Biped Walking Using Coronal and Sagittal Movements Based on Truncated Fourier Series

Nima Shafii[1,2], Luís Paulo Reis[1,2], and Nuno Lau[3,4]

[1] DEI/FEUP – Informatics Engineering Department, Faculty of Engineering of the University of Porto, Rua Dr. Roberto Frias s/n, 4200 465 Porto, Portugal
[2] LIACC – Artificial Intelligence and Computer Science Lab., Porto, Portugal
[3] UA – University of Aveiro, Campus Universitário de Santiago, 3810 193 Aveiro, Portugal
[4] IEETA – Institute of Electronics and Telematics Engineering of Aveiro, Portugal
nima.shafii@fe.up.pt, lpreis@fe.up.pt, nunolau@ua.pt

Abstract. Biped walking by using all joint movements and DOFs in both directions (sagittal plane and coronal plane) is one of the most complicated research topics in robotics. In this paper, angular trajectories of a stable biped walking for a humanoid robot are generated by a Truncated Fourier Series (TFS) approach. The movements of legs and arms in sagittal plane are implemented by an optimized gait generator and a new model is proposed that can also produce the movement of legs in coronal plane based on TFS. Particle Swarm Optimization (PSO) is used to find the best angular trajectories and optimize TFS. Experimental results show that the using joints movements in sagittal and coronal planes to compose the walking skill allowed the biped robot to walk faster than previous methods that only used the joints in sagittal plane.

Keywords: Bipedal Locomotion, Gait Generation, Particle Swarm Optimization.

1 Introduction

In competitive non-deterministic environments like RoboCup soccer humanoid robot leagues, flexible, fast and robust biped walking is one of the keys to win a match. Such movements must accomplish different requirements. For example, in case the target is far away, the humanoid robot must be capable of covering large distances in a short time and when the target is close it must be precise and robust in its approach. In this task, a real 3D bipedal locomotion by using all joints enabling a flexible robot movement has an important role. In the literature, several approaches for bipedal locomotion have been presented until now. They can be divided in two main groups, model based and model free. Model based methods need to obtain and process accurate information of the dynamics of the robot like its parts positions and velocities and their mass after that a controller is built for it. "Zero Moment Point" (ZMP) [1] and "Inverted Pendulum" [2] are two methods that belong to this approach. In many ZMP-based trajectory planning approaches, motion planning is presume and performed in the Cartesian space and has some motion assumptions[3]. Therefore it does not give much freedom for having human like motion.

J. Ruiz-del-Solar, E. Chown, and P.G. Ploeger (Eds.): RoboCup 2010, LNAI 6556, pp. 324–335, 2011.

Model–free approaches use sensory information for developing the humanoid motions. Passive Dynamic Walking (PDW) [4], Central Pattern Generator (CPG) [5] and Ballistic Walking [6] are the most known methods of model-free approach. Passive dynamic walking tries to mimic natural motion in a robot. Using PDW, the robot does not have any actuators' model and it walks just by utilizing the gravity force and can go down a shallow inclined ramp only with its own mechanical dynamics without using any actuation or external control. The Ballistic Walking is originated from the simple human walking model and is based on the observation that the muscles of the swing leg are activated only at the beginning and the end of the swing phase when real humans walk.

In the CPG-based approach, bipedal trajectory generation has been achieved with the use of special neural oscillators. Using non-linear equations to model the neural activities, the oscillators can generate rhythmic walking patterns. Researchers usually focus on complex mathematical models like Hopf [7] or Matsuoka [8] to model these neural activities and generate rhythmic walk patterns (Gait). The walk model developed using CPG is flexible enough to adapt to the environment. However, it is difficult to design the relation and feedback pathways of the neural oscillators, and tune the required parameters in order to achieve the desired walking characteristics manually [9].

In 2007 Truncated Fourier Series Formulation method (TFS) was used as a gait generator and CPG in bipedal locomotion [9]. TFS together with a ZMP stability indicator was used to prove that it could generate suitable angular trajectories for controlling bipedal locomotion but it was not implemented on a real robot [9]. In 2009, an optimized gait generator based on TFS was implemented in a simulated humanoid robot and TFS parameters were also reduced by 2 dimensions (down to 6 dimensions) [10]. Shafii extended the basic TFS enabling the generation of arm angular trajectories that provide smooth and robust walking, also a new method was used to refine signals for reducing the role of inertia to improve the speed and robustness of the robot [11].

In this paper the results of the two previous papers are used to produce walking movements on sagittal plane and also on coronal plane. The method was tested on a simulated NAO humanoid and the experiments were performed using Rcssserver3d [12], a generic three-dimensional simulator which is based on Spark and Open Dynamics Engine (ODE). The robot model has 22 DOF with a height of about 57cm, and a mass of 4.5kg.

The paper structure is as follows. First, optimized TFS gait generator is introduced to generate walking movement on the sagittal plane. Arm angular trajectories generator and the method for reducing the role of inertia are also explained. Then a new model of hip angular trajectory generator based on TFS is introduced which can produce the leg's movement on the coronal plane (Y direction). Particle Swarm Optimization (PSO) is used to optimize the produced signals, to overcome inherent noise of the simulator and Resampling algorithm is used to improve robustness in nondeterministic environments. At the end of the article, results of this approach are presented and the efficiency of the method on producing trajectories to walk a robot in the forward direction are shown.

2 Movements in Sagittal Plane

There are three DOFs in each leg move in sagittal plane: one in the hip, one in the ankle and one at the knee. In this work, similar to [13], swing foot in sagittal plane was kept parallel to the ground by using the ankle joint. This is done in order to avoid swing foot colliding with the ground. Therefore, ankle trajectory can be calculated by hip and knee trajectories and ankle DOF parameters are eliminated. Trunk sagittal and coronal plane motion is fairly repeatable [14] therefore Fourier series can be used.

2.1 An Optimized Gait Generator for Leg's Movement

In this model, legs joint angular trajectories in sagittal plane are divided in two parts; the upper portion and the lower portion. Let us define C_h as the offset of the hip trajectory and C_k as the offset of the knee trajectory. From t_1 to t_2 the left leg is considered as supporting leg and the variation of its knee angle is so little that it can be assumed fixed. This phase of walking is named knee lock phase. In addition, the two leg trajectories signals are repeated once for swinging the right leg and then for swinging the left one hence by producing the trajectory of one leg the other leg's trajectory can be calculated. The trajectories for both legs are identical in shape but are shifted in time relative to each other by half of the walking period. The TFS for generating each portion of hip and knee trajectories are formulated below.

$$\begin{aligned}
\theta_h^+ &= \sum_{i=1}^{n} A_i.\sin\left(iw_h t\right) + c_h, w_h = \frac{2\pi}{T_h} \\
\theta_h^- &= \sum_{i=1}^{n} B_i.\sin\left(iw_h t\right) + c_h, w_h = \frac{2\pi}{T_h} \\
\theta_k^+ &= \sum_{i=1}^{n} C_i.\sin\left(iw_k t\right) + c_k, w_k = w_h \\
\theta_k^- &= c_k \geq 0
\end{aligned} \tag{1}$$

In these equations, the plus (+) sign represents the upper portion of walking trajectory and the minus (-) shows the lower portion. $i=1$ and A_i, B_i, C_i are constant coefficients for generating signals. The h and k subscripts stand for hip and knee respectively. C_h , C_k are signal offsets and T_h is assumed as the period of hip trajectory. Considering the fact that all joints in walking motion have equal movement frequency and stride rates are statistically equal, the equation $w_k = w_h = \frac{2\pi}{T_h}$ can be concluded. Parameter t_1 is the start time of lock phase for knee joint and parameter t_2 represents the end time of lock phase and in this duration of time $\theta_k^- = c_k \geq 0$.

According to [10], by specifying the start and end time of the lock phase, two parameters of t_1, t_2 could be eliminated. Therefore the number of variable for optimization to produce legs movement in sagittal plane decreased to 6. This elimination has many advantages such as: reducing the search space of optimization problem and increasing the convergence speed of PSO algorithm.

2.2 Modeling of Arm Motion in Sagittal Plane

In sagittal plane, during human walking, the arms normally swing in opposite manner to legs, which helps to balance the angular momentum generated in the lower body [15]. Humans swing their arms close to 180º out of phase with their respective legs during walking [16]. Fig. 1 shows the trajectory of legs and arm swings and the relation between them in a stable straight walking [15]. It is shown that Trajectory of arms is similar to sinusoidal signal with same frequency of legs.

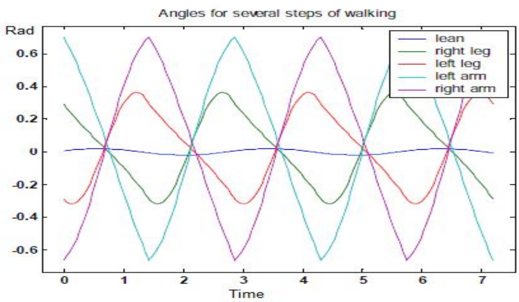

Fig. 1. Trajectories of legs and arms

The walking speed has a strong effect on arm swinging during gait. By increasing gait's speed, the arms may swing higher and faster to reduce the effects of longer, quicker steps by the legs [17]. It can be expected that the utilization of arm swing provides good performance to yaw moment stability, and recovery from stumbling. The effectiveness of this method is confirmed with an improvement of the accuracy of straight walking at different speeds. As has been shown, the trajectory of arms is a sinusoidal signal, therefore, to produce the angular trajectories of arms swinging, it is enough to obtain proper parameters for the equation 2.

$$f\ (t) = A\ \sin(\omega_{arm} t), \omega_{arm} = \omega_{arm} \tag{2}$$

In equation 2, A and w are assumed as the amplitude and frequency of the signal, respectively. In addition the shift phase of the two arm trajectories signals is half of the period of each signal, so by producing the trajectory of one arm the other arm's trajectory can be calculated. Since legs and arms have the same frequency, w_{arms} can be considered equal to w_{legs}. According to the fact that the angle of arms is zero at the start of walking, shift phase factor is assumed as 0.

According to equation 2 and the previous discussion, only the proper value of parameter A must be obtained. To find the best value for A we consider this parameter together with legs trajectory parameters in the optimization problem.

3 Increasing Speed at the Start Time of Walking in TFS

According to [10], when the robot starts walking the walking speed and amplitude should be increased in a controlled way from zero to their maximum values, hence improving walking stability and allowing higher final speeds. We implemented a model for the robot to walk from smaller gait with lower amplitude to bigger gait with higher speed and acceleration. In this model a linear equation is used to lead the robot to increase the amplitude of trajectory linearly form zero (stop state) to desired angular trajectories. T is assumed as a parameter to determine how much time is needed for this increment algorithm to reach these desired trajectories. All angular trajectories such as arms and legs will be multiplied by the product of the following equation.

$$\begin{aligned} K &= time / T \;, time < T \\ K &= 1, time >= T \end{aligned} \tag{3}$$

4 Movements in Coronal plane

The range of motion in the coronal and transverse plane is less than that seen in the sagittal plane [18] but it has an important role to keep the balance of walking and reach the highest speed of walking. The range of its motion depends on the speed of walking, and at higher speeds this range is smaller. Coronal plane movements are periodic motions [14]. Abduction and adduction are terms for movements of limbs relative to the coronal plane. To produce legs' motion in coronal plane and also considering keeping the balance of robot, we proposed a walking sequences and scenario (Fig. 2). It illustrates the walking sequence in a walking period. θ is assumed as the maximum of legs movement. Like in the previous section in coronal plane, feet were kept parallel to the ground in order to avoid collision and considering of the fact that just one of each hip's joint moves each time, the angle of ankle is equal to the hip's angle of the opposite leg.

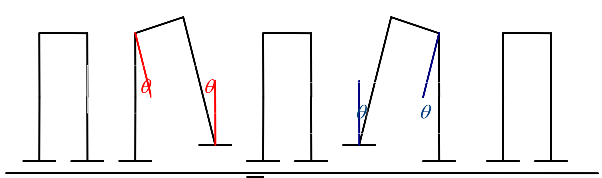

Fig. 2. Coronal plane view of proposed walking Sequence

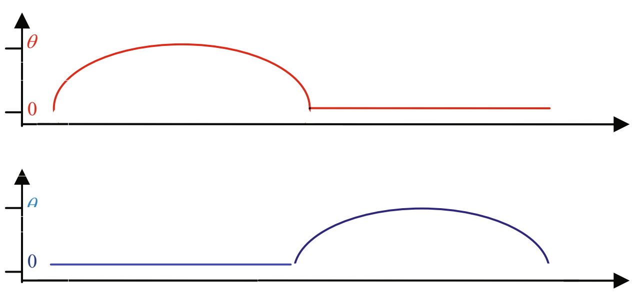

Fig. 3. Left leg and right leg hips angular trajectories

Considering the walking sequence, Fig. 3 can be assumed as the hip angular trajectory in one period of walking. It is a sinusoidal signal that has a lock phase at zero degree. Therefore in order to produce the proper angular trajectories to move the hips in coronal plane , proper parameters for the equation 4 must be obtained.

$$\begin{aligned} f(t) &= H \sin(wt), t < T_h / 2 \\ f(t) &= 0, t > T_h / 2 \end{aligned} \tag{4}$$

In equation 4, H and w are assumed as amplitude and frequency of signal respectively and T_h is assumed a period of hip. As mentioned before, ankle trajectories can be calculated from hip trajectories and as it is shown, left and right hip angular trajectories are the same but with a phase shift of -pi. The period of walking in sagittal plane and coronal plane is equal. Therefore T_h and w are eliminated in this method and for producing the proper abduction and adduction, the proper value of H parameter must be found.

According to the fact that movements in coronal plane decrease at higher speeds of walking, the movement of legs in coronal plane must be reduced when increasing the speed of walking. Therefore all angular trajectories generated by above model will be multiplied by the reverse product of equation 3.

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, 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. According to [19], for this kind of optimization problem, Particle Swarm Optimization can achieve better results. Therefore PSO seems to be an appropriate solution. In the following sections some brief details about the PSO algorithm are explained.

5.1 PSO Algorithm

The PSO algorithm contains the three following parts: generating primitive particle's positions and velocities, velocity update and position update [20].

Equations (5) and (6) are used to initialize particles in which Δt is 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.

$$X_0^i = X_{\min} + Rand(X_{\max} - X_{\min}) \tag{5}$$

$$V_0^i = \frac{X_{\min} + Rand(X_{\max} - X_{\min})}{\Delta t} = \frac{Position}{time} \tag{6}$$

Updating Velocities has an important role in finding the new search direction. current motion, particle own memory, and swarm influence, in a summation approach as shown in Equation below (7) influence on the calculation of new search direction. This equation consists of three weight factors, namely, inertia factor, w , self confidence factor, C_1, and swarm confidence factor, C_2. P_k^g ,P^i are the best position of each particle over time and the best global value in the current swarm respectively.

$$\underbrace{V_{k+1}^i}_{\substack{Velocity\ of\ Particle \\ i\ at\ time\ k+1}} = \overbrace{w}^{[0.4,1.4]} \underbrace{V_k^i}_{\substack{Current \\ Motion}} + \underbrace{\overbrace{C}^{[1,2]}{}_1\, Rand\, \frac{(P^i - X_k^i)}{\Delta t}}_{Particle\ Memory\ Influence} + \underbrace{\overbrace{C_2}^{[1.5,2]}\, Rand\, \frac{(P_k^g - X_k^i)}{\Delta t}}_{Swarm\ Influence} \tag{7}$$

Utilizing a nonlinear decreasing inertia weight as a dynamic inertia weight significantly improves its performance through the parameter study of inertia weight [21]. This nonlinear distribution of inertia weight is expressed as follow:

$$w = w_{init} * U^{-k} \tag{8}$$

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.

Finally the position of each particle in the swarm can be updated by using the current particle position and its own updated velocity vector as shown in the following Equation.

$$X_{K+1}^i = X_K^i + V_{K+1}^i \Delta t \tag{9}$$

The PSO algorithm may be described as follows:

```
Initialize Position (X_0) and Velocity of N particles
according to equation (4 and 5)
P^1=X_0
DO
     k=1
     FOR i = 1 to N particles
          IF f(X_i) < f(P^i) THEN P^i= X_i
          P_k^g = min (P)
          Calculate new velocity of the particle according
          to equation (6, 7 and 8)
          Calculate new position of particle according to
          equation (9).
     ENDFOR
     k=k+1
UNTIL a sufficient good criterion (usually a desirable
fitness or a maximum number of iterations) is met.
```

5.2 PSO Implementation

In PSO, the parameters of the problems are coded into a finite length of string as a particle. According to above sections, for producing movements in sagittal plane, TFS has 6 parameters to generate legs joints angular trajectories and 2 parameters are assumed to swing arms and to increase the speed of walking. There is also one parameter to produce proper legs' movement in coronal plane. Therefore there is a 9-dimension search space for the PSO to find the optimum solution.

Angular trajectories produced by each particle are applied to a simulated robot to optimize it walking capabilities. To use angular trajectory for walking, all individual robot's joints attempt to drive towards their target angles using simple proportional derivative (PD) controllers. To enable the robot with a fast walking skill a fitness function based on robot's straight movement in limited action time is considered. First the robot is initialized in x=y=0 (0, 0) and it walks for 15 seconds. After that, the fitness function is calculated whenever the robot falls or the time duration for walking is over. The fitness function formulation is simply expressed as the distance travelled by the robot along the x axis.

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 [23]. In Resampling, the individual set of parameters (particle) yi , the fitness F(yi) is measured m times and averaged yielding fitness. According to (10) the noise strength of $\overline{F}$ is reduced by a factor $\sqrt{m}$.

$$\overline{F(y_i)} = \frac{1}{m}\sum_{k=1}^{m} F(y_i), y(i) = const. \Rightarrow \overline{\sigma_e} = \sqrt{Var\left[\overline{F(y_i)}\right]} = \frac{\sigma_e}{\sqrt{m}} \qquad (10)$$

Since particles may not satisfy some constraints after updating position procedure, constraint handling is a vital step in PSO algorithm. There are many obvious constraints on parameters in this study (i.e time parameters in TFS must be positive). Therefore Pareto [24] 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 to be minimized:

$$fm\left(x\left(1\right)\right) \le fm\left(x\left(2\right)\right) for \forall m = 1, 2, \ldots, M \quad \text{and}$$

$$fm\left(x\left(1\right)\right) \prec fm\left(x\left(2\right)\right) \quad \text{for some } m.$$

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 Pi, Pgk when it satisfies objects and constraints. Therefore calculating fitness for particles that cannot satisfy constraint is not necessary.

We have 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 used for our experiments a particle swarm composed by 100 particles (N = 100),a maximum iteration of 10 and a Resampling factor m assumed as 3.

6 Results

To compare the presented method with Basic TFS method which uses the joints movement just in sagittal plane, both of them were tested using the same system and the same specification. We also optimized both methods by utilizing PSO with equal specifications and with the same fitness function.

Using basic TFS with 8 parameters and running PSO algorithm on a Pentium IV 3 GHz Core 2 Duo machine with 2 GB of physical memory, 3000 trials were performed in 4 hours. Finally the robot could walk 8.6m in 15s with average body speed of 0.57m/s and period of 0.41s for each step. Fig. 4 shows the average and best fitness values during these 10 generations.

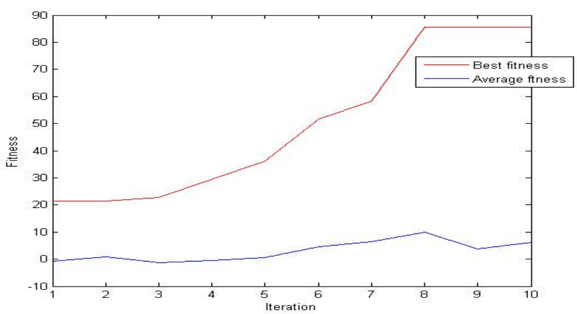

Fig. 4. PSO convergence for previous TFS

Using the new approach presented in this paper, after 3300 trails and 5 hours from starting PSO in a machine with the same specifications, the robot could walk 11.5 m in 15 s. Average and best fitness are shown in fig. 5.

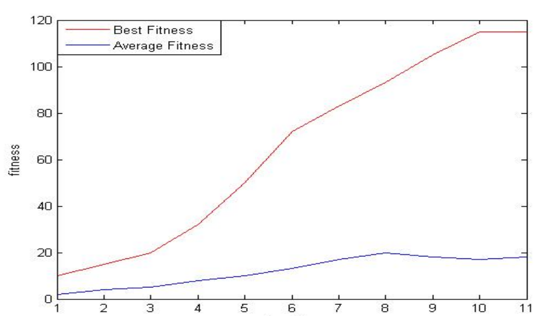

Fig. 5. PSO convergence for new approach and TFS model

Robot could walk with average body speed of 0.77 m/s by using the TFS with arm swing and increasing speed technique. The learned trajectories of left hip and knee after robot started to walk are shown in Fig. 6.A. It is determined that robot increased its speed in 0.20 m/s. The learned trajectory of left arm is also shown in Fig. 6.B. and finally, the learned trajectory of left hip in coronal plane for abduction and adduction movements is also shown in Fig. 6.C.

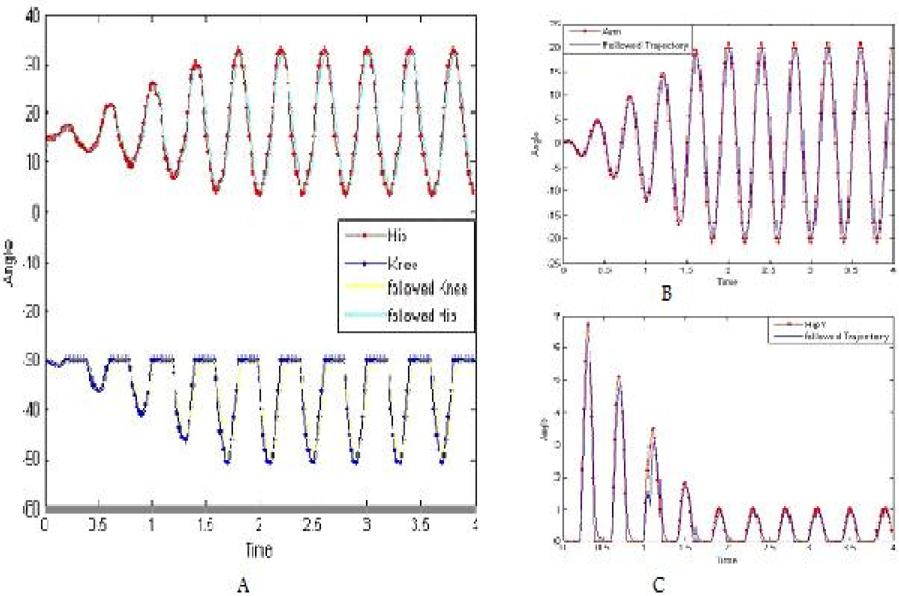

Fig. 6. A) Left Hip and Knee trajectories in the sagittal plane; **B)** Left Arm trajectory; **C)** Left hip trajectory in coronal plane (hipY)

After the learning procedure, the robot could walk with the average speed of 0.77 m/s. The best results of walking behavior for the teams that participated in RoboCup 2008 were chosen for the comparison [25]. Table 1, presents the comparison of the best results of RoboCup teams, compared with the proposed approach. Analyzing the table it is clear that forward walking achieved by our approach, may outperform the same skill of all teams analyzed except SEU.

Table 1. Comparsion the average speed for forward walking in different teams (m/s)

	Proposed approach	FCPortugal	SEU	Wright Eagle	Bats
Forward Walking	0.77	0.51	1.20	0.67	0.43

7 Conclusions

This paper presented a model with 9 parameters for producing all walking angular trajectories that uses all joints' movements in coronal and sagittal plane. An optimized Truncated Fourier Series is used to produce leg movements in sagittal plane and a model for swinging arms. A new model was also used to generate legs walking movement in coronal plane. We are able to increase the speed and stability of the robot's walking when compared to previously TFS model by using this model and the method mentioned in sec. 4 which was used both in sagittal and coronal motion.

According to the fact that this approach is model free and based on robot learning, it is capable of being used on all kinds of humanoid robots with different specifications. In future works we would like to expand this model to produced turn motion and improve the approach so that the robot can walk in any direction.

Future work will be concerned with high level control of the walking behavior and navigation with obstacle avoidance in the context of FC Portugal team [25, 26, 27] for participation in RoboCup 2010 simulation 3D league.

Acknowledgements

The first author is supported by FCT under grant SFRH/BD/66597/2009. This work has been partially funded by FCT Project ACORD - Adaptative Coordination of Robotic Teams (PTDC/EIA/70695/2006).

References

1. Vukobratovic, M., Borovac, B., Surla, D., Stotic, D.: Biped Locomotion Dynamics Stability Control and Application. Springer, London (1990)
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: Proc. of the 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 239–246 (2001)
3. Chevallereau, C., Aoustin, Y.: Optimal reference trajectories for walking and running of a biped robot. Robotica 19, 557–569 (2001)
4. McGeer, T.: Passive dynamic walking. International Journal of Robotics Research 9(2), 62–82 (1990)
5. Pinto, C., Golubitsky, M.: Central Pattern Generator for Bipedal locomption. J. Math. Biol. 53, 474–489 (2006)
6. Ogino, M., Hosoda, K., Asada, M.: Learning Energy Effcient Walking with Ballistic walking. Adaptive Motion of Animals and Machines, pp.155–164 (2006)
7. 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)
8. Matsuoka, K.: Sustained oscillations generated by mutually inhibiting neurons with adaptation. Biol. Cybern. 52, 367–376 (1985)
9. 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)

10. Shafii, N., Javadi, M.H., Kimiaghalam, B.: A Truncated Fourier Series with Genetic Algorithm for the control of Biped Locomotion. In: Proceeding of the 2009 IEEE/ASME International Conference on advanced intelligent Mechatronics, pp. 1781–1785 (2009)
11. Shafii, N., Khorsandian, A., Abdolmaleki, A., Jozi, B.: An Optimized Gait Generator Based on Fourier Series Towards Fast and Robust Biped Locomotion Involving Arms Swing. In: Proceeding of the 2009 IEEE International Conference on Automation and Logistics, pp. 2018–2023 (2009)
12. Boedecker, J.: Humanoid Robot Simulation and Walking Behaviour Development in the Spark Simulator Framework. Technical report, Artificial Intelligence Research University of Koblenz (2005)
13. 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. 48, 177–187 (2003)
14. Konz, R.J., Fatone, S., Stine, R.L., Ganju, A., Gard, S.A., Ondra, S.L.: A Kinematic Model to Assess Spinal Motion During Walking, J. J. Biomechanics 31, E898–E906 (2006)
15. Elftman, H.: The function of the arms in walking. J. Hum. Biol. 11, 529–535 (1939)
16. Collins, S.H., Wisse, M., Ruina, A.: A 3-D passive dynamic walking robot with two legs and knees. Int. J. Robot. Res. 20, 607–615 (2001)
17. Murray, M.P., Sepic, S.B., Barnard, E.J.: Patterns of sagittal rotation of the upper limbs in walking. Physical Therapy 47, 272–284 (1967)
18. Thewlis, D., Richards, J., Hobbs, S.: The Appropriateness Of Methods Used To Calculate Joint Kinematics. Journal of Biomechanics 41, S320–S320 (2008)
19. Shafii, N., Aslani, S., Nezami, O.M., Shiry, S.: Evolution of Biped Walking Using Truncated Fourier Series and Particle Swarm Optimization. In: Baltes, J., Lagoudakis, M.G., Naruse, T., Ghidary, S.S. (eds.) RoboCup 2009. LNCS, vol. 5949, pp. 344–354. Springer, Heidelberg (2010)
20. Shi, Y., Eberhart, R.C.: Parameter selection in particle swarm optimization. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 591–600. Springer, Heidelberg (1998)
21. Jiao, B., Lian, Z., Gu, X.: A dynamic inertia weight particle swarm optimization algorithm. J Chaos 37, 698–705 (2008)
22. Beyer, H.G.: Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice. Comp. Methods Appl. Mech. Engrg. 186, 239–267 (2000)
23. Coello, C.A., Pulido, G.T., Lechuga, M.S.: Handling Multiple Objectives With Particle Swarm Optimization. IEEE Trans. Evolutionary Computation 8, 256–279 (2004)
24. Salman, A., Ahmad, I., Al-Madani, S.: Particle Swarm Optimization For Task Assignment Problem. Microprocessorsand Microsystems 26, 363–371 (2002)
25. Picado, H., Gestal, M., Lau, N., Reis, L.P., Tomé, A.M.: Automatic Generation of Biped Walk Behavior Using Genetic Algorithms. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) IWANN 2009. LNCS, vol. 5517, pp. 805–812. Springer, Heidelberg (2009)
26. Reis, L.P., Lau, N.: FC Portugal Team Description: RoboCup 2000 Simulation League Champion. In: Stone, P., Balch, T., Kraetzschmar, G. (eds.) RoboCup 2000. LNCS (LNAI), vol. 2019, pp. 29–40. Springer, Heidelberg (2001)
27. Lau, N., Reis, L.P.: FC Portugal - High-level Coordination Methodologies in Soccer Robotics, Robotic Soccer, Book edited by Pedro Lima, Itech Education and Publishing, Vienna, Austria, pp. 167–192 (2007)