Chapter 2
Biped Locomotion with Arm Swing, Based on Truncated Fourier Series and Bees Algorithm Optimizer

Ebrahim Yazdi, Vahid Azizi, and Abolfazle T. Haghighat

2.1 Introduction

Bipedal walking is a difficult task due to its intrinsic instability and developing successful controller architectures for this mode of locomotion has proved substantially more difficult than for other types of walking (Beer et al. 1998). This type of walking has been tackled from different directions that all of them have been divided to two major approaches of static walking (Kato et al. 1974) and dynamic walking (Takanishi et al. 1982).

Static walking assumes that the robot is statically stable. This means that, at any time, if all motion is stopped the robot will stay indefinitely in a stable position. It is necessary that the projection of the center of gravity of the robot on the ground must be contained within the foot support area. This approach was abandoned because only slow walking speeds could be achieved, and only on flat surfaces.

Biped dynamic walking allows the center of gravity to be outside the support region for limited amounts of time. There is no absolute criterion that determines whether the dynamic walking is stable or not. Indeed a walker can be designed to recover from different kinds of instabilities(Hodgins and Raibert 1990). However, if the robot has active ankle joints and always keeps at least one foot flat on the ground then the Zero Momentum Point (ZMP) can be used as a stability criterion.

There are two major approaches, model-based and model-free, in Dynamic walking researches. In model-based approaches, controller of robot is dependent on model of robot and from one robot to another everything in controller should be changed. Two well-known methods in this approach are “Zero Moment Point” (Zhang et al. 1999; Vukobratovic et al. 2001) (ZMP) and “Inverted Pendulum” (Kajita et al. 2001). ZMP specifies the point with respect to which dynamic reaction

E. Yazdi (✉)
Department of Electerical & Computer Engineering,
Mechatronics Research Labrotoary (MRL), Qazvin Azad University, Qazvin, Iran
e-mail: e.yazdi@qiau.ac.ir

S.-I. Ao et al. (eds.), *Intelligent Automation and Systems Engineering*, Lecture Notes in Electrical Engineering 103, DOI 10.1007/978-1-4614-0373-9_2,

force at the contact of the foot with the ground does not produce any moment, i.e. the point where total inertia force equals 0 (zero).The ZMP is no longer meaningful if the robot makes multiple non-planar contacts. In Inverted Pendulum, walking is often likened to the motion of two coupled pendulum, because the swing leg behaves like an inverted pendulum moving about the stance foot and like a regular pendulum swinging about the hip.

In model-free approach, 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) (McGeer 1980), Ballistic Walking (Mochon and McMahon 1980) and Central Pattern Generator (CPG).

Passive dynamics as an approach to robotic movement control (especially walking), is based on utilizing the momentum of swinging limbs for greater efficiency. This method is based on using the morphology of a mechanical system as a basis for necessary controls. Ballistic Walking is using from Lagrange equations. Central Pattern Generator (CPG) as a model-Free approach uses from a set of neural oscillators for controlling robot and uses genetic algorithm as a weight optimizer.

In this paper, a model free approach described, a Truncated Fourier Series (TFS) formulation has been used for controlling of robot. TFS was used in 2006 for the first time for gait generation in bipedal locomotion (Yang et al. 2006). TFS together with a ZMP stability indicator are used to prove that TFS can generate suitable angular trajectories for controlling bipedal locomotion. In the TFS model applications for 2D walking, three key parameters determine the locomotion: the fundamental frequency which determines the pace of walking, the amplitude of the functions which determines the stride, and the constant terms used to adjust to different inclinations of the terrain (Yang et al. 2007). Humans naturally swing their arms when they walk or run. Although arm swing has often been compared with pendulum motion, it is not a purely passive phenomenon. Muscle activity controls arm swing magnitude and timing during human locomotion (Hinrichs 1990). Elftman first proposed that arm swing during walking balances torso torques caused by swinging of the lower limbs (Elftman 1939). This idea has been studied further by others with the same general conclusions (Hinrichs 1990; Li et al. 2001). Our approach is also capable to produce hand angular trajectories with emphasizing the role of hands in smooth and robust walking.

In this approach, the Bees Algorithm (BA) (Pham et al. 2006) technique with constraint handling on angles and time is used to find optimum parameters of TFS and train the robot to achieve fast bipedal forward and backward walking for the first time.

2.2 Simulator and Robot Model

The target Robot of our study is a 22-DOF (degrees of freedom) NAO robot with 4.5 Kg weight and 57 Cm stand height. The robot has four DOF for each arm, six DOF for each leg and has a head with 2 degrees of freedom.

Table 2.1 Morphological major parameters of the NAO robot

Joint name	Motion	Range
Ankle	Front and back (Y)	−75–55
Hip	Front and back (Y)	−100–25
Knee	Front and back (Y)	−130–0
Shoulder	Front and back (Y)	−120–120

The simulation performed by Rcssserver3d simulator which is a generic three dimensional simulator based on Spark and Open Dynamics Engine (ODE) Smith (http://www.ode.org). 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 (Boedecker 2005). The time-integrated simulation is processed with a resolution of 50 simulation steps per second.

In this approach, the body trunk of the robot is not actuated. Our experiences show, the leg and arm joints are more effective joints for walking, that the joints of hip, knee, ankle and shoulder which move on the same plane of forward-backward are the major ones and these joint have restricted degree (Table 2.1 shows description of them).

2.3 Joint Angular Trajectory

Human motions are recognized flexible and periodic but more challenging for motion stability issue. Therefore, human-like motion patterns are included into the research objectives. The walking trajectory is divided into several types. Positional trajectory and angular trajectory are two of them. In this paper angular trajectory has been used.

2.3.1 Lower Body Joint Angular Trajectory

Similar to (Kagami et al. 2003) Foot was kept parallel to the ground by using ankle joint in order to avoid collision. Therefore ankle trajectory can be calculated by adding hip and knee and multiply with (−1), so ankle DOF parameters are eliminated. Successful walking is defined as an acceptable motion tracking based on the optimized walking (Hinrichs 1990). If a gait period divides to six time slice, (2.1) and (2.2) formulate the joint control in the smooth plane. The following notations have been used for variables in the joint angular trajectory formulations:

1. Subscripts 1 and 2 refers to the joint trajectory from the beginning of swing and stand phase respectively.
2. Subscripts h and k denote the hip and knee respectively.
3. Subscripts l, r, lo refers to left leg, right leg and lock phase respectively.

$$\begin{aligned} \theta_{kl} &= \theta_{lo} \quad [t_0, t_2] \\ \theta_{kl} &= \theta_{kl} \quad [t_2, t_6] \\ \theta_{kr} &= \theta_{k1} \quad [t_0, t_4] \\ \theta_{kr} &= \theta_{lo} \quad [t_4, t_6] \end{aligned} \tag{2.1}$$

$$\begin{aligned} \theta_{hl} &= \theta_{h2} \quad [t_0, t_3] \\ \theta_{hl} &= \theta_{h1} \quad [t_3, t_6] \\ \theta_{hr} &= \theta_{h1} \quad [t_0, t_3] \\ \theta_{hr} &= \theta_{h2} \quad [t_3, t_6] \end{aligned} \tag{2.2}$$

According to (2.1) and (2.2) we can realize that the trajectories for both legs are identical but they have been shifted in time relative to each other by half of the walking period. The joint angle trajectories can be separately looked by offsets. The values of the defined offsets actually influence the biped's posture during walking. Note that in lock phase knee joints only have those offsets for angular trajectory. It can be note that we assume left leg is stand leg in first half period.

2.3.2 Upper Body Joint Angular Trajectory

As humans change walking speed, their nervous systems adapt muscle activation patterns to modify arm swinging for the appropriate frequency. Humans have neural connections between their upper limbs and lower limbs which coordinate muscle activation patterns during locomotors tasks. Mechanical analysis indicates that arm swing during human locomotion helps to stabilize rotational body motion (Zehr and Duysens 2004). During human walking, the arms normally swing in opposite manner to legs, which helps to balance the angular momentum generated in the lower body (Hinrichs 1990; Elftman 1939). Humans swing their arms close to 180° out of phase with their Respective legs during walking (Collins et al. 2001). It means that left arm swing opposite of right leg and right arm pendulum opposite manner of left leg.

2.4 Truncated Fourier Series Motion Generator

In (2.3) the original Fourier series of periodic function F (t) has been written:

$$f(t) = \frac{1}{2}a_0 + \sum_{i=1}^{\infty} a_i \sin\left(\frac{2\pi i}{T}t\right) + \sum_{i=1}^{\infty} b_i \cos\left(\frac{2\pi i}{T}t\right) \tag{2.3}$$

a_i and b_i are constant coefficients and T is the time period. When i is infinite value this formula can produce any complicated signal. But when the value of i is definite accuracy of signal is decrease. Because, walking angular trajectories are too complicated signals, this equation cannot create true signals with definite i. Therefore a modified definite Fourier series as a Truncated Fourier series (TFS) has been used as follow:

$$f(t) = \sum_{i=1}^{n} a_i \sin(i\omega t) + c_f \tag{2.4}$$

Where a_i and c_f are constants that should be determined and ω is the fundamental frequency determined by the desired period of the gait. n determines the number of terms and can be chosen as a trade-off between the accuracy of the approximation required and the computational load. Note that c_f is an offset of angular trajectories that we talked about it before. With this approach and (2.1) and (2.2), the TFS for hip-pitch and knee-pitch angles are formulated as follow:

$$\begin{aligned}
\theta_{h1} &= \sum_{i=1}^{n} B_i \sin(i\omega_h t_{h1}) + c_h, \omega_h = \frac{2\pi}{T_h} \\
\theta_{h2} &= \sum_{i=1}^{n} A_i \sin(i\omega_h t_{h2}) + c_h \\
\theta_{k1} &= \sum_{i=1}^{n} C_i \sin(i\omega_h t_{k1}) + c_k, \omega_k = \frac{2\pi}{T_k} \\
\theta_{klo} &= c_k, c_k \geq 0
\end{aligned} \tag{2.5}$$

In these equations A_i, B_i and C_i are constant coefficients. C_k and C_h are signal offsets and T_k and T_h are assumed as period of knee and hip trajectory respectively. With reliance to the fact that all joints in walking motion have equal movement frequency we can assume $T_k = T_h$. T_{h1}, T_{h2} show the end time of hip stand trajectory and hip swing trajectory respectively and their values can be calculated with half period of walking. T_{k1} represents the end time of the lock phase that computed with optimizer.

As we said arm, swing in opposite manner of leg, so we can use from hip motion generator to control arm. But according to this fact that wave length of arm joint angle trajectories and hip joint angle trajectories are different and according to this truth that angle of arms is different from angle of hip in start of walking, joint angular motion generator for arms can be written as (2.6).

$$\begin{aligned}
\theta_{h1} &= \sum_{i=1}^{n} D_i \sin(i\omega_h t_{s1}) + c_a, \omega_h = \frac{2\pi}{T_s} \\
\theta_{h2} &= \sum_{i=1}^{n} E_i \sin(i\omega_h t_{s2}) + c_a
\end{aligned} \tag{2.6}$$

In these equations D_i, E_i are constant coefficients. c_a is a signal offset and T_s assumed as period of arm trajectory that it is equal with $T_h.t_{s1}$ and t_{s2} show the end time of arm stand trajectory and arm swing trajectory respectively and their values are exactly equal to t_{h1} and t_{h2}.

2.5 Bees Algorithm

Bees Algorithm (BA) which is a nature-inspired algorithm, mimicking the food foraging behavior of swarms of honey bees. It is developed by Pham and et al. recently. The BA algorithm is simple in concept, few in parameters, and easy in implementation, it has been successfully applied to various benchmarking and real-world. The algorithm requires a number of parameters to be set, namely: number of scout bees (n), number of elite bees (e), number of patches selected out of n visited points (m), number of bees recruited for patches visited by "elite bees" (nep), number of bees recruited for the other (m-e) selected patches (nsp), and size of patches (ngh). The pseudo code of the BA is as follows:

1. Initialize population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met)
4. Select sites for neighborhood search.
5. Recruit bees for selected sites (more bees for better sites) and evaluate fitness.
6. Select the fittest bee from each patch.
7. Assign remaining bees to search randomly and evaluate their fitness.
8. End while

In Step 1, the BA matrix is filled with as many randomly generated solution vectors as the m. Their results calculate in Step 2. In Step 4, bees that have the highest fitness are chosen as "selected bees" and sites visited by them are chosen for neighborhood search. Then, in Steps 5 and 6, the algorithm conducts searches in the neighborhood of the selected sites, assigning more bees to search near to the best e sites. The bees can be chosen directly according to the fitness associated with the sites they are visiting. Step 7; assign the remaining bees in the population, randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met.

2.6 Applying Bees Algorithm

With the rule base fixed, the Bees Algorithm was used to tune the parameters of the input and output membership functions and the scaling gains for the input and output variables.

Table 2.2 Lower band and upper band that used for harmony memory initialization

Parameter	Lower bound	Upper bound
A	0	40
B	−40	0
C	−40	0
D	0	40
E	−40	0
C_h	−10	30
C_k	−40	0
C_a	−40	0
t_k	0.01	1
t_{k1}	0.01	1

In theory, each bee is a vector comprising 10 real numbers. Eight of those numbers have been reserved for constant parameters and two have been reserved for the parameter variable of time period and T_{k1}. Bees are generated randomly and uniformly for the first iteration between lower and upper bound. In this study the lower and upper bound data for initialization are depicted in Table 2.2.

The following parameter values are set for optimization, population (n = 100), number of selected sites (m = 15), number of elite sites (e = 5), initial patch size (ngh = 2) for coefficients and (ngh = 0.04) for Time period, number bees around elite points (nep = 10), number of bees around other selected points (nsp = 3). Note that ngh defines the initial size of the neighborhood in which follower bees are placed.

Fitness function has a critical role in BA and is used to judge how well a solution represented by a Bee is. To achieve more stable and faster walk, a fitness function based on robot's straight movement with having limited time for walking is assumed. The amount of deviation from straight walking is subtracted from the fitness as a punishment to force the robot to walk straight. We ran the simulator for 15 s, first the robot is initialized its X and Y values equal to zero and fitness function is calculated whenever robot falls or time duration for walking is over. Equation 2.7 shows the pseudo code of computing of fitness function in forward walking respectively.

$$
\begin{aligned}
& \mathit{if}\,(\mathit{CurrentTime} \geq \mathit{TimeDuration})\{ \\
& \quad \mathit{fitness} = X - Y; \\
& \quad \mathit{if}\,(\mathit{RoboIsFallen}) \\
& \qquad \mathit{fitness} = \frac{X - Y}{\mathit{TimeDuration} - \mathit{CurrentTime}}; \\
& \} \\
& \mathit{else} \\
& \quad \mathit{fitness} = -10000;
\end{aligned} \tag{2.7}
$$

When robot falls down during walk the fitness is divided to remaining time of simulation. This punishment forces the robot to achieve a stable walk.

2.7 Result

We ran the simulator on a Pentium IV 3 GHz Core 2.6 Duo machine with 4 GB of physical memory, with Linux SUSE 10.3. The time period for the simulation was 15s. About 2 h after starting BA under the MATLAB, 800 trials were performed. Figure 2.1 shows the average and best fitness values during these eight generations.

In our previous work (Yazdi et al. 2010) without swing of arm during walking robot could walk only 4.2 m in 15 s, Figs. 2.2 and 2.3 shows the result of learning without arm swing.

Gait period at the best found fitness equals to 0.54 s. according to this consequence Fig. 2.4 shows angular trajectories of hip, knee, ankle, and arm generated by oscillators.

In human walking left leg and right leg angular trajectories have half period difference to occur, Figs. 2.5 and 2.6 shows this truth for hip and knee joint angular trajectories respectively.

Arms swing is in same manner of their opposite hips with difference in wave length. Figure 2.7 shows that arm's wave length is about twice greater than hip's wave length in same manner.

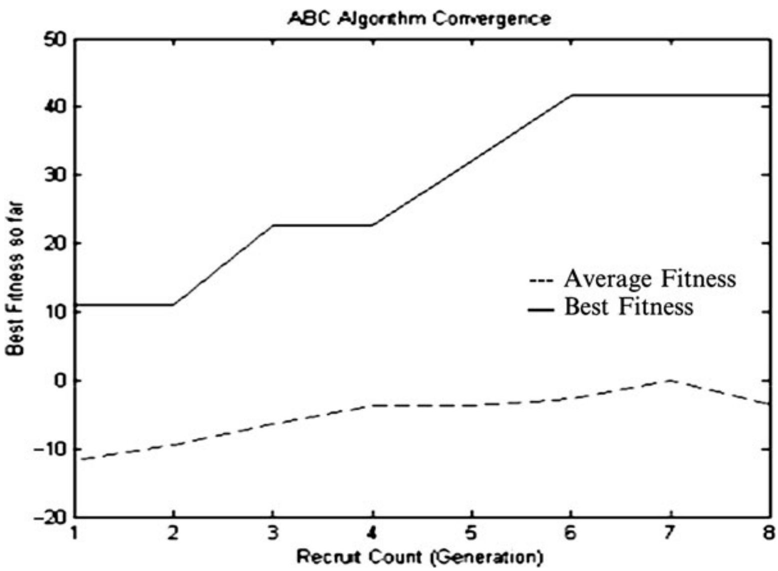

Fig. 2.1 BA convergence, with using hand during walking robot could walk 6.8 m in 15 s with average body speed 0.45

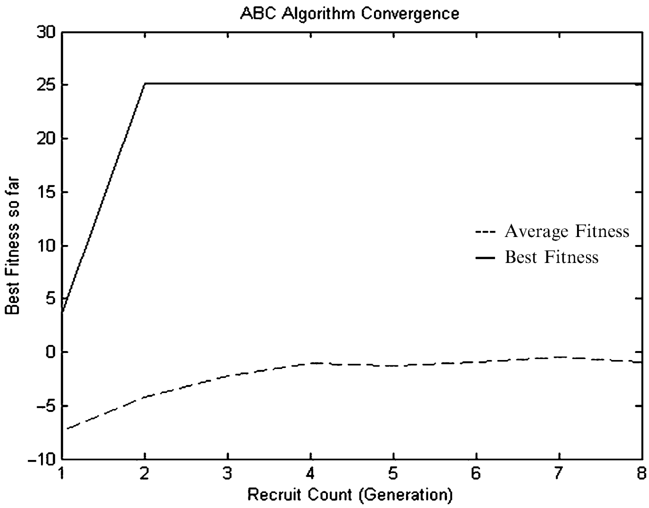

Fig. 2.2 BA convergence, without swing hands during walking robot could walk only 4.2 m in 15 s with average body speed 0.28

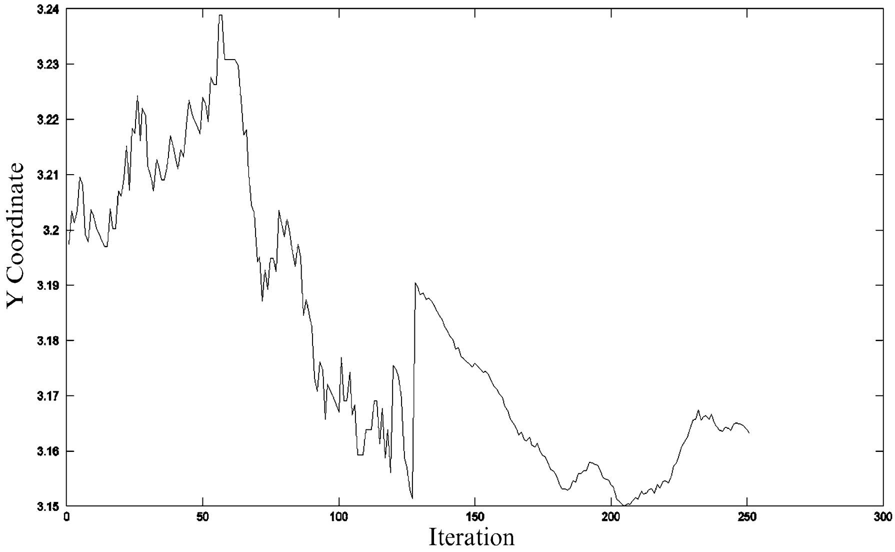

Fig. 2.3 Shows y coordinate of robot during walking with and without arm swing, and from this figure we can result that robot could walking straighter with arm swing

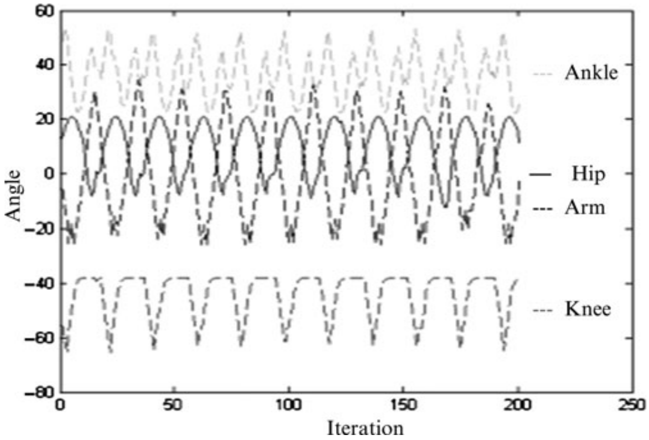

Fig. 2.4 In sequence angular trajectories of Left ankle, Left hip, Left arm and Left knee joints during walking. Arm swing in opposite manner of its respective leg. Every 50 iterations equal to 1 s

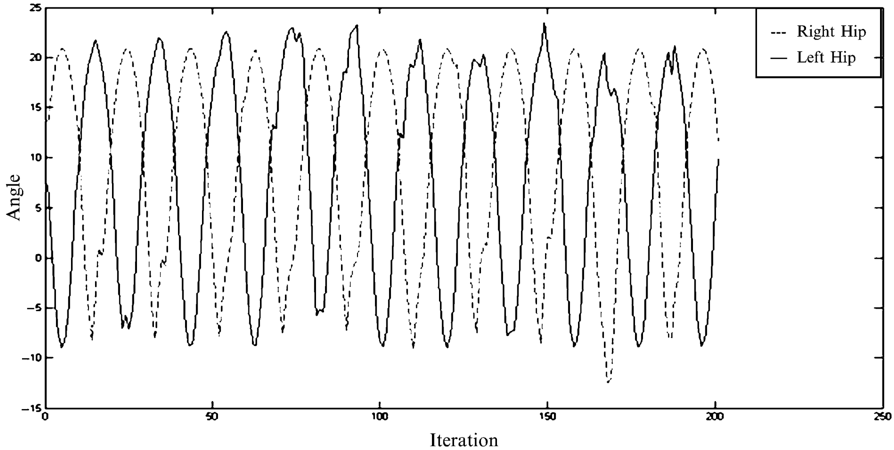

Fig. 2.5 Left and right hip angle trajectories

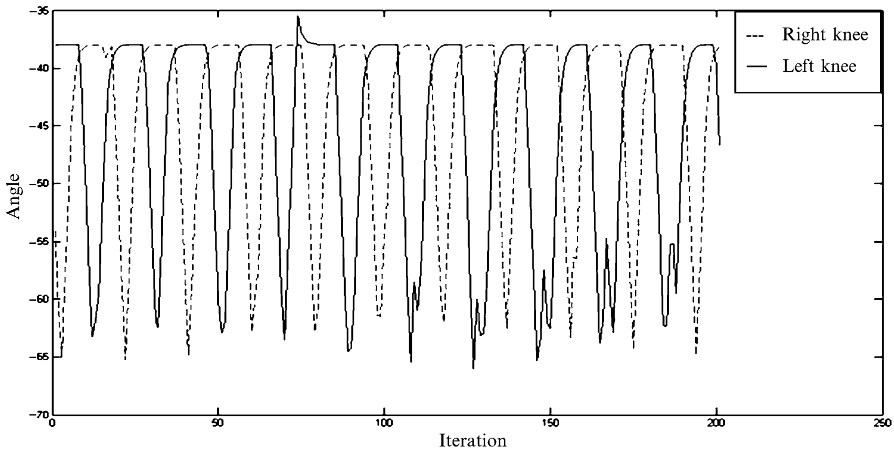

Fig. 2.6 Left and right knee angle trajectories

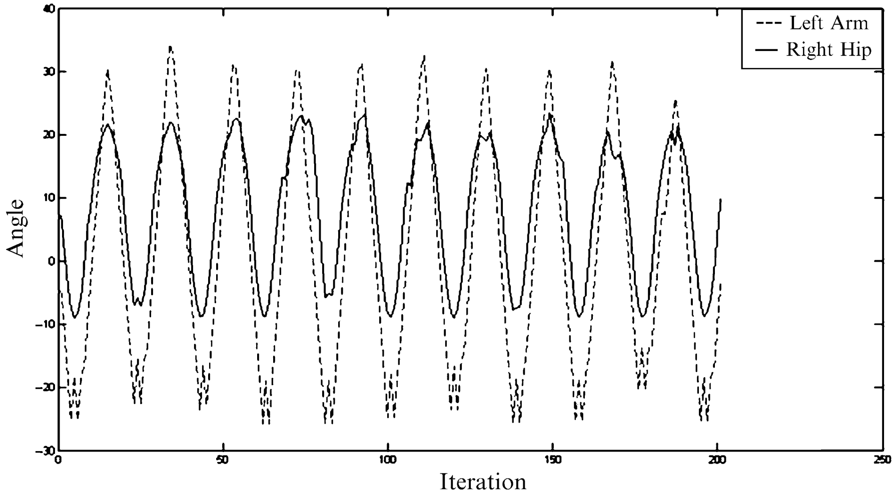

Fig. 2.7 Trajectories of *right* hip and *left* arm joint during walking

2.8 Concolution

In this paper, we are able to increase the speed and stability of the robots walking compare to previously presented model by adding arm swing during walking. The current implementation is capable of walking in a straight line on a planar surface without the use of proprioceptive input. In this study TFS with BA is implemented in a simulated NAO robot that can walk fast and stable. Using from BA as optimizer shows that The Bees Algorithm is a computationally fast multi-objective optimizer tool for complex engineering multi-objective optimization problems.

References

Beer RD, Chiel HJ, Quinn RD, Ritzmann RE (1998) Bio robotic approaches to the study of motor systems. Curr Opin Neurobiol 8(6):777–782

Kato I, Ohteru S, Kobayashi H, Shirai K, Uchiyama A (1974) Information-power machine with senses and limbs. First CISM-IFToMM symposium on theory and practice of robots and manipulators, Springer-Verlag

Takanishi A, Naito G, Ishida M, Kato I (1982) Realization of plane walking by the biped walking robot WL-10R. Robotic and Manipulator Systems 283–393

Hodgins JK, Raibert MH (1990) Biped gymnastics. Int J Robot Res 9(2):115

Zhang S, Zhu C, Sin JKO, Mok PKT (1999) A novel ultrathin elevated channel low-temperature poly-Si TFT. IEEE Electron Device Letts 20:569–571

Vukobratovic M, Borovacand B, Surdilovic D (2001) Zero-moment point proper interpretation and new applications. In: Proceedings of the 2nd IEEE-RAS international conference on humanoid robots, pp 237–244

Kajita S, Kanehiro F, Kaneko K, Yokoi K, Hirukawa H (2001) 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

McGeer T (1980) Passive dynamic walking. Int J Robot Res 9(2):62–82

Mochon S, McMahon TA (1980) Ballistic walking. J Biomech 13:49–57

Yang L, Chew CM, Poo AN (2006) Adjustable bipedal gait generation using genetic algorithm optimized fourier series formulation. In: Proceedings of the 2006 IEEE/RSJ international conference on intelligent robots and systems, pp 4435–4440

Yang L, Chew CM, Zielinska T, Poo AN (2007) A uniform biped gait generator with off-line optimization and on-line adjustable parameters. Robotica 25(5):549–565

Hinrichs RN (1990) Whole body movement, coordination of arms and legs. In: Winters JM, Woo SLY (eds) Walking and running. Springer-Verlag, New York, pp 694–705

Elftman H (1939) The function of the arms in walking. Hum Biol 11:529–535

Li Y, Wang W, Crompton RH, Gunther M (2001) Free vertical moments and transverse forces in human walking and their role in relation to arm-swing. J Exp Biol 204:47–58

Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S, Zaidi M (2006) The Bees Algorithm A Novel Tool for Complex Optimization Problems. In: Proceedings 2nd international virtual conference on intelligent production machines and systems(IPROMS), pp 454–45

Smith R (2007) Homepage of open dynamics engine project; http://www.ode.org.

Boedecker J (2005) Humanoid robot simulation and walking behavior development in the spark simulator framework, Artificial Intelligence Research University of Koblenz technical report

Kagami S, Mochimaru M, Ehara Y, Miyata N, Nishiwaki K, Kanade T, Inoue H (2003) Measurement and comparison of humanoid H7 walking with human being. Robot Auton Sys 48:177–187

Zehr EP, Duysens J (2004) Regulation of arm and leg movement during human locomotion. J Neuroscientist 10:347–361

Collins SH, Wisse M, Ruina A (2001) A 3-D passive dynamic walking robot with two legs and knees. Int J Robot Res 20:607–615

Yazdi E, Azizi V, Haghighat AT (2010) Evolution of biped locomotion using bess algorithm based on truncated fourier series. In: Proceedings of the world congress on engineering and computer science 2010(WCECS 2010), pp 378–382

Springer

http://www.springer.com/978-1-4614-0372-2

Intelligent Automation and Systems Engineering
Ao, S.-I.; Amouzegar, M.; Rieger, B.B. (Eds.)
2011, X, 430 p., Hardcover
ISBN: 978-1-4614-0372-2