International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:12, 2010

Intelligent Dynamic Decision-making Model Using in Robot's Movement

Yufang Cheng*, Hsiu-Hua Yang
Graduate Institute of e-Learning,
National Changhua University of Education,
No.1, Jin-De Road,
Changhua City, Taiwan, R.O.C.

Abstract**—**This work develops a novel intelligent "model of dynamic decision-making" using cell assemblies network architecture in robot's movement. The "model of dynamic decision-making" simulates human decision-making, and follows commands to make the correct decisions. The cell assemblies approach consisting of fLIF neurons was used to implement tasks for finding targets and avoiding obstacles. Experimental results show that the cell assemblies approach of can be employed to efficiently complete finding targets and avoiding obstacles tasks and can simulate the human thinking and the mode of information transactions.

Keywords**—**Cell assemblies, fLIF, Hebbian learning rule

I. Introduction

CELL assemblies (CAs) were established to mimic the human thinking. This work develops the novel "model of dynamic decision-making" (DDM) using a cell assemblies network architecture. The "model of dynamic decision-making" simulated human decision-making, and followed commands to make the correct decisions.

The CAs were based on the Hebbian Learning Rule and extended the concept, which Hebbian [1] proposed in 1949. The concept that CAs are reverberating circuits of neurons, that form a network that can simulate human thinking. CAs has been applied to solve many complex problems. For example, Huyck and Orengo [2] applied this approach for information retrieval and classification. Their classification system correctly categorized the congresspeople's party 89% of the time. Osamu, Masayuki, Mei, and Kazuharu [3] applied CAs to classified the Japanese vowels, such that the system can recognize human voices. That system recognized learned vowel sounds but also vowel sounds the model heard for the first time. Alnajjar and Murase [4] applied this approach in mobile robots, such that the robots reached a target faster than when using a genetic algorithm. Huyck and Belavkin [5] encoded symbols by CAs after training using a compensatory learning rule. Training results indicated that experiments failed in the human also mistakes on. Huyck [6] used CAs to develop hierarchical categories. The advantage of overlapping CAs was that neurons could participate in more than one CA. The classification accuracy rate achieved was as high as 97%.

In this work, to train the system learning rules, the CAs approach consisting of fatiguing Leaky Integrate and Fire (fLIF) neurons to implement tasks in robot movement. This work has two tasks. The first task is to find targets that were human; the second task was to avoid obstacles, which were non-human objects.

H. Yang is with the Graduate Institute of e-Learning, National Changhua University of Education, corresponding author to provide phone: 00886-4-7232105; fax: +00886-4-7211290.

II. Cell assembly networks

Cell assemblies based on the theory proposed by Hebbian in 1949 and then extended the concept. Many CAs have been developed in recent years e.g. magnetic resonance imaging (MRI) and positron emission tomography (PET) technology.

A. Hebbian Learning

Hebbian proposed that when both pre-synaptic and post-synaptic neurons fired, the weight between the two neurons increases; conversely, when one neuron fires and the other does not, the weight decreases. Additionally, reverberating circuits consist of a group of neurons, such that the weight between neurons can be adjusted via the Hebbian learning rule. For example, neuron A fires neuron B, neuron B then fires neuron C, and neuron C in turn fires neuron A, such the weight strength between neurons increase markedly. The activities of this connection of neurons increase activation likelihood. This cycle is called a CA.

B. fLIF Neurons

Cell assemblies imitate human thinking and use of a series of neurons for storage. Sustained activation of these neurons can be stopped until the external environmental stimulation has been stopped. Cell assemblies are based on environmental input. When neurons received more activation than the energy threshold, the neuron fires, and the energy is spread through synapses to other neurons. Equation (1) describes the activation of neuron *i* at time *t* if it does not be fired at time *t*-1:

$$E_i(t) = E_i(t\text{-}1)\ /\ d + \sum W_{ji}\ ,\ d > 1 \qquad (1)$$

The current activation is at time *t*-1 divided by a decay factor d plus the sum of inputs weights of all neurons from neuron *j* to neuron *i*; the decay factor is a constant representing the energy leaky in the transfer process.

When fLIF neurons have fired, the neurons are fatigued. That is, the neurons need additional energy to fire again, whereas the neurons fired easily the next time if the neurons have not yet fired. Equation (2) indicates that the activation threshold θ increases if a neuron fired. Equation (3) indicates that if the neuron did not fire, threshold θ is reduced to a base resting level.

$$\theta_t = \theta_{t-1} + F_c \tag{2}$$

$$\theta_t = \theta_{t-1} - F_r \tag{3}$$

The goal of this work is to use the principles of CAs to develop a dynamic decision-making system that searches out targets and avoids obstacles. Thus, a novel DDM model was developed in this work.

III. Experimental Study

The DDM model adopted the CAs approach and was coded in JAVA. The "model of object recognition" (OR) was created as an adjuvant model. Fig. 1 shows the processes of two models (The model of OR, and the model of DDM). First, the image of surrounding scene was captured from the camera, and sent to the OR model for image analysis. Then, analytical data generated by the OR model are sent to the DDM model for generating commands based on correct decisions. Each model is discussed as follows.

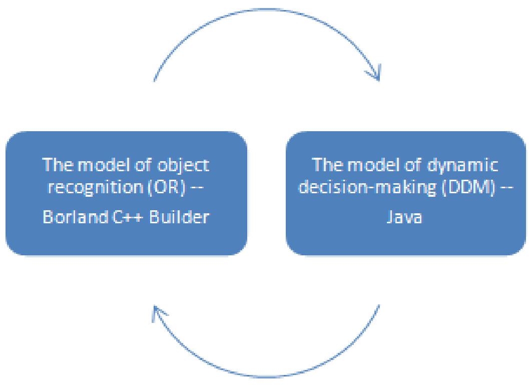
The model of object recognition (OR) -- Borland C++ Builder
The model of dynamic decision-making (DDM) -- Java

Fig. 1 The process of the OR model and the DDM model

A. The Model of Dynamic Decision-making (DDM)

This DDM model was using the logical rules for training (Fig. 2). Each CA network is represented as a rectangle, each rectangle is composed of some fLIF neurons. The logic rules can be divided into three modules. The input module is used to represent the location of each obstacle in the images. The icon in the input module below simulates the location at which human eyes are looking. The network in the input module is similar to brain's working memory; it is a temporary network. The fact module simulates the turning factors. The fact module simulates experiences after learning that is stored in the brain's long-term memory as a permanent network. The action module receives output from the fact module, and generates appropriate instructions; action module is also a temporary network.

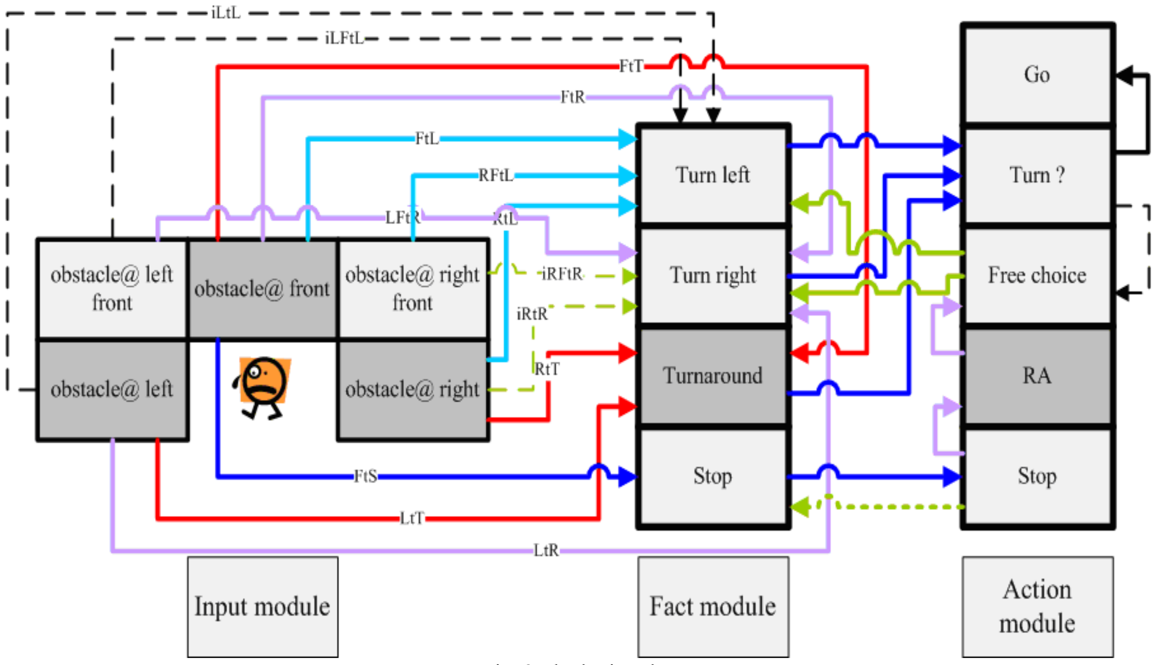

Fig. 2 The logic rules

The solid line in Fig. 2 represents the excitatory stimulus, such as from “front” of the input module to the “turnaround” of fact module (FtT). The dotted line represents the inhibition stimulus, such as the “left” of the input module connected to the “Turn left” of the fact module (iLtL), and RA is a resistant network. The initial value is a negative number. When energy spreads through the RA, it will delay some running steps based on its threshold, initial energy and input energy. This work utilized the positions of obstacles with the logic rules to make decision commands [7]. The Fig. 2 is utilized for avoiding obstacles; however, the stimulus to find a target (human) is opposite that for avoiding obstacles, which aim is go to the position of target.

Equation (4) describes activation of network *n* at time *t*, where network *m* (input) is a temporary network, network *n* is a temporary or permanent network, E_{n0} is the initial energy value of network *n*, W_{mn} is the weights between network *m* connected to network *n*, and d is the decay constant. When $E_n(t)$ is greater than or equal the threshold value, network n will fire.

$$E_n(t) = E_{n0} + \sum W_{mn} \times \left(d(1-\frac{1}{d})^t \;/\; (d-1) \right),\; d > 1 \qquad (4)$$

If the input network is a permanent network, activation will continue to deliver the energy and will not disappear; for example, the fact module transmits the activation to the action module. Activation of acceptance network *n* at time *t* is calculated by (5), where *m* is the input network, and *n* is the acceptance network.

$$E_n(t) = E_{n0} + \sum W_{mn} \times t \qquad (5)$$

B. The Model of Object Recognition (OR)

The captured images from the camera was processed in the OR model that ascertains object type, its coordinates, and size, which are then sent to the DDM model. The DDM model then generates a command to find the objectives and avoid obstacles. The OR model was coded in Borland C++ builder.

C. Hardware and Software

The system CPU is a 1.86GHz Intel Core2 Duo E6320 with 1GB of memory and a 160GB hard drive. The operating system is Microsoft Windows XP professional.

IV. Experimental Results

A. Experimental Scene Design

In the experimental scene was designed to use the artificial objects, which contains artificial objects of human, house, car, tree, rectangle, and sphere. Fig. 3 and Fig. 4 is the designed experimental environment, and red circles are targets (the human with different colors of skin). The amount of the targets has 3 human objects. This experiment intended to find out the best path of robot’s movement for the targets-finding and obstacles-avoidance.

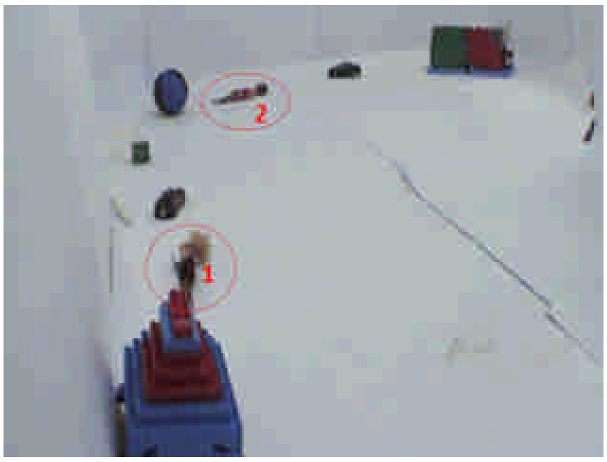

Fig. 3 Experimental environment (a)

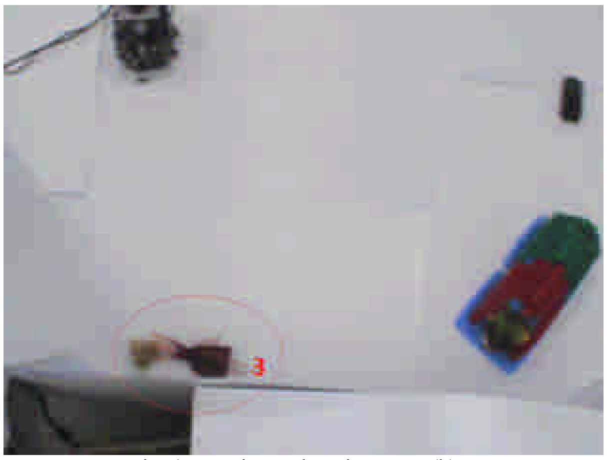

Fig. 4 Experimental environment (b)

B. Results

Fig. 5 presents the one of the path for the movement of robot’s targets-finding and obstacles-avoidance. The basic results showed that the first target of human was more difficult to be identified, because of the angle of view in the first target of human is different, for example, the shape of the human body in the captured image was affected by the shadowing of its hair after image-processing. The robot can not to identify the type of human object. Then the second and third targets of human object are more easily to be identified than first target. Because the skin color of the second target has dark skin and the shape of third target of human body did not be affect by its hair’s shadow.

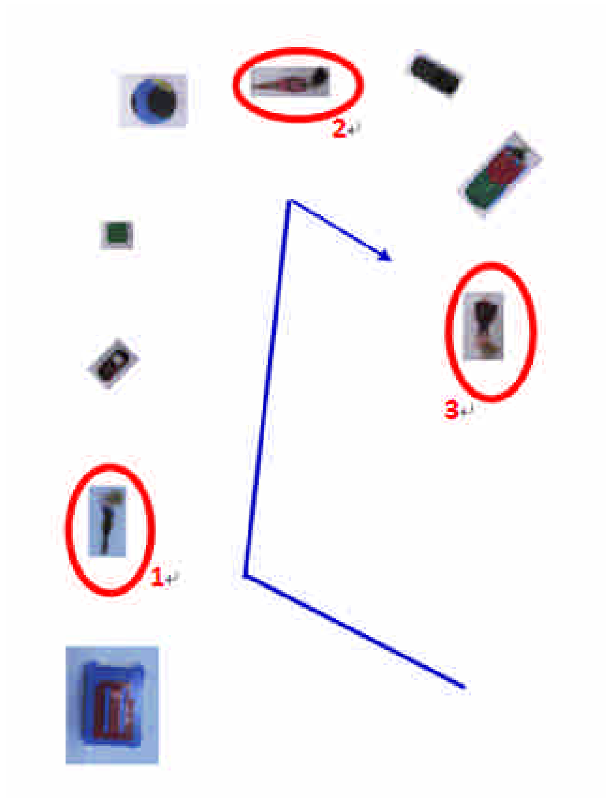

Fig. 5 The path-movement of robot

V. Discussion and Conclusion

In this work, the DDM model used the CAs approach effectively. This model can simulate human thinking and information processing, and issue correct commands (e.g. turn left and turn right) to effectively find targets and avoid obstacles.

However, the potential problems require improve. First, the OR model requires lots of computing memory to process images for analysis, so that it affects the computing speed becoming slowing. This was required to increase the hardware and optimize the function of OR model and DDM model. Second, finding an object depends on identifying the object during image-processing. Thus, the recognition rate of the OR model requires improvement.

References

[1] D. O. Hebb, *Organization of behavior*, New York: Wiley, 1949, pp. 46–56.

[2] C. Huyck, and V. Orengo, "Information retrieval and categorisation using a cell assembly network," *Neural Computing and Applications*, vol. 14, no. 4, pp. 282–289, 2005.

[3] H. Hosamu, M. Miyamoto, M. Zheng, and K. Kuroiwa, "A neural network model for encoding and perception of vowel sounds," *Neurocomputing*, vol. 44–46, pp. 435–442, 2002.

[4] F. Alnajjar, and K. Murase, "Self-organization of spiking neural network that generates autonomous behavior in a real mobile robot," *International Journal of Neural Systems*, vol. 16, no. 4, pp. 229–239, 2006.

[5] C. Huyck, and R. Belavkin, "Counting with neurons: rule application with nets of fatiguing leaky integrate and fire neurons," in *Proceedings of the 7th International Conference on Cognitive Modeling*, Trieste, Italy, 2006, pp. 142–147.

[6] C. Huyck, "Creating hierarchical categories using cell assemblies," *Connection Science*, vol. 19, no. 1, pp. 1–24, 2007.

[7] Y. Fan, and C. Huyck, "Implementation of finite state automata using fLIF neurons," in *IEEE SMC UK & RI 6th Conference on Cybernetic Systems*, London, UK, 2008, pp. 1–5.