Polytechnic-Parsian RoboCup 2010 Team Description Paper

Soheil Jahangiri, Mohammad Taghi Khalifeh, Karim Moradi, Hojjat Nikan, Abolfazl Nadi, Zabihollah Sarvari, and Hedayat Vatankhah

Electrical Engineering Department,
Amirkabir University of Technology, Tehran, Iran
parsian.soccer3d@gmail.com

Abstract. In this paper, motion control of the soccer agents of Parsian team is considered. We developed a system to represent action trajectories with Truncated Fourier Series (TFS). On the other side, we are working on a CPG based action controller specially for the walking of the robot. For other actions like kicking, a reinforcement learning methodology is being developed. Also, we are developing a facilitating software called Simspark Toolkit, based on QT4 which provides some useful functionalities for researchers.

1 Introduction

Biped robots have better mobility than conventional wheeled robots, especially for moving on rough and uneven terrain. Study of these robots and their stability has been the main focus of too many researchers in the last decades.

Robocup competitions specially humanoid, standard and 3d soccer simulation leagues have provided researchers with a great platform to test the stability and advantages of their algorithms on humanoid locomotion and motion control. Polytechnic Parsian 3d soccer simulation team presents our research results on locomotion and motion control of humanoid robots specially on Hoap and Nao.

Because of interdisciplinary and challenging characteristics of the humanoids, many controllers have been developed throughout three past decades.

Currently our agents use a walking algorithm based on Truncated Fourier Series (TFS) which is introduced in the next section. But also we are working on other approaches to generate more stable and controllable actions.

The approach which is used to generate actions should be able to produce stable and perturbation resistant actions. To achieve this goal, we are trying to employ Central Pattern Generators (CPGs) as a technique to mimic human internal mechanisms of online trajectory production.

Tuning parameters of such a system is hard and mostly experimental. We solve this challenging problem by using Comprehensive Learning Particle Swarm Optimization (CLPSO).

We hadn't reached a complete stable walk via this approach yet but the current results provide us with promising feedbacks.

For the kick action that highly needs adaptation to the environment changes, like changes in the ball position, we are implementing a Reinforcement Learning (RL) methodology to learn the path between different robot states. A state is said to be the set of all agent joint angles.

Along this research scheme, we are working on a toolkit called Simspark Toolkit, to ease the process of research in a standard interface. Other researchers on the field, specially those who will use Simspark simulation environment can easily use and extend the toolkit as they need.

In the next section, we will present our work on robot's actions. In section 3, Simspark Toolkit is introduced. Finally, we will explain some ideas to work on, in a near future.

2 Generating Actions

2.1 From trajectories to Fourier series

In the field of robot motion generation, using adaptive algorithms has been interesting for many researchers. But there are some critiques about these algorithms, such as lengthy learning times, instability, low speed of robot motion, and inflexiblity of the generated trajectories. We have overcome these challenges by employing Fourier series.

Considering human movements and robots that mimic humans, we can easily find out the periodic nature of such movements.

As shown in previous works, it is possible to model the angular movement of each joint using a periodic function of time. Given such a function $f(t)$, its Fourier series can be written as:

$$f(t) = \frac{1}{2}a_0 + \sum_{i=0}^{\infty} a_i sin\left(\frac{2\pi i}{T}t\right) + \sum_{i=0}^{\infty} b_i sin\left(\frac{2\pi i}{T}t\right) \quad (1)$$

In this equation, T is the function period and a_i and b_i are constant coefficients. To make the computation affordable, we will use Truncated Fourier Series by limiting the number of terms to a small number. The final result will be:

$$f(t) = \sum_{i=0}^{n} a_i sin\left(i\omega t\right) + b_i cos\left(i\omega t\right) + c_f \quad (2)$$

In which ω is the fundumental frequency which is calculated as $\omega = \frac{2\pi}{T}$ using the desired period of the gait; a_i, b_i and c_f are constants to be determined. Parameter n is the number of terms in the Fourier series and provides a tradeoff between the computational load and the required approximation.

The use of TFS allows for a reduced series with fewer parameters and constraints for the same accuracy, which significantly reduced the subsequent computational load when looking for optimal parameters.

In the learning phase, we can use a fixed movement frequency, but after finding suitable values for the parameters using the Comprehensive Learning

Particle Swarm Optimization method, we can change this frequency to control the robot behavior.

For a robot model with 20 degrees of freedom and 7 coefficients for each Truncated Fourier Series, the dimension of particles will be high. To simplify, we can assume the movement of legs are the same with 180 degrees phase difference. Also, some of the joints are omitted in the learning process and just 3 joints of one leg are considered. Hands of the robot are waved with a sinusoidal signal.

By considering more restrictions in Fourier coefficients and studying coefficient intervals, the results of the learning process are very interesting. The robot's movement converges to a human like stable walk.

We also implemented other evolutionary methods such as GA and PSO to find TFS parameters and compared their results with the results of CLPSO. We submitted this comparison as a paper to Robocup Singapore 2010 symposium.

2.2 CPG

Trying to implement human-like locomotion capabilities for robots, researchers have always been turn between a technological approach and a biological approach. The former relies on concepts and techniques known from robotics. This approach can either be motivated by the attempt to find a better solution or lack of the comprehension of nature.

The biological approach on the other hand starts by thoroughly analyzing the motion of animals or humans. These mechanisms are then adapted and translated into algorithms understandable to machines.

Biological investigations suggest that locomotion in vertebrates, including human beings, is mainly generated by a rhythm generator which is called Central Pattern Generator (CPG) within the spinal cord. The CPG is defined as a neural circuit that can produce self-sustained patterns of behavior (Fig. 1).

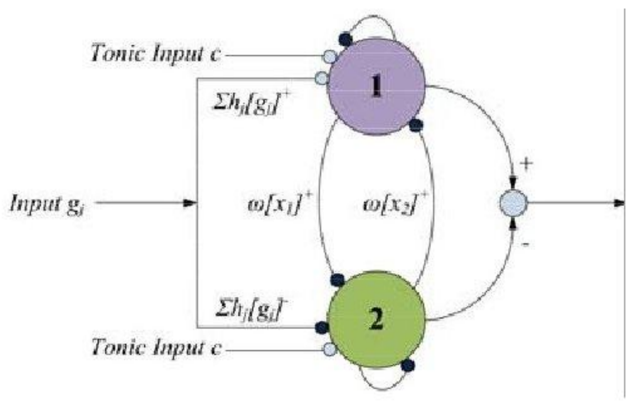

Fig. 1. An individual neural oscillator

The implementation of the artificial CPG is usually achieved by explicit differential equations.

$$T_{ri}\frac{\mathrm{d}\,x_i}{\mathrm{d}\,t} + x_i = -a_{ij} + s_i - b_i f_i \tag{3}$$

$$y_i = g(x_i) \tag{4}$$

$$T_{ai}\frac{\mathrm{d}\,f_i}{\mathrm{d}\,t} + f_i = y_i^q \tag{5}$$

The use of CPGs offer certain advantages, like robust control of the joints and easy adjustment of walking speed and step length, also no dynamics model of the robot or the environment is needed. But also it has several drawbacks, such as a high number of parameters that have to be specified; and it is rather difficult to determine the appropriate parameter settings for the oscillator network in the CPG in order to generate a suitable pattern for control of robot's walking. In other words there is no methodology to tune the value of the parameters of CPG. For modeling the central pattern generators, Matsuokas neural oscillator model has been used. This model consists of two mutually inhibiting neurons called flexor and extensor neurons. We implemented the Matsuoka model by Simulink in Matlab software as shown in Fig.2.

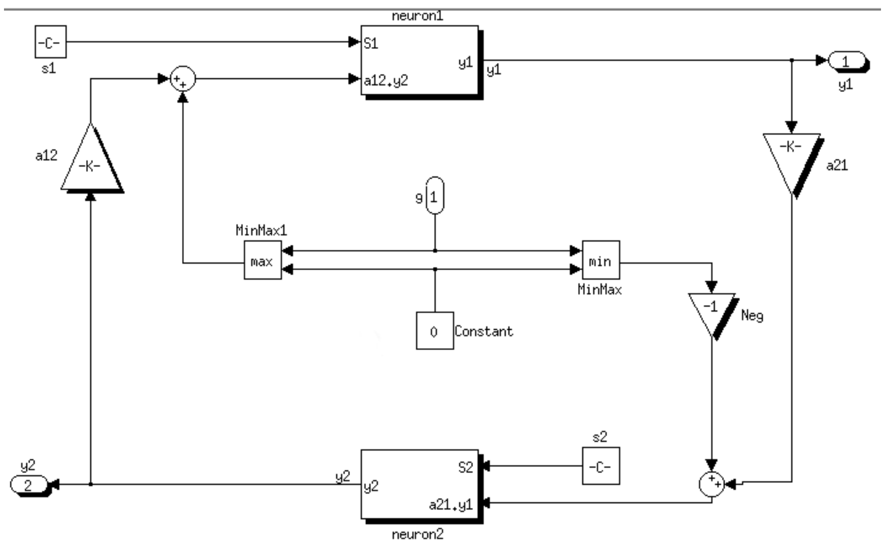

Fig. 2. Coupled neuron model implemented in Matlab

For tuning the parameters of CPG, evolutionary computation methods such as Genetic Algorithm (GA), multi-objective Genetic Algorithm and so on are often used to optimize the parameters. Convergence is an important factor in the selection of a suitable method to prevent a robot from numerous iterations of the algorithm. Particle Swarm Optimization (PSO) is a population based optimization method proposed by Kennedy and Eberhart in 1995 and is inspired by the social behavior of natural populations such as birds or fish. The main advantages of PSO is that it is simple to understand, easy to implement and quick in convergence compared to other global optimization algorithms such as Genetic Algorithms (GA) or Simulated Annealing (SA). PSO has been successfully applied in continuous nonlinear function optimization. In our first experience, we tried the canonical PSO, but this method could easily stick to a local minimum. We can get rid of the local minimum by using an enhanced method like Comprehensive Learning Particle Swarm Optimization (CLPSO) which we are still working on it.

2.3 The Kick Action

Generating adaptive and online trajectories for special actions of a robot is an important and challenging issue in humanoid robots. We apply a novel method for online generation of an adaptive trajectory for the kick action of a humanoid robot using reinforcement learning. We obtained important joints for a kick action by visual inspection of human kick and statistical analysis of kick actions of humanoid robot models in a simulated 3D soccer environment. We reduced the search space of the applied reinforcement learning algorithm by imposing some simplifications and restrictions. Finally we are employing a neural network to estimate the value function of the reinforcement learning algorithm.

3 Simspark Toolkit

Simspark Toolkit is an application software developed to extend the capabilities and to ease the use of Simspark. It has features that let the user to gain better results in less time. An interesting point about this software is its implementation and development under QT4 that permits interested developers to implement their extra features and changes easily to extend the software.

The Simspark toolkit has the abililty of sending direct commands to the server. This feature enables the user to manualy control the robot and change the angle of each joint seprately (Fig. 3).

The Toolkit has an instance of a 2D monitor, a 3D monitor and their corresponding configuration can be changed easily. In the 2D monitor, one can drag and move the agent to set its position in the 3D environment. Also ball position and agents' formation can be easily changed via the 2D monitor.

Configuration of the settings can be done in manual or automatic mode. In the automatic mode, the server default values will be used; otherwise, the user can set configuration parameters manually.

Also, in 2D monitor, lines and polygons can be added to show the selected agent's frustum of view and corresponding distances.

Another interesting feature of the toolkit is its capability to show the requested data as plots (Fig. 4). Also, analytical information of such data will be

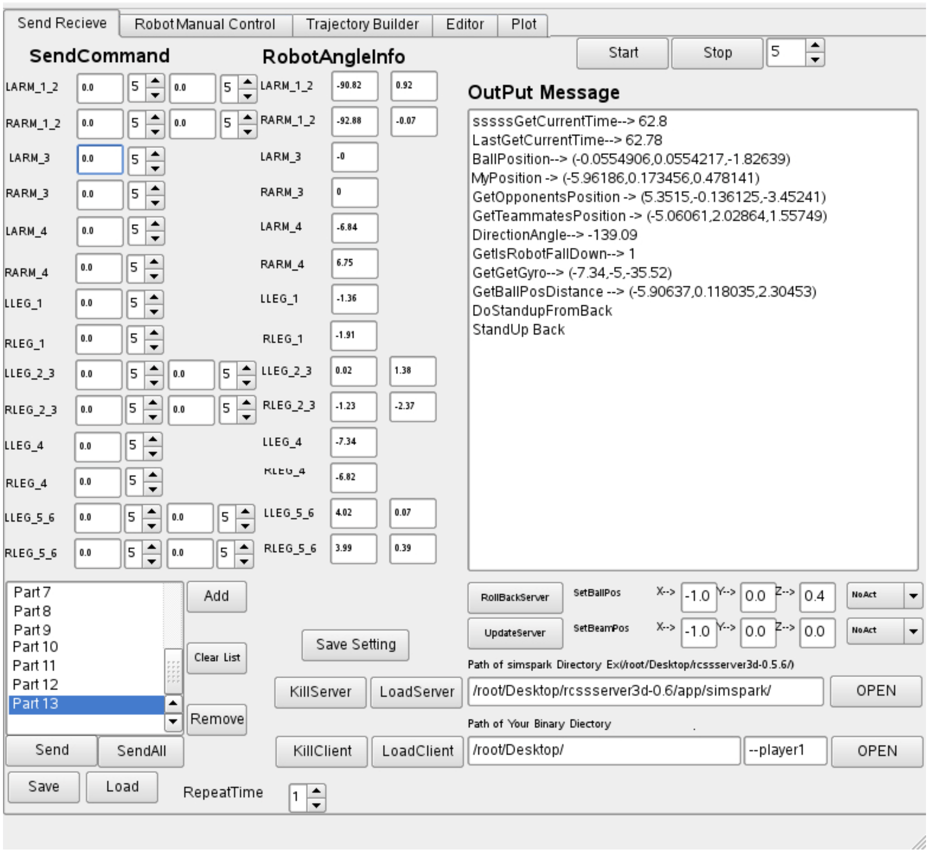

Fig. 3. Robot manual control

shown. The data can be read from a file or can be read online from local or remote computer via TCP/IP connection.

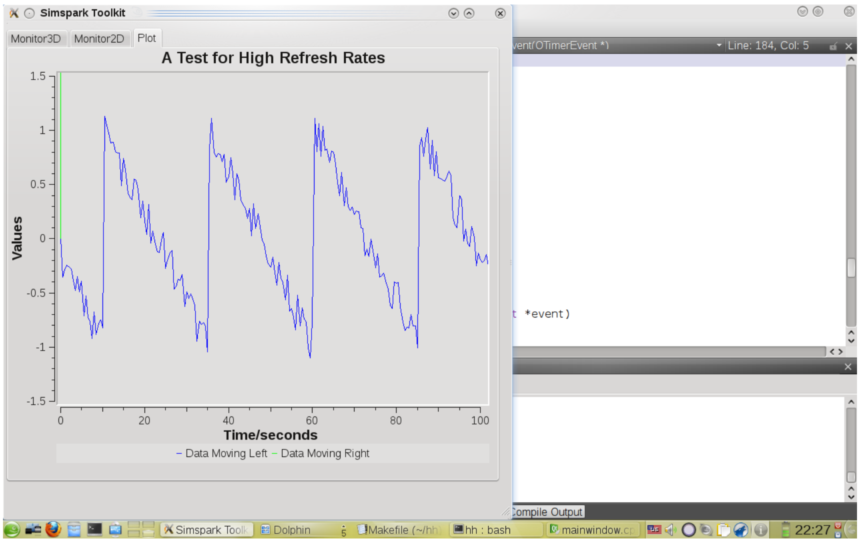

Fig. 4. Plotting in Simspark Toolkit

4 Future Work

We are planning to use more joints of the robot for the CPG model of walking, to explore all of the search space of walking conditions. For this purpose, we need faster training algorithms to search the problem space, which is the most time consuming part of the system is Matlab. We want to implement the whole CPG network in C++, to increase the speed of learning in larger search spaces.

We are developing a decision making engine which is used to describe high level actions using low level actions as building block of the high level ones. This engine supports a hierarchical structure of actions and can break a high level action to a set of basic actions to achieve its goal. For example, going behind the ball will be substituted by low level actions like turning, walking, etc. High level actions can be defined using a GUI interface. For more flexibility, a scripting facility will be implemented.

Also, for Simspark Toolkit we want to add joystick control support for easily tuning of the robot joints.

References

1. H.Inada and K.Ishii. A Bipedal walk using a Central Pattern Generator. In Brain-Inspired IT I, pages 185 188. Elsevier, 2004.

2. A. J. Ijspeert and J-M. Cabelguen. Gait transition from swimming to walking: investigation of salamander locomotion control using nonlinear oscillators. In Proceeding of Adaptative Motion in Animals and Machines, 2003.
3. K. Matsuoka. Mechanisms of frequency and pattern control in the neural rhythm generators. Biological Cybernetics, 56:345 353, 1987.
4. K. Matsuoka, Sustained oscillations generated by mutually inhibiting neurons with adaption, Biol. Cybern. 52(6), 367376 (1985).
5. Seiichi Miyakoshi, Gentaro Taga, Yasuo Kuniyoshi, and Akihiko Nagakubo. Three Dimensional Bipedal Stepping Motion using Neural Oscillators Towards Humanoid Motion in the Real World. In Proceedings of 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1998.
6. G. Taga, Y. Yamaguchi, and H. Shimizu. Selforganized control of bipedal locomotion by neural oscillators in unpredictable environment. Biological Cybernetics, 65(147159), 1991.
7. Nima Shafii, Siavash Aslani, Omid Mohamad Nezami, Saeed Shiry. Evolution of Biped Walking Using Truncated Fourier Series and Particle Swarm Optimization
8. 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).
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. James Kennedy' and Russell Eberhart. Particle Swarm Optimization. PurdueSchool of Engineering and Technology Indianapolis, IN 46202-5160
11. 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)
12. Jun Morimoto, Gordon Cheng, Christopher G. Atkeson, and Garth Zeglin. A Simple Reinforcement Learning Algorithm For Biped Walking. Proceedings of the 2004 IEEE International Conference on Robotics & Automation New Orleans, LA April 2004
13. Kaelbling, Leslie P.; Michael L. Littman; Andrew W. Moore (1996). Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research 4: 237285.
14. Sutton, Richard S.; Andrew G. Barto (1998). Reinforcement Learning: An Introduction. MIT Press. ISBN 0-262-19398-1.