NEUROEVOLUTION BASED TIME/COST DISTRIBUTOR FOR REAL TIME SYSTEM CONTROLLERS

NABIL M. HEWAHI
Department of Computer Science
Islamic University Of Gaza- Palestine
nhewahi@mail.iugaza.edu

ABSTRACT. In this paper we present a theoretical model based on neuroevolution for systems that work under real time constraints. The proposed model is to decide the needed time based on the given situation and accordingly the necessary action. The proposed system has four main stages, the first one is to decide the time constraints based on the given environment surroundings, the second stage is to distribute the time/cost to determine the importance (priority) of each behavior based on the decided time by stage one. Stage three is to take the output of stage two to place an appropriate controller order which finally applied to the forth stage to recognize the final action of the system. It is shown in general how the proposed system can be applied on a soccer robot example.
Keywords: Neural networks, Genetic algorithms, Neuroevolution

1 INTRODUCTION

One of the interesting increasing research areas in the current years is the combination of genetic algorithms and neural networks. One of these combinations is the neuroevolutions. Neuroevolution is the artificial evolution of neural networks using genetic algorithms. Neuroevolution has proven very high capabilities in various applications and in reinforcement learning tasks [1-4,6-13]. In difficult real-world learning tasks such as controlling robots, playing games, or pursuing or evading an enemy, there are no direct targets that would specify correct actions for each situation. In such problems, optimal behavior must be learned by exploring different actions, and assigning credit for good decisions based on sparse reinforcement feedback. Comparing neuroevolution to the standard reinforcement learning, neuroevolution is often more robust against noisy and incomplete input, and allows continuous states and action naturally. Much of the research in neuroevolution is on control tasks such as pole balancing and mobile robot control [6][10][11][13]. Some other applications are related to industry controllers [2][9]. Many approaches for evolving recurrent neural network controllers have been tried [4][6][13]. These approaches are becoming widely used in the area of neuroevolution. The study of these approaches and their efficiency is beyond our research, but one of those approaches could be used in our proposed system in evolving neural networks. All these approaches use the same genetic operators to evolve new nets but in different ways. According to various experiments the method used effect seriously the final results of the developed system.

In most of the mobile robot experiments, robots are stable in the sense the absence of a control signal the robot will either stay in the current state or quickly converge to a nearby state. This makes the robot to mostly perform competently in the real world as long as its behavior is preserved qualitatively after transfer. This is not the case with many systems such as plants, aircraft and rockets that are inherently unstable. In such environments, the controller must continuously output a precise control signal to maintain system equilibrium and avoid failure. Therefore, controllers for unstable systems may be less amenable to techniques that have worked for transfer in robots.

2 RESEARCH OBJECTIVES

Designing controllers for various applications using neural networks or neuroevolution approaches have been explored (as mentioned above). These approaches have shown in general good performance. Because most of the controllers might work under the real time systems where the control action should be taken within time constraints, it is important to distribute the time limits among the controller inputs that affect its actions. The distribution of time occurs after deciding which input is more important than the other based on the current situation of the system. In all the previous related work, neural networks or neuroevolution worked as controllers neglected the time constrains. Some systems that might need such kind of time distributor systems are:

1. Robots that play soccer. At certain position(mostly), the robot has to know where to pass the ball very quickly (might not check all his surroundings), otherwise, one of his opponents might come and get the ball.
2. Automatic pilot in cases of emergency. A very fast response is required based on the situation or the plane might get crashed.
3. Games where players should do some action or otherwise destroyed by other player.

Another version of the same time distribution for controllers is the cost distributors. Cost distributors can be used in economic and commercial applications. It can also be used in information retrieval based on speed, memory and the size of the databases.

Deciding the necessary time (changeable) to perform the action which is changeable in real time applications, is one of the very challenging and not yet tackled problems. This problem is difficult to solve using neural networks alone because in many different situations the time needed and the action to be taken is changeable (very difficult to give examples), hence generalization is difficult. Genetic algorithms proved to be very efficient in optimization specially in difficult problems. In this paper we try to touch the problem posed in this research by presenting only a proposed general neuroevolution based system with which we try to put the points in the right directions and open a new research avenues to the researchers.

3 THE PROPOSED SYSTEM

The proposed system is a cycle consisting a sequence of operations. Each operation has a specific function. Figure 1 depicts the proposed system outline. The system consists of four main operations, Time /Cost Limit Decider (T/CLD), Time/Cost Distributor (T/CD), The controller, and the Industry or the system to be controlled. Below we shall explain each stage in details:

1. T/CLD : This is the first stage in the process, where the time/cost limit is decided based on the current status of the system (the current status of the environment). The input of the neural net is the environment inputs where the output is the decided time/cost. The net structure is almost n inputs to 1 output (we shall refer to such a structure with (n-1). This net can be developed based on backpropagation algorithm because enough examples can be provided. In many cases this net is not considered if the decided time/cost is provided.
2. T/CD : It is to decide the level of importance of the inputs of the controller. The output of T/CLD is the input of T/CD and the output of T/CD is the inputs for the controller but with consideration to their priority (importance). This net can be constructed using neuroevolution and the fitness value can be computed based on the comparison of the results obtained by our evolved net and the real results. The structure of the net is 1-n.
3. The controller : This can be implemented using neuroevolution approach or neural network using backpropagation (because we can provide examples). The inputs of the controller are the output of T/CD and its output is a set of possible actions and one or more actions (orders) will be considered. The net structure is n-m.
4. The System to be controlled: The net here plays a simulation role to the industry or the system to be controlled. The inputs of the simulation net comes from the output of the controller (not all) where the actions are the system's behavior based on the controller results.

4 THE SOCCER ROBOT EXAMPLE

In this section we shall show how the steps of the proposed system can be
Implemented on a soccer robot example. Soccer robot example is a robot which plays soccer with other robots, the robot should work quickly to move towards the ball to kick it before any other robot comes and kicks it. The kicking could be passing the ball to a partner or to score a goal. This example is chosen because real time behavior is significant. Below we shall explain in a brief description what is the inputs and outputs for each stage(neural net) in the proposed system.

1. T/CLD : The inputs for the neural net is the current state of the whole system's environment in regard to our robot which holds the ball, and the output is the time limits in which the action of the system should be taken. Figure 2 shows the T/CLD neural network for the robot example.

From Figure 2 we notice that in the input layer we have the current position of the robot holding the ball and all the surrounding positions (could be opponents or partners). The positions can be expressed as certain areas. The opponent can have 0 value, and 1 is the value for the partner. –1, a value that indicates an open area (no partner or opponent).

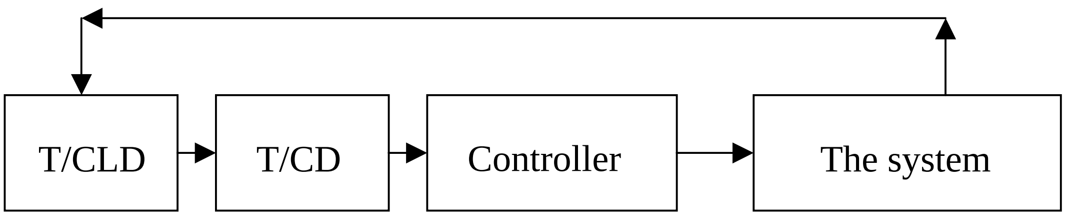

FIGURE 1. THE OUTLINE OF THE PROPOSED TIME/COST SYSTEM

2. T/CD : The T/CD in the case of soccer robot is a time distributor. The distributor is to decide the level of importance of the inputs of the controller. The input in T/CD neural network is the time limit obtained by T/CLD. Figure 3 shows the neural net of T/CD. It is to be noticed that the time needed for each operation for the robot such as look ahead, look left, look right, or look by some angle is known. According to the descending order of the output values which are above the threshold value, we specify the order of the operations. The operations to be performed within time limits by the controller is given value 1 and 0 for those which are not to be performed. To make the idea clear, let us consider the following example:

Assume we have 4 output operations and the output values for the operations are (expressed as operation number :output value)

Op1:0.5 Op2:0.7 Op3:0.6 Op4:0.8

We assume further that the time limits is 0.8, and the time for the operations as below(expressed as operation-time needed)

Op1-0.2 Op2-0.6 Op3-0.4 Op4-0.3

Based on the output values, the order of the importance of the operations is Op4, Op2, Op3 and Op1 respectively. This means that the system should do the operations Op4 and Op3 because the sum time is less than 0.8. Op2 is neglected despite it is the second in the importance of the operations because the summation of

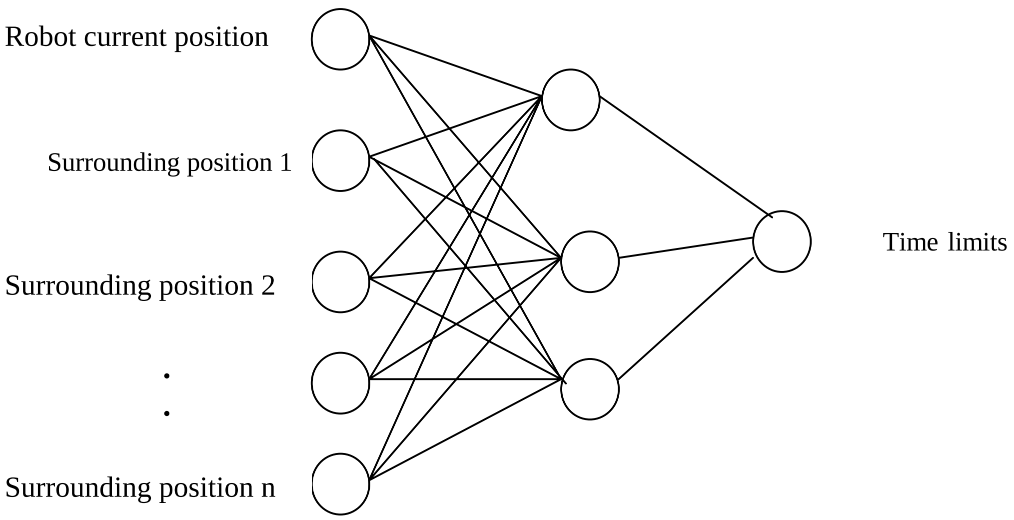

FIGURE 2. T/CLD NEURAL NETWORK BASED ON THE CURRENT STATUS OF THE ROBOT

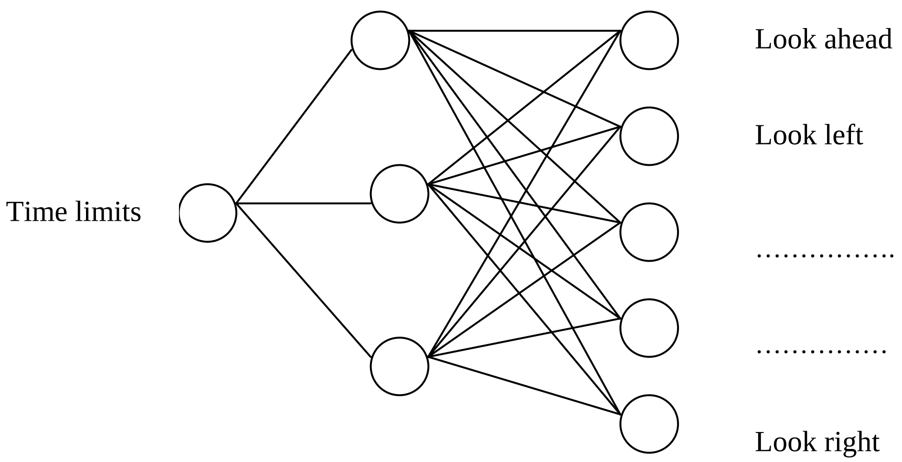

FIGURE 3. THE NEURAL NET OF T/CD OF THE SOCCER ROBOT . THIS IS 1-n NETWORK

Op4 plus Op2 is more than 0.8. This means the values for the controller inputs would be 0,0,1,1 for Op1, Op2,Op3 and Op4 respectively.

It is to be clear that for each time limit, we shall have several neural nets from which we finally choose the best. We finally will have one net for each time limit. In cases where time limit is not previously tested, the closest one can be chosen.

Genetic operators such as crossover and mutation are applied within each set corresponding to the same time limit. One simple crossover is exchanging weights between the nets and mutation is increasing or decreasing some weights chosen at random.

It is to be noted that as mentioned in section 3, if the time limit is provided, the T/CLD neural network will be neglected and the inputs of the supposed T/CLD plus time limits will be the inputs of the neural network of the T/CD. Figure 4 shows T/CD neural network in case T/CLD neural network is neglected.

3. The Controller : The input of the controller are the outputs of T/CD and its output is the set of possible actions. One or more inputs can be effective and that depends on the T/CD but only one output will be activated in our robot example which might be more than one in other applications. Figure 5 shows the neural net for the soccer robot. The actions obtained by the controller are the orders to correct/treat the current status in favor of our robot. Some of these orders might be as move left, pass the ball to your left partner or give back the ball to the goal keeper.

4. The System: This is a simple neural network not in its structure but in its behavior. The inputs of this net is the controller orders and the output are the real action of the chosen order. In our robot soccer one action is taken at a time which is a real interpretation of the controller order. The system simulation network is given in Figure 6. We notice that the inputs are orders and the outputs are actions in which if any is chosen the robot ‘s situation and the surrounding situations might change, hence another cycle is needed for a new situation.

It is to be noted that it is not necessary to have controller orders mimic the actions and the difference is only the act itself. It is also in many applications the number of inputs is not related at all to the number of actions which means we may have a n-m simulation neural network.

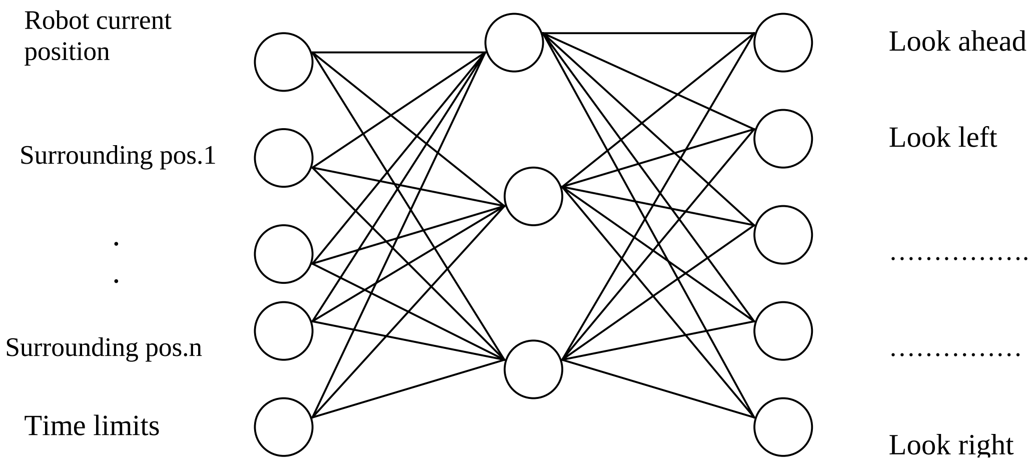

FIGURE 4. THE NEURAL NET OF T/CD WHEN T/CLD NEURAL NETWORK IS The NOT NEEDED. THIS IS n-m NETWORK.

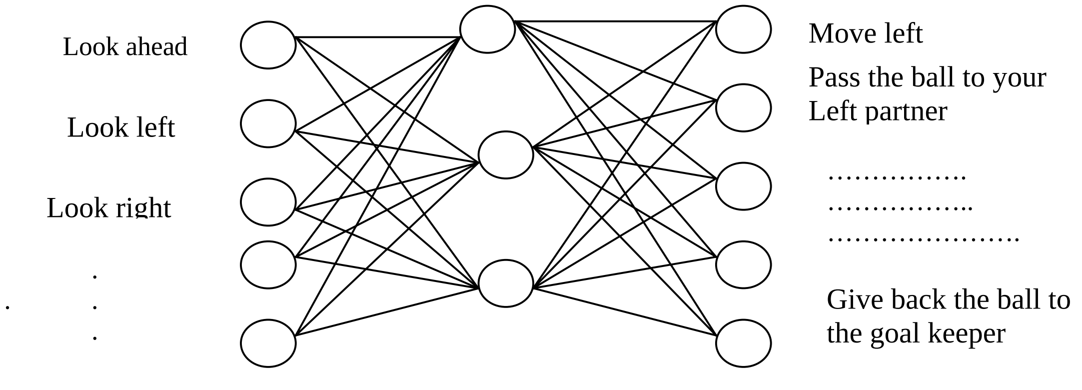

FIGURE 5. THE NEURAL NET FOR THE CONTROLLER ORDERS

This makes the simulation important and which might not appear here in our soccer robot example (except the act),but still the proposed system keeps the generality regardless of the application.

5 CONCLUSIONS

In this paper we presented and discussed a theoretical system that takes care of distributing time/cost on the related sensors to allow a real time system to respond on the bases of time constraints. Before distributing the time, the proposed system should know first the priority of each sensor so it can use the most important ones to give the proper results. This proposed system is based on learning process using neural networks and neuroevolution. Neuroevolution is used here in the TC/D which is the stage in the developed system that scales and finds out the importance and priority of each sensor. The reason behind using neuroevolution in this stage is the inability to perform this task with only neural networks or genetic algorithms. In complex systems, this stage is very difficult because no generality can be generated, moreover, the number of possible control cases could be too much. Fitness function of genetic operators plays here a significant role to direct the neuroevolution process. Various applications can get advantage of implementing such kind of our proposed system, these applications might be related to games, robot control, industry control and automatic pilot. Another direction of such systems applications could be in commerce, economy or information retrieval. This research is the first stage of a long project that should lead to systems that can tackle real time systems with considerations to time limits based on the surrounding system situation. Some of the future directions would be 1. Exploring other possible approaches to solve such kind of problems 2. Implementing the proposed system and apply it on various real life applications. 3. Developing a comparative study for the proposed system in stages where neural network or neuroevolution can be used.

REFERENCES

[1] A. Agogino, K.Stanley and R.Miikkulainen, Online interactive neuro-evolution, *Neural Processing Letters* 11(2000),pp.

[2] A.Conradie, Risto Miikkulainen and C.Aldrich, Adaptive control utilizing Neural swarming, *Proceedings of the 2002 Genetic&Evolutionary Computation Conference (GECC-2002*),(2002).

[3] G.Drake and J.Smith, Simulation system for real-time planning, scheduling and control, *Proceedings of 28th Conference on Winter Simulation* (1996),pp. 1083-1090.

[4] J.Fan,R.Lau and R.Miikkulainen, Utilizing domain knowledge in neuroevolution, *Proceedings of the Twentieth Inter. Conf. On Machine Learning(ICML-03*) Washington,Dc,(2003).

[5] R.Florian, Evolution of alternate object pushing in a simulated embodied agent: *Preliminary report, Center for Cognitive and Neural Studies (Coneural)*, Romania, August, 2004.

[6] F.Gomez , Robust non-linear control through neuroevolution, *Ph.D dissertation, The university of Texas at Austin,USA*, 2003.

[7] F.Gomez and R.Miikkulainen, Transfer of Neuroevolved Controllers in Unstable Domains, *Proceeding of The Genetic Evolutionary Computation Conference (GECCO 2004)*, 2004.

[8] F.Gomez and R.Miikulanien, Active Guidance for a Fitness Rocket using Neuroevolution, *Proceedings of Genetic Evolutionary Computation Conference (GECC-03)*,2003.

[9] N. Hewahi, Engineering Industry Controllers Using Neuroevolution, accepted for publications in the journal of AIDAM: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, USA.

[10] D.Law and R.Miikkulainen, Grounding Robotic Control with Genetic Neural Networks, (Technical Report AI-94-223). Department of Computer Sciences, The University of Texas at Austin (1994).

[11] S.Nolfi and D.Floreano, *Evolutionary Robotics*, MIT press, Cambridge, 2000.

[12] S.Nolfi and D.Parisi, Evoultion of Artificial Neural Networks, in M.A.Arbib(Ed.) *Handbook of Brain Theory and Neural Networks*, 2nd edition, Cambridge, MA: MIT press,,2002, pp. 418-421.

[13] K.Stanley and R.Miikkulainen, Efficient evolution of neural network topologies, *Proceedings of the 2002 Congress on Evolutionary Computation(CEC'02*) ,2002.

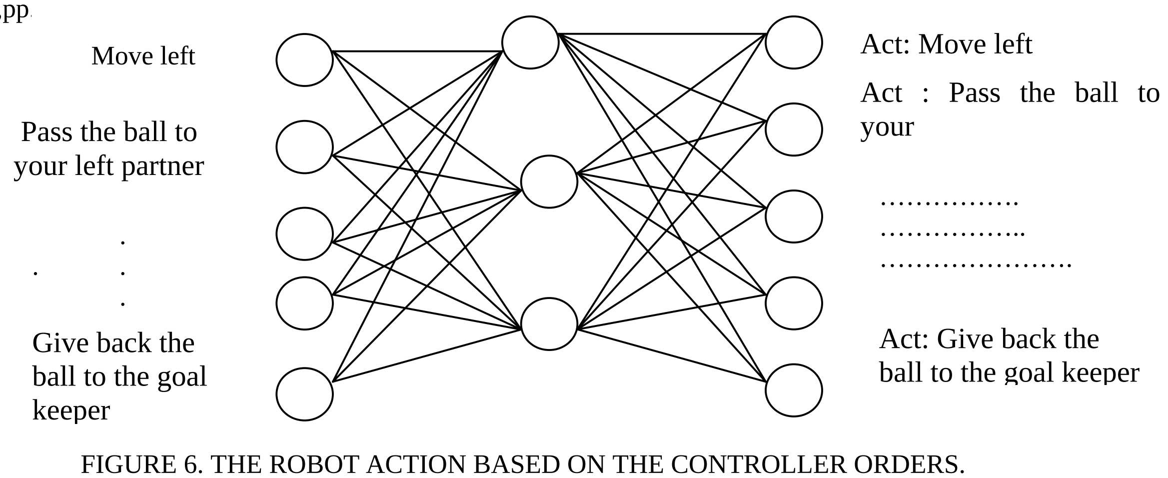

FIGURE 6. THE ROBOT ACTION BASED ON THE CONTROLLER ORDERS.