사족보행 로봇 제어를 위한 진화신경망의 체계적 분석

The Systematic Analysis of Evolved Neural Controllers for Quadruped Robots

Sehar Shahzad Farooq° KyungJoong Kim
Sehar146@gmail.com, kimkj@sejong.ac.kr
Department of Computer Science and Engineering, Sejong University, South Korea

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 the Quadruped Robots using Evolutionary Neural Network (ENN) and the performance is measured in terms of the distance travelled by the robot from its origin. The developed ENN helps the robot to choose the best possible solution that provide with the maximum distance. The evolved neural controllers are analyzed and the results show the performance of the quadruped robot in terms of number of iterations over the distance covered to the desired direction.

1. Introduction

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

It is necessary to make a robot which builds controllers automatically from the interaction with environments. Instead of the 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 (for example, physics-based simulators), the robot can run thousands of testing in a few seconds.

Under the name of evolutionary computation, there are a lot of different optimization algorithms inspired by biological evolution. The evolutionary robotics (ER) combines the robot learning with evolutionary computation. It evolves a controller of robots from the scratch using some genetic operators (selection, crossover and mutations). 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 using ENN algorithm first. Based on the interaction with the environment, the robot determines a number of things such as 1) direction to move, 2) how much 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 experimental results with a simple quadruped robot show that the evolution could generate walking behavior automatically. In addition to 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.

2. A Quadruped Robot

The design of a robot simulation required a physics engine. For this purpose, ODE (Open Dynamics Engine) is used. ODE is an open source physics engine to simulate rigid body dynamics. It is stable and mature platform. It has special features such as advanced joint types, collision detection with friction and more suitable for simulating vehicles in virtual reality environment. As mentioned earlier, ODE is an open source, therefore it is provided with a number of built-in test applications. We use one of these applications which simulate a buggy. We modified the buggy simulation and design our quadruped robot depicted in Fig. 1.

The developed quadruped robot consists of a main body which is connected with four upper legs (Hip) 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.

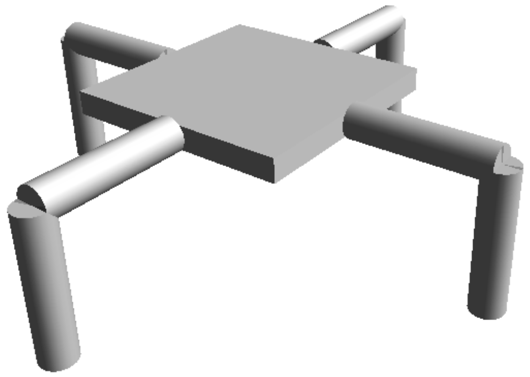

Fig. 1 Physical design of a quadruped robot

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°, 45°].

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.

Before applying ENN to the robot, it is checked with a fixed angle provided to each motor. The speed and force of every motor is adjusted to avoid the robot for jittering.

3. Evolutionary Neural 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 and is considered as the fitness of ENN. This fitness is compared with the parent fitness and if the fitness is greater than the parent fitness then child matrix becomes the parent and older parent matrix is discarded. The new parent matrix is perturbed with a specified value of 0.05 and new child matrix is created. The process is repeated for three hundred iterations and the matrix with the highest fitness is measured. The final parent matrix 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 weight matrix 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

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

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

ENN interface with ODE simulation such that a 4x8 matrix is outputted in a text file which is read by simulation. The output of ENN is scaled between [-45°, 45°] before applying to motors. The motors force Hips and Knees to move in either positive or negative direction to produce a certain distance from the origin. ODE writes out the distance covered (By the robot) with the given solution which ENN reads the fitness value. The fitness is compared and the parent matrix is modified to repeat the cycle. The simulation runs for 300 iterations.

4. The Analysis of Evolved Neural Controllers

There are twelve number of neurons scattered around 360° with an increment angle of 2π/n in Fig. 2. 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.

The direction, step size and the pattern of steps is depicted in Fig. 3. 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.

The graph in Fig. 4 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.

5. Conclusion and Future Works

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. 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 research was supported by the Original Technology Research Program for Brain Science through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (2010-0018950).

References

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

[2] S.-I. Lee and S.-B. Cho, "Observational emergence in evolutionary fuzzy robotics," *International Journal of Modelling, Identification and Control,* vol. 3, no. 1, pp. 88-96, 2008.

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

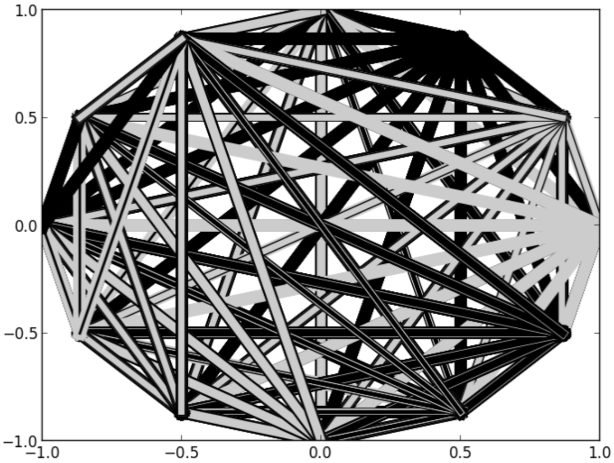

Fig. 2 Effect of synapse weight over the best neurons

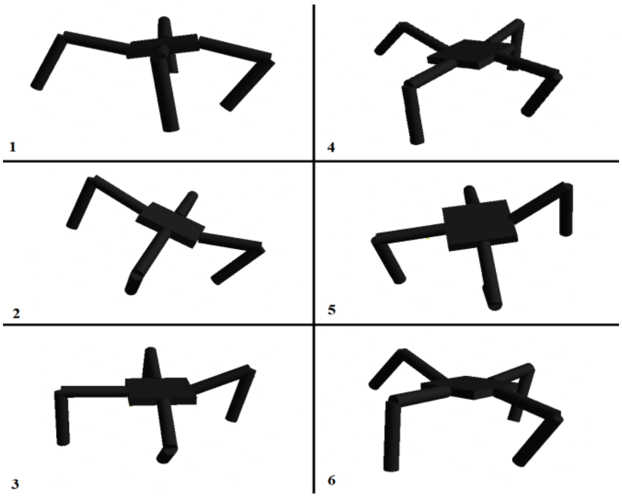

Fig. 3 The best quadruped robot movements

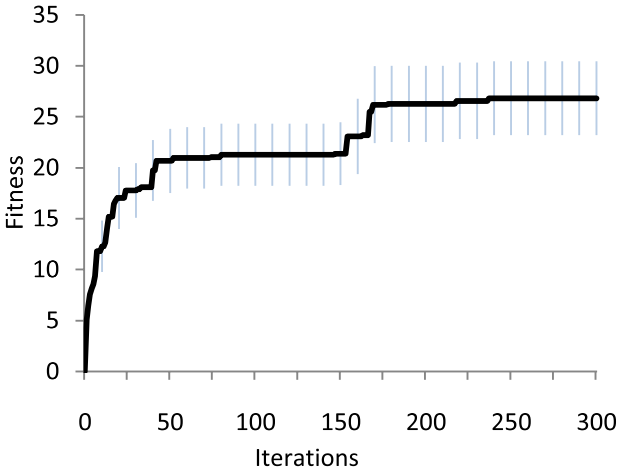

Fig. 4 Fitness of ENN over iterations