2013 Second International Conference on Robot, Vision and Signal Processing

Evolution of Neural Controllers for Simulated and Real Quadruped Robots

Sehar Shahzad Farooq
Department of Computer Science and Engineering
Sejong University, Seoul, Republic of Korea
E-mail: sehar146@gmail.com

KyungJoong Kim*
Department of Computer Science and Engineering
Sejong University, Seoul, Republic of Korea
E-mail: kimkj@sejong.ac.kr

***Abstract*—Evolutionary robotics is an approach that employs evolutionary computation to develop a controller for an autonomous robotic system. Evolutionary computing usually operates depending on a population of candidate controllers, initially selected from a random distribution. The population is iteratively modified according to the fitness function. In this paper, an automatic control system is designed for quadruped robots using an Evolutionary Neural Network (ENN) and the performance is measured in terms of the distance travelled by the robot from its origin. The evolved neural controllers are analyzed in the simulation environment and the results are implemented in a real quadruped robot. The comparison between the simulated and real robot shows the performance of the quadruped robot in terms of number of iterations over the distance covered in the desired direction. The developed ENN helps the robot to choose the best possible solution to achieve the maximum distance.**

Keywords-component;Evolutionary Robotics; Quadruped Robot; Evolutionary Neural Controller; Systematic Analysis;

I. INTRODUCTION

The classical approach to controlling a robotic system consists of three modules i.e. perception, planning and action. First of all, while designing a controller for a robotic system, tasks need to be defined. After this, the instruction set is designed for the controller and, finally, the robotic system performs its actions in accordance with predefined strategies. However, a robotic system created in this manner suffers from uncertainty in unknown or changing environments [1].

It is thus necessary to make a robot which builds controllers automatically from interaction with environments. Instead of using predefined strategies, it is necessary to learn a new strategy suitable for the new or changing environment. The interaction means the robot tests its control mechanism and body shapes in the environment. Using some realistic simulators, the robot can run thousands of tests in a few seconds.

Under the name of evolutionary computation, there are a lot of different optimization algorithms inspired by biological evolution. Evolutionary robotics combines robot learning with evolutionary computation, and involves evolving a robotic control system from scratch using genetic operators. Based on the output of the learning phase, the algorithm suggests the most feasible and suitable action which the robotic system should perform [2]. In this paper, we developed an automatic control system in which the robot learns about the environment by first using an ENN algorithm. Based on the interaction with the environment, the robot determines a number of things, such as 1) direction to move, 2) the step size it should take, and 3) the pattern of steps. We built our simulator based on the open evolutionary robotics platform "LUDOBOTS" [3]. The best results from the simulation are implemented on a quadruped robot, which is created using the Bioloid Premium Kit. The experimental results with a simple quadruped robot in the simulation, and its response in the real robot, show that evolution could generate walking behavior automatically. The comparison between the simulation and real robot response is visualized, and the difference – the reality gap – is also discussed. In addition to discussing the evolution, we attempt to analyze the properties of the neural networks evolved to see their working mechanism. It is challenging to analyze the results from the automatic design but it can give some insights to the human designers. Fig. 1 shows an overview of the analysis and comparison.

The remaining paper is organized as follows. In Sec. II background knowledge about the design of the quadruped robot is discussed. Sec III explains the simulated and Bioloid quadruped robot, Sec. IV explains the evolutionary neural network, Sec. V provides the analysis of evolved neural controllers, Sec. VI shows the comparison between the simulated and the real quadruped robots and Sec. VII includes the conclusions and the future work.

II. BACKGROUND KNOWLEDGE

Traditional wheeled robots need plane surfaces in order to enter the complex environments as a substitute of human being. There have been many legged robots who overcome these demands by adopting rugged roads and obstacle avoidance phenomena. Among those legged robots, quadruped robots obtained high significance over biped and hexapod robots due to easy and efficient construction, maintenance, control and stability.

In our design, quadruped robot structure is opted because it provides better stability over biped robot and lesser redundancy and complexity over hexapod robot. However the design of leg for our robot is different from animals in a way that animal's leg consist of more than two DoF, the

* Corresponding author: kimkj@sejong.ac.kr

978-1-4799-3184-2/13 $31.00 © 2013 IEEE
DOI 10.1109/RVSP.2013.73

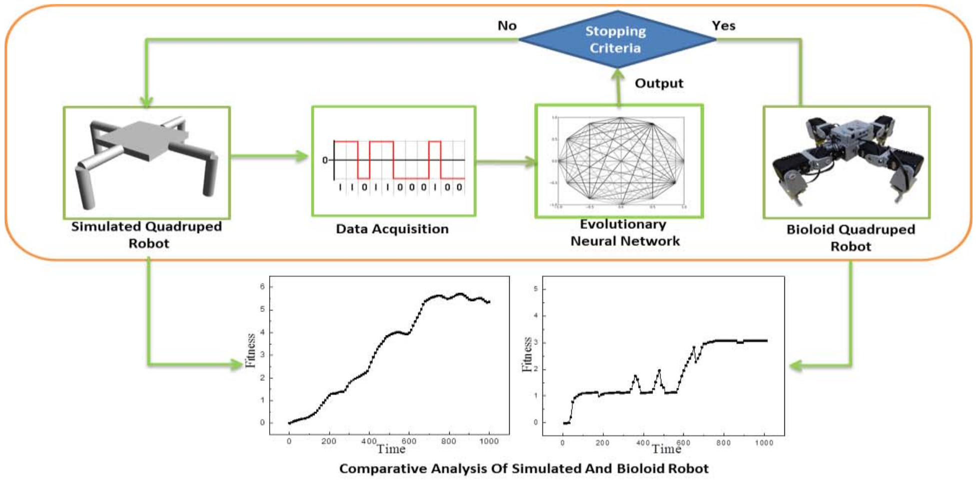

Figure 1. An Overview of Analysis and Comparison of Simulated and Real Quadruped Robot

Ankle, the Knee, the Hip joint and more specifically another DoF in order to turn around. There have been many robots following the biological concepts. Arikawa worked on TITAN family robots using three DoF per leg [4]. Ma et al. in 2011 presented a mechanism for optimizing simplified mule robot using practice swarm optimization algorithm [5]. Pei et al. developed Hydraulic quadruped robot which use Matsuoka's central pattern generator to control the movement of each leg constitute of multiple DoF per joints [6]. But these designs focus on three or more than three DoF per joints. In [7], Holonomic omni directional mobile robot decoupled with three DoF motion control via three actuators considering two degrees for translational motion and one degree for rotational motion. But it didn't consider the legged structure. We borrowed its strategy using legged robot and design our quadruped robot with lesser DoF.

Three legged robot can also be chosen but it's hard to keep the balance in case of any damage of the leg which is our future work. So it is preferable to focus on quadruped robot [8].

III. The Robot

A. A Simulated Robot

The design of a robot for the simulation required a physics engine. For this purpose, ODE (Open Dynamics Engine) is used. ODE is an open source physics engine provided with a number of built-in test applications. We modified one of the applications (simulation) and design our quadruped robot depicted in Fig. 2 (A).

The developed quadruped robot consists of a main body which is connected with four upper legs (Hips) and these legs are further connected with four lower legs (Knee). The length and width of the body is 1.25m while the depth is 0.15m. The height of the body is adjusted by the lower legs length. The length of upper and lower legs is kept the same during simulation for the analysis.

Each part of the body is connected with a hinge joint. The constraints of two parts are in the same location and in line to the hinge axis. The movement of joints is limited between [-45^{o}, 45^{o}]. These joints are then attached to motors that force the legs either inward or outward of the joints with a force capable to react sufficiently. The knees can move back and forth while hips can move up and down. With the help of knees and hips, the robot body can move to the desired direction. The speed and force of the motor is controlled and adjusted using the proposed algorithm to avoid the robot from jittering.

Moreover, a binary touch sensor is connected at the bottom of each lower leg which can provide information of the collision of the robot with the ground. Each sensor fires when it collides with the ground and stops when the leg is off the ground. The force of gravity is set to $-9.8m/s^2$. In this paper, we used built-in collision detection mechanism provided by ODE. Initially, the robot is set to zero position in the simulation window and the fitness measures how far the main body of the robot moves at a particular iteration.

B. A Bioloid Robot

The real robot is designed using BIOLOID premium kit which is one of the commercial products of ROBOTIS. Bioloid premium kit provides everything to build dozens of top performing different robots. It includes different typed and shaped plastic body parts with versatile expansion mechanism, various sensors, remote control capability, USB interface, Dynamixel actuators and CM-510 microcontroller. We use two degrees of freedom per leg in our design. The design of our quadruped robot is depicted in Fig. 2 (B).

It consists of a main body built using part F51 and F52 of Bioloid Premium Kit. At the top of the body, CM-510 main microcontroller is attached which is supplied with the battery from the bottom of the main body. Two Dynamixel AX-12A actuators are attached with the body's left and right side using part F7 on the top and bottom of each actuator, while two more actuators are attached at back and forth using part F3. These four actuators are further connected with four more Dynamixel actuators using part F2 along with part F3. Part F4 is attached to each Dynamixel rotary path forming the Knee of the robot, further connected with a touch sensor OTS-10 supported by part F6 at the bottom. These four

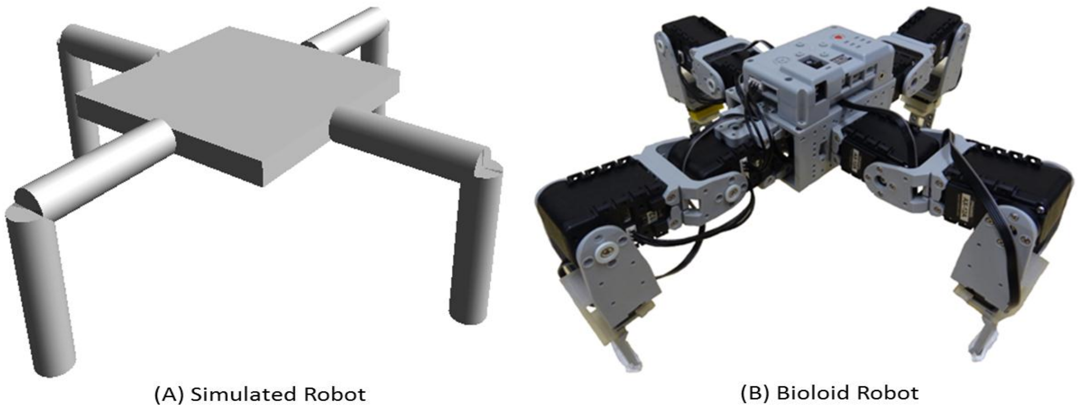

Figure 2. Physical Design of a Quadruped Robot

actuators provide inward and outward motion of the legs. The length of the all eight legs is 120mm while the body dimensions are 80*100mm. The weight of the robot is 0.65kg.

In order to measure the distance covered by the robot, a Logitech HD Pro webcam C910 camera is placed 150cm higher from the ground which captures the latest position of the robot per millisecond. This captured image is then used to measure the distance covered by the robot from its origin. Because the quadruped robot is connected with the computer using serial port, the communication speed of the serial port is set to 57600 which is the maximum speed supported by CM-510.

IV. EVOLUTIONARY NEURAN NETWORK

ENN is one of the suitable techniques to find the best possible solution out of a number of candidate solutions for the problem. In a typical ENN, the solution is optimized in such a way that the fitness of individual solution (in a set of random population) is measured at first using the fitness objective function, then those candidate solution with lower fitness are discarded and the remaining solutions are modified. Among those remaining solutions, each one is again examined and the candidate solution with the highest fitness value is considered to be the best solution.

The process of ENN is started with a random population. A 4x8 matrix is randomized between values [-1, 1]. The order of the matrix is set because the quadruped robot has four sensors and eight actuators. Random values from [-1, 1] give a directional angle to move hips and knees either inward or outward. The robot covers a distance from its origin by using a pattern of movements defined by ENN. The positive distance covered by the quadruped robot is measured as the fitness of ENN. The process is repeated for three hundred iterations and the matrix with the highest fitness is considered to be the best solution.

The ENN performs evolution on a set of matrix which contains the information of motor angles that are to be applied to the quadruped robot. The input to ENN is given by the sensors while the output is applied to motors attached to each joint of the robot. Inputs are binary values while the synapse weights consists of continuous values from [-1, 1], which gives continuous output values in this range. The synapse value indicates the influence of source neuron to the target neuron. The activation of a neuron for the next time step can be expressed as

$$a_i = \sigma\left(\sum_{j=1}^{n} w_{ij} a_j\right) \qquad (1)$$

Where n is the number of neurons connected to i, activation of j^{th} neuron is a_j, synapse weight w_{ij} connects j neuron to i, and σ ( ) is the thresholding function.

V. ANALYSIS OF EVOLVED NEURAL CONTROLLERS

There are twelve neurons scattered around 360° with an increment angle of 2π/n in Fig. 3. Each neuron influences on all other neurons. The gray lines show the negative while black lines show the positive synapse values. The width of the synapse lines indicates the magnitude of the synapse weight. The positive and negative synapses produce a pattern for the robot to move far from the origin.

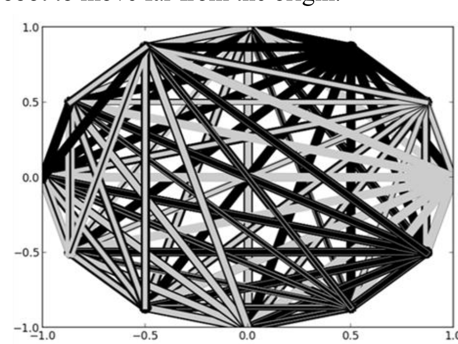

Figure 3. Effect of Synapse Weight Over the Best Neurons

The direction, step size and the pattern of steps for the simulated and Bioloid quadruped robots is depicted in Fig. 4. The robot moves towards right direction in counter clock wise rotation with only one leg at the ground while pushing from second leg. When the second leg touches the ground, the third leg pushes the robot in the same direction. The inward to outward movement of the actuator forces the robot to move in air while the other leg touched with the ground support the robot from falling down.

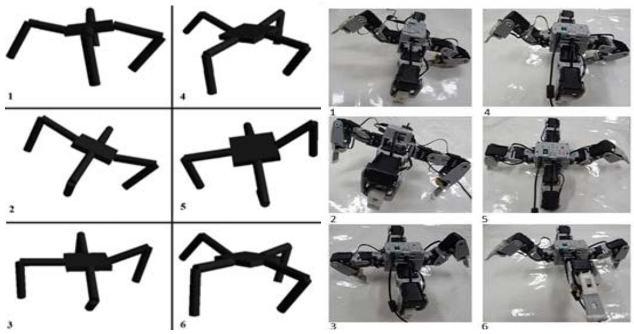

Figure 4. Simulated and Bioloid Quadruped Robot Movements

The graph in Fig. 5 shows the number of iterations and the average distance with standard error travelled by the robot per iteration. The increment in distance per iteration is the performance of ENN. It can be seen that the performance of the ENN increases rapidly at the start rather than to rise smoothly. While after several number of iterations, it becomes constant because of the higher distance covered by the robot in previous iteration and next ENN is not suitable enough to move the robot further. The maximum distance covered by the robot for a specific ENN is considered to be the best evolved.

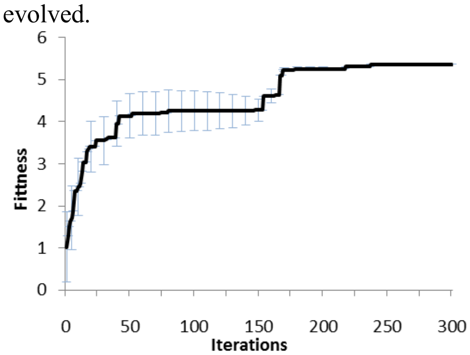

Figure 5. Fitness of ENN Over Iterations

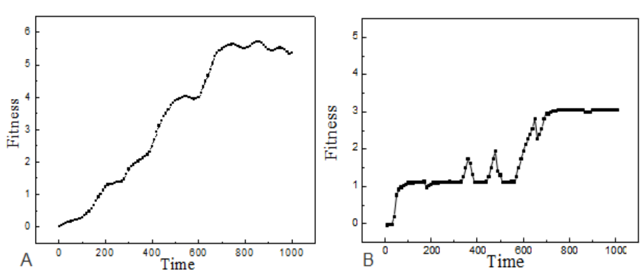

Figure 6. The Trajectory Movement of Simulated and Real Bioloid Quadruped Robot

VI. Comparison of Simulated Quadruped Robot and Real Quadruped Robot

To compare performance of the simulated quadruped robot with the real quadruped robot, the size of the simulated robot is made equal to the size of the real robot and the best ENN from the simulation is applied to real robot. The graph in Fig. 6(A) shows the trajectory motion of the simulated robot for the best ENN evolved, while the graph in Fig. 6(B) depicts the trajectory motion of the real Bioloid quadruped robot. The difference in the progress of the real quadruped robot with the simulated robot is due to the reality gap. Still there are many features that involve in the degradation of the real robot's performance such as weight of the robot, slippery surface, response of the touch switches and the processing speed of the actuators.

VII. Conclusion and Future Work

Although ENN outputs the fitness which is sufficient enough to optimize the quadruped robotic system but there is a room for improvement. In future, different evolutionary algorithms like Distributed Evolutionary Algorithm and NeuroEvolution of Augmented Topology (NEAT) can be applied to the same quadruped robot to obtain the maximum outcome. Furthermore, the simulation environment can be modified with some obstacles in the path and the control system is forced to avoid these hurdles and find a suitable direction to move. The Bioloid Quadruped can be modified with DMS sensors to analyze the objects around the path and to avoid collision. Instead of the simulation for the ENN, the evolution can directly be applied to the Bioloid quadruped robot and a batter and efficient response can be considered as the best evolved neural network. A fault tolerance in the quadruped robot can be introduced and an automatic control system for fault tolerant quadruped robot can be designed based on this work.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (2013R1A2A2A01016589, 2010-0018950).

References

[1] M. Givehchi, A. Ng, L. Wang, "Evolutionary optimization of robotic assembly operation sequencing with collision-free paths," Journal of Manufacturing Systems, vol. 30, Oct 2011, pp. 196–203.

[2] S.-I. Lee, S.-B. Cho, "Observational emergence in evolutionary fuzzy robotics," International Journal of Modelling, Identification and Control, vol. 3, May 2008, pp.88-96.

[3] Josh Bongard, Josh Bongard, (2013, March 28). LUDOBOTS Morphology Evolution & Cognition Lab, "Evolutionary Robotics". Retrieved from http://www.uvm.edu/~ludobots/index.php/Main/EvolutionaryRobotic

[4] K. Arikawa, S. Hirose, "Development of Quadruped Walking Robot TITAN-VIII" Proc. IROS, Nov1996.

[5] M. Ma, J.Wang, "Hydraulic-actuated quadruped robot mechanism design optimization based on practicle swarm optimization (PSO) algorithm," Artificial intelligence, management science and Electronic commerce (AIMSEC), 2nd International conference on, Aug 2011, pp. 4026-4029.

[6] Z. Pei, L. Song, B. Chen, X. Guo, "Adaptive control of a quadruped robot based on Central Pattern Generators," In industrial informatics (INDIAN), 10th IEEE international Conference on, Jul 2012, pp. 554-558.

[7] H. Asama, M. Sato,L.Bogoni, H.Kaetsu, A.Matsumoto, I.Endo, "Development of an Omni-Directional Mobile Robot with 3 DOF Decoupling Drive Mechanism," Robotics and automation, Proceedings IEEE international conference on, vol. 2, May 1995, pp. 1925-1930.

[8] S.-S. Farooq, K.-J. Kim,"The Systematic Analysis of Evolved Neural Controllers for Quadruped Robots" Korea Congress on Computer. 2013.