Bipedal Walking Control Using Genetic Algorithm

Mehdi Azhdari, Amir Massah Bavani, Hossein Heidari,

Mehdi Zade Minaee, Moslem Farahmandian, Azam Khosravi,

Zahra Hamneshin, Masoome Kiani

Azad university, Mamasani branch

MahdiAzhdary@yahoo.com

Abstract. Bipedal walking control is a very complex and interesting dynamic problem for researchers. In this paper we try to probe a method based on Genetic programming to solve it. Genetic algorithm is a searching method based on random genetic evolution in nature [John H. Holland, 1986]. There are some different methods to bipedal walking control problem. What is presented in this paper is an evolution programming approach used to control bipedal walking. This approach is tested in simulation bed.

1. Introduction

Humanoid robots try to imitate human abilities. Researchers focus on the fact that humanoid robots should imitates human movement exactly. To some extent this is because of humanoid skeletons which suitable to do their human-like functions. To do their functions compared with the other types of robots, bipedal humanoid robots are more flexible on spongy and uneven surface [Raibert, 1986; Huang et al., 1999, 2001; Park & Cho, 2000].

Bipedal movement is complex dynamic system [Vukobratovic et al., 1990], repeatability of motion [Vukobratovic et al., 1990; Faconti, 2003], Changeability of kinematical Structure [Kajita et al., 2002; Shih & Gruver, 1992; Vukobratovic & Borovac, 2004; Denk & Schmidt, 2001] and powerless Degree of Freedom (DOF) [Vukobratovic et al., 1990].

Four major categories of different methods proposed to bipedal walking control in recent years. 1-off line trajectory control method, 2-real time trajectory control method, 3-passive dynamic control, 4-neural oscillatory control methods [2].

Genetic algorithm belongs off line trajectory control method category .Almost all methods of controlling bipedal motions except genetic algorithms which need few data require full and precise dynamic information [2]. In next section we introduce our design. Implementation is presented in section 4.

2. Design

In this section we present our idea to adapt Genetic Programming to use in bipedal walking control problem.

2.1. Why generic algorithm

We know that the generic algorithm is a parallel searching algorithm. And it has more capability than the other method; it can be used as machine learning.
With this algorithm we can provide a parallel learning method sheared among some robots in a simulation test bed; In other words, we share a large population between some robots and run those together. This method need less dynamic information to bipedal walking control. In the end another preference of this method adaptability with another different environment is only possible by changing the primary population.

2.2. Server

Developing physical humanoid robots is more complex and costly. Since Generic Algorithm is a repetitive and time consuming method, we need computer a simulation approach to cheek this, also, in a simulation environment. He can implement parallel learning easily. Our simulated humanoid robot is a legged spark agent [3, n] in Robocup 3D soccer simulation that has only the lower coat.
This robot has a five-freedom degree in his right and left legs. Both hip and ankle joints are universal joint with one freedom degree[3] this server use a powerful open dynamic engine(ode) library to simulate real physics rules. Figure 1 show this robot Skelton and his environment.

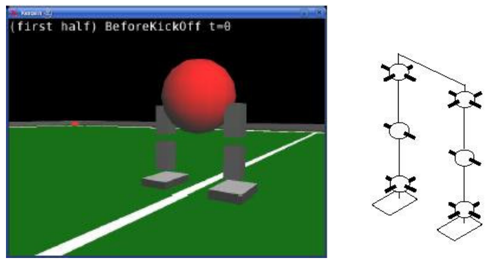

Fig. 1. Agent skeleton

2.3. Design with genetic algorithm

Before we describe our method, we must describe based joint. Based joint is one joint of five joints above mentioned and this joint starts changing, before the other joints. In this design the base joint is considered as the base for the motion of all the other joints. That is, all the other joints start motion in a t-period of time following the base joints motion (t >= 0).

3. Implementation

We pose two chromosomes for each leg. Each chromosome has some genes per joint. Genes indicate sequential time section that curve has: positive, negative or zero gradient. We pose that is one gene in each chromosome for each joint motions change. Figure 2 shows motion diagram and gene design.

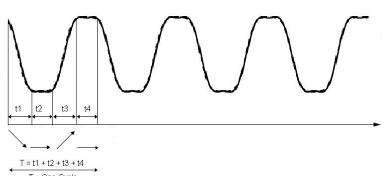

Fig. 2. motion diagram and gene design

Correct designing is extremely important parameter in generic programming. coupled operation and fitness function depend on chromosome, too. We use a function-based on an unsupervised privilege according to horizon direct distance passed similar to

$$Fitness - value = X * L \quad (1)$$

L is distance passed in horizon direct

X is the fixed value

Some important parameters in genetic programming is mentioned as follow:

1. Population size: 10* 300=3000 (number of agent number of population size for each agent)
2. Number of run 1500, run for each agent parallelism (10*1500)
3. Crossover ratio: 0.09
4. Mutation ratio: 0.05
5. Initialize method: random
6. Selection method: tournament selection

At start we create random population and divide them between all agents. Each agent has to evaluate his chromosome. The Following algorithm is used for evaluate chromosome.

1. Selecting one chromosome
2. Agent requires the server to put it on the ground in a stance position.
3. Looks at one fixed object in the environment and calculates the distance.
4. Applying crossover function in his joints according to the based joints
5. Agent calculates the distance to fixed object in section 3 when agent height is lower than his shank.
6. Calculates the difference between section (3,5) distance and remainder chromosome fitness.
7. If a chromosome exists, it repeats all the stages.

When all the agents evaluate their chromosome, all the population combine with agent and result is a new generation population.

Based on all the robot that we have, we can divide the population and then the robots repeat the stages 1-7. This work goes ahead until we get to a chromosome which can case our robot to walk two steps. Figure 3 show agent walking.

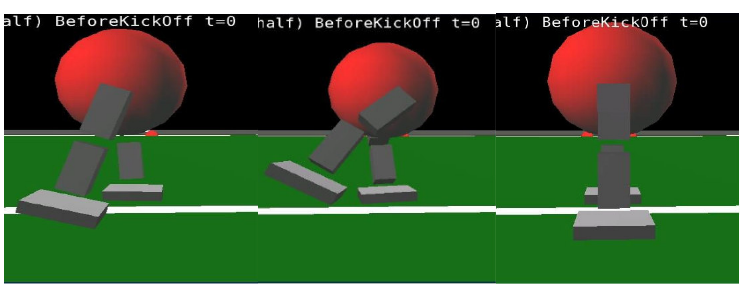

Fig. 3. Agent walking

Future work:

This paper introduce our idea to use generic algorithm for bipedal walking in old soccer 3D simulator version (0.5.1) simulated humanoid robot. We try to optimize and use this paper result on new version of simulated humanoid robot version (0.5.3).

Reference:

1- Bachar Y., "Developing Controllers for Biped Humanoid Locomotion", M.Sc. thesis, School of Information, University of Edinburg, 2004.

2- Louis S. J., "Genetic Algorithms as a Computational Tool for Design", Ph.D thesis, August 2003.

3- Boedecker Joschka, "Humanoid Robot Simulation and Walking Behavior Development in the Spark Simulator Framework", Diploma Thesis Proposal, Artificial Intelligence Research Koblenz, University of Koblenze-Landau, April 2005.

4- Russell Smith. "Open Dynamics Engine (ODE) User Guide". ODE webpage http://www.ode.org/, ODE Library Version 0.5 at URL http://opende.sourceforge.net/, 2005.

5- RoboCup Soccer Server 3D Maintenance Group(2005). *The RoboCup 3D Soccer Simulator.*

6- RoboCup Soccer Server 3D Mailling List, with Mail address sserver-three-d@lists.sourceforge.net, 2005.

7- RoboCup Soccer Server 3D Package version 0.5.1 at URL http://ovh.dl.sourceforge.net/sourceforge/sserver/, 2006.

8- Rollmann M., Spark-a Generic Simulator, Thesis submitted for Diploma, AI Research Group, University Koblenz-Landau, June 2004.

9- Righetti L., Ijspeert A. J., Programmable Central Pattern Generators: an application to biped locomotion control, Biologically Inspired Robotics Group, School of Computer and Communication Sciences, Ecole Polytechnique F´ed´erale de Lausanne (EPFL) – Switzerland, 2005.