generator for Bipedal locomption. J. Math. Biol. 53, pp. 474—489, (2006)
5. Mochon, S., McMahon, T.A.:Ballistic walking. J. Biomech. 13, pp. 49--57 (1980)
6. Buchli, J., Iida, F., Ijspeert, A.J.:Finding Resonance: Adaptive Frequency scillators for Dynamic Legged Locomotion. In: Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. PP. 3903--3910 , (2006)
7. Matsuoka, K.:Sustained oscillations generated by mutually inhibiting neurons with adaptation. Biol. Cybern. 52, 367--376 (1985)
8. Yang, L., Chew, C.M., Poo, A.N.:Adjustable Bipedal Gait Generation using Genetic Algourithm Optimized fourier Series Furmulation. In: Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, PP. 4435-4440 (2006)
9. Boedecker, J.:Humanoid Robot Simulation and Walking Behaviour Development in the Spark Simulator Framework. Technical report, Artificial Intelligence Research University of Koblenz (2005)
10. Kagami, S., Mochimaru, M., Ehara, Y., Miyata, N., Nishiwaki, K., Kanade, T., Inoue, H.: Measurement and comparison of humanoid H7 walking with human being. Robotics and Autonomous Sys, vol. 48, pp. 177—187 (2003)
11. Yang, L., Chew, C.M., Zielinska, T., Poo, A.N.:A Uniform Bipedal Gait Generator with Offline Optimization and Online Adjustable Parameters. Robotica, Cambridge University Press, vol. 25, pp. 549--565 (2007)
12. Schot, P.K., Decker, M.J.:The force driven harmonic oscillator model accurately predicts the preferred stride frequency for backward walking. In: Human movement science. vol. 17, pp. 67--76 (1998)

13.Shi Y., Eberhart RC.:Parameter selection in particle swarm optimization. In: Evolutionary programming VII proceedings of the seventh annual conference on evolutionary programming, pp. 591--600. New York (1998)
14.Jiao, B., Lian, Z., Gu X.:A dynamic inertia weight particle swarm optimization algorithm. j Chaos. vol. 37, pp. 698--705 (2008)
15.Beyer, H.G.;Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice. Comput. Methods Appl. Mech. Engrg., vol 186, pp. 239--267 (2000)
16.Coello, C. A., Pulido, G. T.,Lechuga, M.S. :Handling Multiple Objectives With Particle Swarm Optimization. IEEE Transaction on Evolutionary Computation. vol. 8. O. 3, pp. 256--279 (2004)
17.Salman, A., Ahmad, I., Al-Madani, S.,: particle swarm optimization for task assignment problem. Microprocessorsand Microsystems, vol. 26, PP. 363–371 (2002)