Nexus 3D 2008 Team Description

Amin Milani Fard, Amir Farzad, Amin Zamiri, M. Hossein Ansari, Ehsan Saboori,
Mahdi Aledaghi, Mahmoud Naghibzadeh

Software Simulation and Modeling Laboratory, Department of Computer Engineering
Ferdowsi University, Mashhad, Iran
nexus@um.ac.ir
http://nexus.um.ac.ir

Abstract. This paper presents an overview of our 3D Soccer Simulation Team. First we describe the main strategies and methods used in Nexus 2005 3D Soccer Simulation team. Then we address important features and improvements that are going to be in Nexus 2006. The main development we made was using a two-phase selection mechanism to determine the best action among all possible ones carried out by the ball controller agent for a given situation.

1 Introduction

Nexus 3D Soccer Simulation team developed by a group of M.S and B.S students of Ferdowsi University of Mashhad. The Nexus team successfully participated in Robocup 2005. Previously our 2D team participated in Robocup 2003 in Padova. Nexus 2006 Soccer Simulation 3D team is mainly based on the architecture of Nexus 2005 3D team [1]. Our aim has been to construct stable and flexible agent architecture for our further development and research. This architecture is organized such that in each release of server we can apply the changes to this architecture, easily [2].

2 Agent skills

As the primitive commands are rather difficult for the decision-making component to use directly, we develop some high level skills which are easy and convenient to use. These skills include “dash to some position”, “kick the ball to some position with a specified 3D velocity”, “shoot the goal to a specified point”, “dribble to some target position with a specified velocity” and so forth. And the decision-making component should decide to take some action to gain the highest profit according to the world model.

The Skills are an abstraction to low level server commands and should provide the decision layer with high-level commands, like ’go to a certain position’ or ’kick the ball with a certain speed’. The intercept skill (either of the ball or a position) will be implemented in a kind of greedy fashion, i.e. drive in the direction of the target and slow down soon enough.

2 Action evaluation

To determine the best action among all possible ones carried out by the ball controller agent for a given situation, we first recognize the best of each action, i.e., the best shoot, the best dribble, and the best pass, independently. It is clear that, when the best possible shoot is sought the parameters that affect the shooting action are considered, only. For dribble and pass actions the same kind of process is followed. In the next phase, we select the best of bests, i.e., the system chooses the best action among three best actions shoot, dribble, and pass. In this phase, common measures are used in order to evaluate the actions. To determine the priority in the second step, the calculated priorities in the first step is not considered [2].

3 Team strategy

We use very simple team strategy, that is when the agent is the fastest player to get the football, and then it tries to intercept the ball; when the agent is not the fastest, it drives to its strategic position; when the agent can kick the ball, it kicks the ball directly to the opponent's goal, passes the ball to one of its teammates or performs a long dribble.

5 Future works

Our future work would mainly base on implementation of shoot module using Reinforcement Learning approach. After tight considerations of RL methods we decide to implement Q-Learning for this matter. We have our Q-Table initially filled by manually collected data set. Each row of Q-Table represents a state. We consider 3 parameters for defining a state which are: ball distance to goal line, angle between ball and goal sides, and goalie distance to goal line. Uniform clustering is used to quantify them as ball distance to goal line are 21 clusters, angel between and goal sides (Left bar, Right bar) are 45 clusters and goalie distance to goal line are 5 clusters. Thus, in this problem, the number of states is 21*45*5 which sums up to 4725.

References

1. Salmani, V., Seifi, F., Moienzadeh, H., Milani, Fard A., Naghibzadeh, M.: Nexus 2005 - 3D Team Description. Proceeding of Robocup 2005, (2005)
2. RoboCup Soccer Server 3D Maintenance Group (2003). The RoboCup 3D Soccer Simulator. Available at http://sserver.sourceforge.net/
3. Salmani, V., Naghibzadeh, M., Seifi, F., Taherinia, A.:A Two-Phase Mechanism for Agent's Action Selection in Soccer Simulation" , The Second World Enformatika Conference, WEC'05, Istanbul, Turkey, (2005)
4. Salmani, V., Milani, Fard, A., Naghibzadeh, M.: A Fuzzy Two-Phase Decision Making Approach for Soccer Simulation Agent, Submitted to the IEEE International Conference on Engineering of Intelligent Systems, ICEIS'2006, Islamabad, Pakistan

[1] P. Nordin and M. Nordahl, "An Evolutionary Architecture for a Humanoid Robot," *In Proceeding of Symposium on Artificial Intelligence (CIMAF99)*, Havana Cuba, 1999.
[2] O. Kajita Matsumoto and M. Saigo, "Real-Time 3D Walking Pattern Generation for a Biped Robot With Telescopic Legs," *In Proceedings of the IEEE International Conference on Robotics and Automation*, May 21–26, 2001, Seoul, Korea, pp. 2299–2308.
[3] Fujitsu Automation, "Humanoid Robot HOAP-2 Specification." *Fujitsu Corporation Web site*, http://jp.fujitsu.com/group/automation/downloads/en/services/humanoid-robot/hoap2/spec.pdf.
[4] M. Vukobratović and B. Borovac, "Zero-moment point, Thirty five years of its life," *International Journal of Humanoid Robot*ics, *Vol. 1*, *No. 1*, 2004, pp. 157-173.
[5] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi and K. Tanie, "Planning walking patterns for a biped robot," *IEEE Transactions on Robotics and Automation*, 2001, pp. 280-289.
[6] A. Boeing, S. Hanham and T. Braunl, "Evolving autonomous biped control from simulation to reality," *In Proceedings of the 2nd International Conference on Autonomous Robots and Agents*, December 13–15, 2004, ACM press, New Zealand, pp 440–445.
[7] L. Yang, C. Chew, T. Zielinska, and A. Poo "A uniform biped gait generator with off-line optimization and online adjustable parameters," *Robotica*, *Vol. 25*, *Issue 2*, Cambridge Press, March 2007, pp. 1-17.
[8] M. Eaton and T. J. Davitt, "Evolutionary control of bipedal locomotion in a high degree-of-freedom humanoid robot: first steps," *Artificial Life and Robotics*, *Vol. 11*, *No. 1*, Springer, pp. 112-115.
[9] A. Boeing and T. Braunl, "Evolving splines: an alternative locomotion controller for a bipedal robot," *In Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision (ICARCV¢02)*, December 2002, Singapore, pp 798–802.
[10] J. Boedecker, "Humanoid Robot Simulation and Walking Behaviour Development in the Spark Simulator Framework," *Artificial Intelligence Research University of Koblenz*, April 2005.
[11] J. Hoffmann, U. Düffert, "Frequency Space Representation and Transitions of Quadruped Robot Gaits," *In Proceedings of the 27th Australasian Computer Science Conference (ACSC 2004)*, Vol. 26, University of Otago, New Zealand, 2004, pp 275 – 278.
[12] cyberbotics.com.

Soccer simulation as an effort for motivating researchers in the field of artificial intelligence and robotic research has always been a progressive approach. Robotic soccer is a particularly good domain for studying multi-agent systems and behaviors. In this paper, we describe researches done by Nexus team from the prior 2D soccer simulation environment to the curent humanoid simulation version. The main development features were done on decision making, action selection, and coach strategy making modules using fuzzy logic mechanism and game theory approach. Some very basic humanoid actions are also explained.

1. Introduction. Robotic soccer is a particularly good domain for studying multi-agent systems. It has been gaining popularity in recent years with international competitions like RoboCup which is planned for the near future [1]. Soccer simulation environment is a client-server platform which provides an excellent testbed for development of multi-agent systems. Using this testbed frees researchers fomr getting involved in the complexities of physical robot developmets.

Nexus, established in 2002, is the RoboCup soccer simulation team of Ferdowsi University of Mashhad, Iran. As an important first step, the team was qualified to participate in RoboCup contest in 2003 Padova, Italy, in Soccer-2D league. Afterwards, Nexus could go as high as the third round in RoboCup 2005 Osaka, Japan, and ranked between 9th and 12 th place among 33 teams. The team has also participated and won some domestic leagues in Iran during these times. Currently, Nexus has become eligible and was qualified to participate in the RoboCup 2007 Atlanta, US, in the humanoid soccer robot simulation league. In this paper, we propose a comprehensive review of our research projects done in the RoboCup simulation filed from the early stablishement of the Nexus team.

The current development of 3D Soccer Simulation League leads towards humanoid robots known as *soccerbot* agent, which already can be controlled by a lower level interface. However, controllers for these robots have to be developed in order to provide an easy-to-use interface. The rules has been maturated in many points and gained focus on the issues that are essential from a technical point of view. Thus, the center of mass of all robots has to be on a certain height in relation to the size of the feet. Fundamental for playing soccer are the abilities to walk and to kick. As body contact between the physical agents is unavoidable, the capability of getting up after a fall is also essential. For keeping a goal, the robot must be able to perform special motions.

4.1. Walking skill. Transferring the weight from one leg to the other, shortening the leg not needed for support, and leg motion along the walking direction are the key ingredients of this gait. Walking forward, to the side, and rotating on the spot are generated in a similar way. As the first step toward a skillful humanoid agent, walking is performed with a traditional control method that follows a set of generated ZMPs[1] along the path. This working dynamic model for biped robot walking is shown in Fig.6.

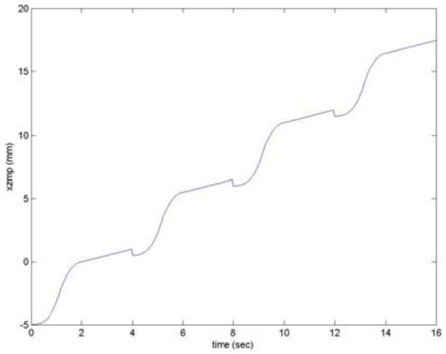


Fig.6 ZMP trajectory

The trajectory tracking methods (specially generated by a series of ZMPs) to control the agent balance while moving has been investigated in [15]. Generated trajectory is followed by a precise controller. The controller, knowing the exact path of the agent's joints, determines the velocity of the joint motors to direct different parts of the robot along the computed path. The walking skill of our agent is depicted in Fig.7.

[1] Zero Moment Point

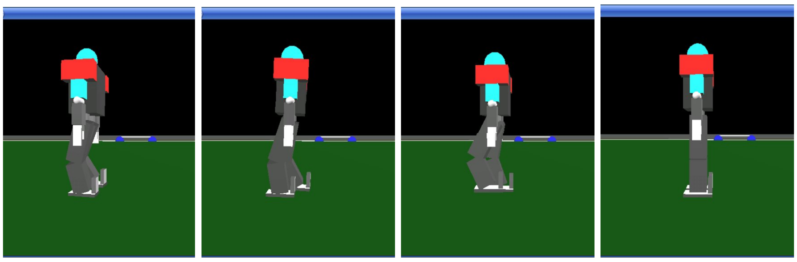


Fig.7 Soccerbot walking skill

4.2 Kicking skill. After inhibiting the walking behavior and stopping, the robot moves its weight to the non-kicking leg and then shortens the kicking leg, swings it back and accelerates forward. The kicking leg reaches its maximal speed when it comes to the front of the robot. Same principles for keeping robot's balance while walking or running are applied in performing actions like kick or dribble. The effectiveness of using dynamic methods like following the path generated by ZMPs with the help of new control methods like fuzzy PID control is already proved in such fields [16, 17].

4.3. Goalie dive skill. The goalie is capable of diving into both directions. First, it moves its center of mass and turns its upper body towards the left while shortening the legs. As soon as it tips over its left foot, it starts straightening its body again. While doing so it is sliding on its hands and elbows. These steps are depicted in Fig.8.

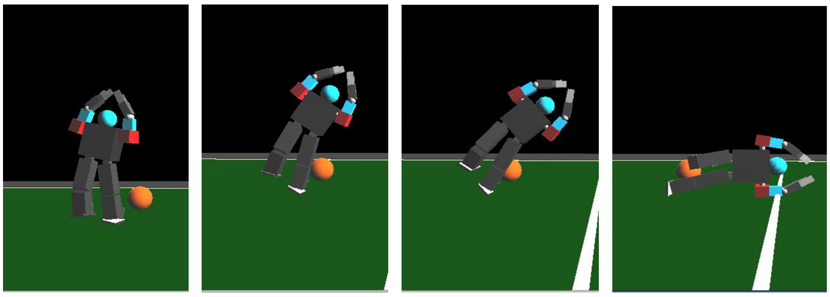


Fig.8 Soccerbot diving skill

5. Future work. Further improving the controller will be the next stage. Number of learning and optimizing methods such as artificial neural networks, genetic algorithms and other evolutionary approaches will be considered to give the controller an adaptive smooth behavior. For example genetic algorithm could be used to search the trajectory path, computed by the traditional dynamic model, with a small margin to achieve a better walking performance. Fuzzy logic, as a powerful tool in dealing with imprecise environments, can also improve the performance of the designed controller.

References: 1. *H. Kitano, Y. Kuniyoshi, I. Noda, M. Asada, H. Matsubara, and E. Osawa.* RoboCup: A challenge problem for AI. AI Magazine, 18(1):73–85, Spring 1997. **2.** *M. Chen, E. Foroughi.* Robocup Soccer Server manual 7.07, August, 2002. RoboCup Federation. available: http://sserver.sourceforge.net. **3.** *S. J. Russell, P. Norvig.* Artificial Intelligence A modern Approach, Prentice Hall, 1995. **4.** *V. Salmani, M. Naghibzadeh, F. Seifi, and A. Taherinia.* A Two-Phase Mechanism for Agent's Action Selection in Soccer Simulation, The Second World Enformatika Conference, WEC'05, Istanbul, Turkey, pp. 217-220, February 2005. **5.** *V. Salmani, A. Milani Fard, M. Naghibzadeh.* A Fuzzy Two-Phase Decision Making Approach for Simulated Soccer Agent, IEEE International Conference on Engineering of Intelligent Systems, pp. 134-139, Islamabad, Pakistan, April 22-23 2006. **6.** *B. Kosko*. Neural Networks and Fuzzy Systems, 1992, Englewood Cliffs: Prentice Hall. **7.** SoccerDoctor v1.0, developed by Wright Eagle, 2002. Available at: http://wrighteagle.org/sim/tools/SoccerDoctor.exe. **8.** *A. Milani Fard, V. Salmani, M. Naghibzadeh, S. K. Nejad, H. Ahmadi.* Game Theory-based Data Mining Technique for Strategy Making of a Soccer Simulation Coach Agent, the 6th International Conference on Information Systems Technology and its Applications, (ISTA2007), May 23-25, 2007, Kharkiv, Ukraine. **9.** *J. M. Vidal.* Learning in Multiagent Systems: An Introduction from a Game-Theoretic Perspective In Eduardo Alonso, editor, Adaptive Agents: LNAI 2636. Springer Verlag, 2003. **10.** *A. Rubinstein.* Finite automata play the repeated prisoner's dilemma. ST/ICERD Discussion Paper 85/109, London School of Economics, 1985. **11.** *A. Neyman.* Bounded complexity justifies cooperation in finitely repeated prisoner's dilemma. Economic Letters, pages 227-229, 1985. **12.** *M. J. Kearns and U. V. Vazirani.* An Introduction to Computational Learning Theory. MIT press, Cambridge, Massachusetts, 1994. **13.** *V. Salmani, F. Seifi, H. Moienzadeh, A. Milani Fard, M. Naghibzadeh.* Nexus 3D 2005 Team Description, RoboCup Soccer simulation team description, International Symposium of RoboCup, 2005, Osaka, Japan. **14.** *M. Mozafari, A. Milani Fard, V. Salmani, M. Naghibzadeh.* An Improved Fuzzy Mechanism for 3D Soccer Player Agent's Shoot Skill, IEEE INDICON 2006, September 15-17, 2006, New Delhi, India. **15.** *Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie.* Planning walking patterns for a biped robot, IEEE Tr. On Robotics and Automation, vol. 17, no. 3, pp. 280–289, 2001. **16.** *T. Yanase, T. Iba.* Evolutionary Motion Design for Humanoid Robots, GECCO’06, July 8–12, 2006, Seattle, Washington, USA. **17.** *A. Akramizadeh, A. Hosseini, M. Akbarzadeh.* Design, construction and fuzzy PID control of a 3DOF robot manipulator, CIS2001, Conference on Intell. Sys., Khaj-e-Nasir Univ., Tehran, IRAN, 2001.

An Evolutionary Gait Generator with Online Parameter Adjustment for Humanoid Robots

A. Zamiri, A. Farzad, E. Saboori, M. Rouhani, M. Naghibzadeh, A. Milani Fard

Abstract

This article proposes a new hybrid methodology, together with an associated series of experiments employing this methodology, for an evolutionary gait generator that uses trigonometric truncated Fourier series formulations with coefficients optimized by a Genetic Algorithm. The Fourier series is used to model joint angle trajectories of a simulated humanoid robot with 25 degrees of freedom. The humanoid robot in this study learns to imitate the human walking behavior on flat terrains in a dynamically simulated environment. The simulation result shows the robustness of the developed walking behaviors even in extremely high and low speeds providing appropriate frequency. Number of range limitations were applied to the genetic algorithm used in this research to improve the learning period to less than 48 hours. The research seeks to improve upon the previous works on evolutionary gait generation, in robots with lower degrees of freedom. In addition, the proposed solution adapts a hybrid approach, thereby avoiding the long learning curves and unstable and slow gaits associated with evolutionary approaches.

Introduction

Building robots that imitate human behavior to perform their actions like walking and running is amongst the most popular, and at the same time most complex tasks, in autonomous system design. This complexity is mainly due to the difficulty to cope with the many Degrees of Freedom (DOF) of a humanoid robot. This high number of degrees of freedom in a biped mobile robot creates new problem spaces in control and navigation where conventional methods often fall short [1]. To reduce the complexity of the analysis, some researchers adopted a simplified dynamic model such as the inverted pendulum with certain assumptions on the robot's motion and structure [2]. While these simplifications come handy in designing initial trajectories, there still exist significant differences between the dynamics of a simple bipedal robot and a genuine humanoid robot with a high DOF.

A popular approach used for joint trajectory planning for bipedal locomotion is based on the Zero Moment Point (ZMP) stability indicator. In many ZMP-based trajectory planning approaches, motion planning is presupposed and performed in the Cartesian space [4, 5].

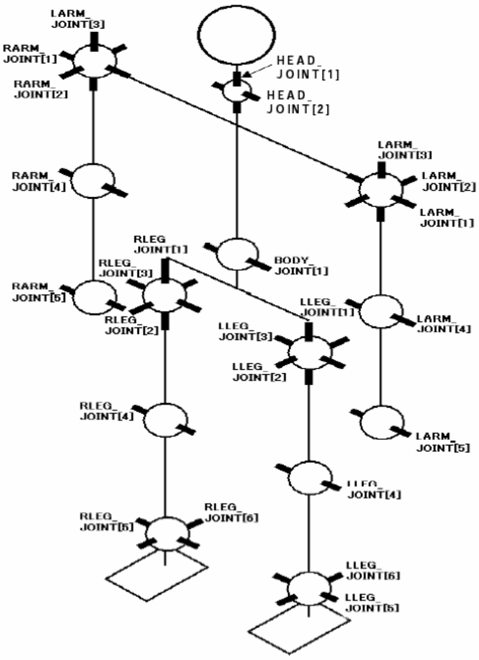
Hence, evolving control systems for robot locomotion is becoming a standard approach for the generation of improved or newer control systems for robots [6]. Various learning approaches for bipedal locomotion have been proposed by several researchers. Lin Yang et. al presented the Genetic Algorithm Optimized Fourier Series Formulation (GAOFSF) method for stable gait generation in bipedal locomotion in [7]. They use Truncated Fourier Series (TFS) formulations together with a ZMP stability indicator to generate the feasible gaits for a simple seven-link planar robot. A genetic algorithm is then utilized to search for optimal gaits according to the objective functions considering the specified constraints.

Some researches have used genetic algorithm to directly generate joint trajectories for each step [8, 9]. These trajectories are then applied to joints repeatedly while walking. Although this method is successfully utilized for biped robots, the generated gaits can not be changed to achieve desirable real-time motion adjustment.

In this paper, automatic evolution of walking behavior in a simulated humanoid with online adjustable speed is discussed. The robot in this study is a simulated model of Fujitsu's HOAP-2 that is a genuine humanoid with two arms and two legs and 25 DOF (Figure. 1). The simulation is performed by Spark, a generic three-dimensional physics simulator based on 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 [10]. Robot models simulated in Spark can be easily controlled using programming languages like c++ and java. A simple PD controller was implemented to control the joint motors, but due to the adaptive nature of evolutionary methods, any other type of controller can be used.

Since the movements of the robot are known to be periodic while walking on flat plains, the motion of every joint can be expressed in terms of a trigonometric truncated Fourier series. Coefficients of the Fourier series are determined by using a genetic algorithm. Each individual in the genetic algorithm contains a set of coefficients for every joint's Fourier series, and thus defines a gait. These gaits are then tested in the simulation environment until the robot falls down over the ground or a sufficient amount of time passes by. The fitness is calculated based on the forward movement of the robot and the total time of the test. Once all individuals of the current generation are tested, the next generation is generated by applying GA operators over the best fit individuals. Using this method, a relatively fast and stable walking gait is evolved within a three-day simulation time on a Pentium IV 2.8GHz machine with 1GB of physical memory.

2. Truncated Fourier series



Evolutionary approaches include trying to optimize the parameters of a given type of motion model [11]. Common to these methods is the fact that a certain amount of knowledge of how locomotion is performed is implicitly present in the model. This narrows down the search space thus reducing the time needed for the optimization process. For a deeper look into biped motion properties we recorded and processed HOAP-2 (Figure. 2) walking gait which is included in the Webots simulation software [12].

Figure. 2(a) and Figure. 2(b) give the knee and hip trajectories for the HOAP-2. These trajectories are identical in shape for both legs, but are shifted in time relative to each other by half of the walking period. The gait period is given by 2π/ where is defined as the gait frequency in radians per second (rad/s). Because of the periodic nature of the motion we can formulate joint angles using Fourier series. The Fourier Series of a periodic function of time f(t) can be written as:

$$f(t) = a_0 + \sum_{n=1}^{\infty}(a_n \cos\frac{2n\pi}{T} + b_n \sin\frac{2n\pi}{T} t)$$

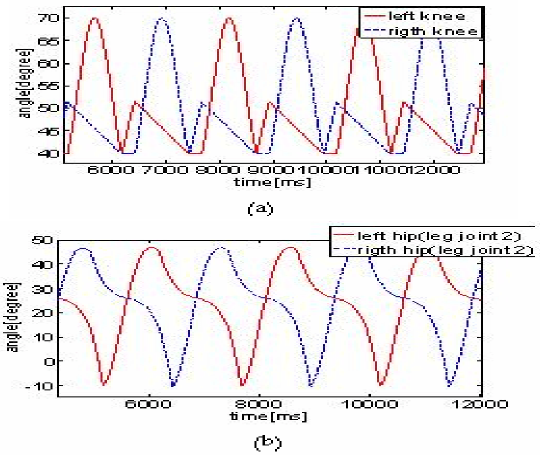
Where a_n and b_n are constant coefficients and t is the period. The period can be calculated by desired fundamental frequency ωby t= 2π/ ω

Since the servo motors used in the robot act as low pass filters, we expected to be able to omit higher order frequencies without decreasing performance because these frequencies cannot contribute to the motion that is actually being executed by the robot. Thus we use Truncated Fourier Series (TFS) which have only the 3 first terms of trigonometric form of Fourier series.

$$f(t) = A\left[a_0 + \sum_{n=1}^{m}(a_n \cos\frac{2n\pi}{T} + b_n \sin\frac{2n\pi}{T} t)\right]$$

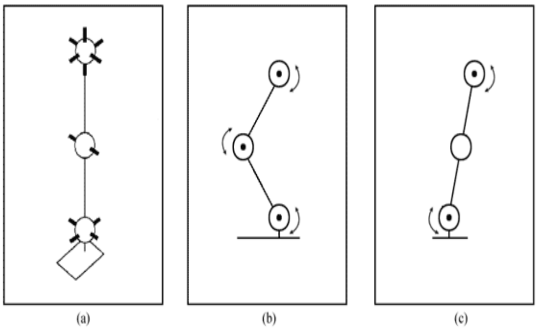
Where a_n, b_n and t are same as Equation 1, and a is an amplitude scaling parameter used for changing the step length. The parameter m determines the number of terms in the Fourier series. Using this formula significantly reduces the subsequent computational load in the search for feasible and optimal solutions using GA which is discussed in next section.

One of the advantages of this approach is that trajectory generation is done directly in the joint space. As such, inverse kinematics computation is not required thus avoiding the singularity problem. The walking rhythm, speed, and walking pattern can also be adjusted online through tuning either a single or two parameters.



3. The genetic algorithm

As discussed in the previous section of this study, each joint trajectory can be modeled by a truncated Fourier series. A genetic algorithm was used to search for the optimal values of the coefficients so as to achieve stable walking behavior with desirable char- acteristics for the robot model. The first 3 terms of the general trigonometric Fourier series were used in the formulations. This is due to the fact that very high frequencies are normally filtered by the joint motors. The frequency of the Fourier series is considered as a constant value and can be changed after the offline learning is finished. This way every Fourier series could be represented by 7 real numbers and as for HOAP-2 model, 25 series are required to define a gait (One for each joint). This makes the size of the individuals relatively large and as a result the learning time gets very long. Therefore, some improvements should be taken into consideration to make the learning process more effective.



Joints of the left part of the robot’s body get the same periodic values as the right ones while walking straight, with a delay equal to a period time and there’s no need to consider a separate set of Fourier series for the right part in the chromosome structure. This makes the chromosome almost half in size. Besides, the joints in charge of controlling the foot can get their values from the knee and hip joints as they could always be kept parallel to the ground line. (Figure. 3) Such automatic control of the foot is extremely beneficial in terms of keeping robot's balance, especially in the early stages of the learning phase. Head joints can be also ignored in the gait for the sake of simplicity.

With omitting excessive joints, the final chromosome will contain 6 real numbers for every one of the 12 primary joints. Considering this relatively large chromosome size, there are 300 individuals per generation, and 900 generations.

The basic GA algorithm is as follows:

- A random number generator is used to give random real values to Fourier series coefficients in chromosomes.

- The robot resets and locates in the initial position in the simulation environment.

- Chromosomes are tested one by one. In every simulation step, the motor value for each joint is calculated using a PD controller, based on the corresponding Fourier series that is defined by the current chromosome.

- Each test continues until the robot falls over or 60 seconds has passed by. After the test is finished the fitness is calculated on the basis of the time passed multiplied by the distance from the start point and the next test starts.

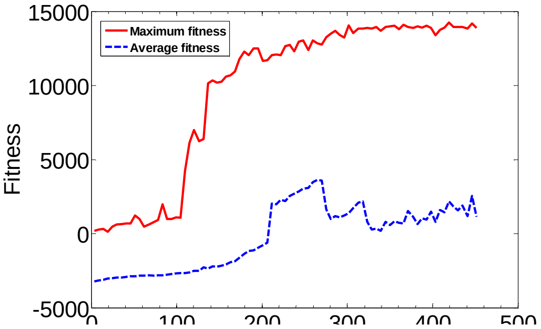
- Once all chromosomes have been tested, the roulette-wheel selection decides which individuals are allowed to reproduce using mutation and crossover with specified probability.

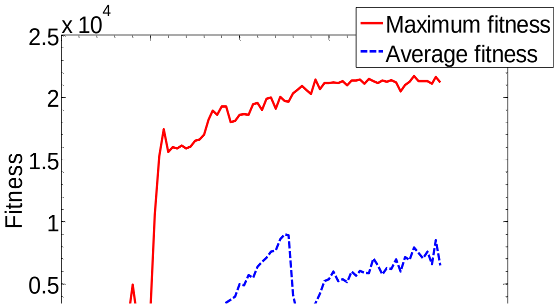
A custom crossover was implemented that consists of a simple two-point crossover together with a creep operator. The creep operator randomly increases or decreases some real values of the chromosome by a very small value between 0 and 5. This crossover was applied with the probability of 0.6 together with a mutation with probability of 0.05. Valid range for the coefficients was also limited to -50 to 50. These values were arrived at after some experimentation.

4. Initial results

The learning process takes about 7 days to complete. However some decent walking behaviors begin to emerge within the second day of the evolution. These walking gaits are somewhat unstable and do not follow a forward straight line. After the 4th day of the process, stable gaits evolve gradually. Figure. 4 shows the average and maximum fitness values for the robot over 460 generations. These figures are averaged over 3 runs. Note that although elitism was employed, because of the accurate physics simulation with noise, etc., this allows for the maximum fitness to fall as well as rise from generation to generation.

Studying the initial simulation results with different frequencies reveals that almost all of the generated gaits during the learning phase convergence to a human-like walking behavior with similar movement patterns. These similarities include human-like movements of the hand and sinusoidal movements of the waist.



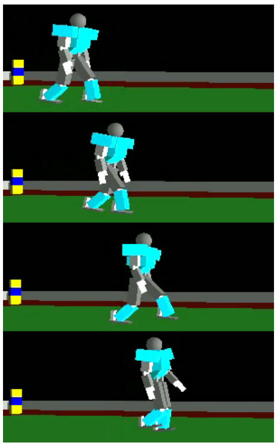


5. The improved algorithm

In the initial experiments it was found that some joint trajectories are similar in all of the generated gaits. These common trajectories share the same characteristics and differ in terms of their speed and frequency. For example, the elbow joint bends a little at the start of the walking and stays almost the same during the rest of the movement. Studying these similarities helps to further improve the evolutionary learning process by removing the unnecessary parameters. Experiments show that stable straight walking gaits can be generated by using only two joints of the leg: the hip joint 3 and the knee joint. This simplification affects the learning process dramatically and decreases the learning time to less than 48 hours (Figure. 5).

Based on the initial experiments, the fitness function was also changed in order to achieve better results. In this stage the time factor was ignored for the gaits that keep the robot walking for more than half of the test time. This helps faster gaits to get more fitness value over the stable but slow ones. Furthermore, the average amount of deviation was taken into account so that straight walks have more chance to be selected for regeneration. The simplified model of fitness function is as follows:

```
if (CurrentTestTime < TotalTestTime / 2)
    Fitness := Time * Distance
else
    Fitness := Distance - AverageDeviation
fi
```



6. Online parameter adjustment

Walking behavior can be adjusted through changing one or two parameters. By changing the Fourier series' frequency and joint movement gain, desired walking speed can be achieved dynamically. In the final experiments we were able to stop the robot after some periods of walking by gradually increasing frequency and decreasing controller's gain.

The movement direction can also be determined by the leg joint 1. This joint is responsible for rotating thigh around Z axis. The value of this joint was kept unchanged during the learning process, but it can be slightly modified while walking to make the robot change its direction a few degrees. Figure. 6 shows the simulated robot in Spark simulation environment while walking with maximum speed.

7. Summary and future works

Using a set of Fourier series with parameters adjusted by a genetic algorithm, a simulated robot has been able to teach itself how to walk. We have described how to use an evolutionary algorithm to search for the optimal values for series coefficients to develop quite realistic walking sequences through the manipulation of up to 24 of the 25 motors used in the robot in relatively short time spans (certainly compared to human walking). The different evolutionary techniques used, such as custom cross-over operator and roulette wheel selection, were briefly described. The development of the algorithm used to determine the best walk sequences has been described in some detail. Work is continuing on using this technique for the development of controllers for humanoid robots, by again extending the number of degrees of freedom, and by the application of the techniques described here to different robot morphologies and environments. We feel that as well as aiding in the future development of humanoid robot locomotion, further research in this area may help in understanding certain aspects of human walking.