Development of Behavioral-Emotional Selection System based on Self-Organizing Map (SOM) and Markovian Model

自己組織化マップ(SOM)とマルコフモデルに基づいた行動と感情の選択システムの開発

WISANU JITVIRIYA

Table of Contents

Page

Page

Page

Page

Chapter 1

Introduction

1.1 Background

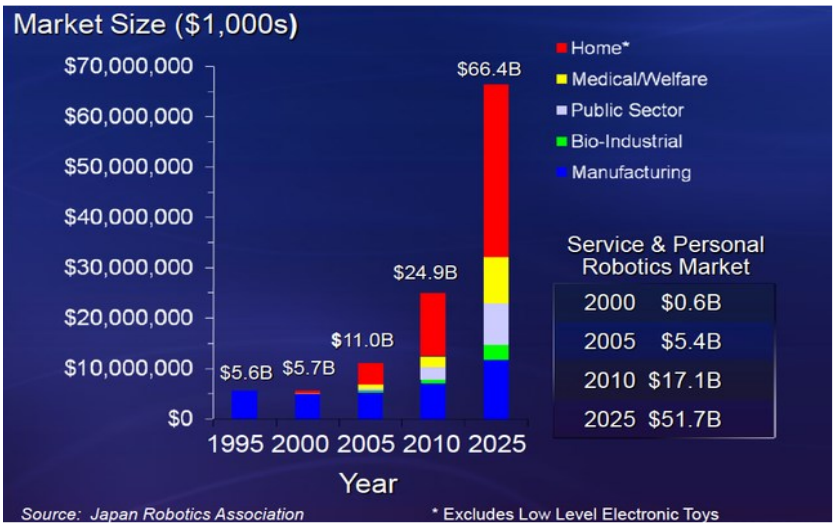


Fig. 1-1 Worldwide Robotics Market Growth

Nowadays, the rapid development of high technology has produced robots not only for industrial factories (industrial robots) but also for museums, homes, healthcare institutions and so on (non-industrial robots). There are various types of non-industrial robots such as service robots, welfare robots, therapeutic robots and domestic robots. In Fig. 1-1 shows the worldwide robotics market growth, this information has been gleaned from the Japan Robotics Association. The market of service and personal robots is expected to grow increasingly in the future [1]. These robots are designed with artificial intelligence (AI) to improve the robotics system is to have them imitate human thinking and behavior. Human-robot interaction (HRI) plays an important role in modern autonomous robots. HRI requires that robots not only passively receive information from the environment, but also can make appropriate decisions and actively change in varying environments, thus functioning more autonomously and intelligently [2] - [5]. However, Designing robots are able

to interact with human beings that is still a huge challenge, for example, demonstrates cognition in a complex environment, enable actions to be selected autonomously, or models emotional expression and smooth communication. McCarthy's research has described the essential characteristics of robots must have a consciousness, introspective knowledge, and some philosophy to perform in the common-sense world and to accomplish tasks effectively [6].

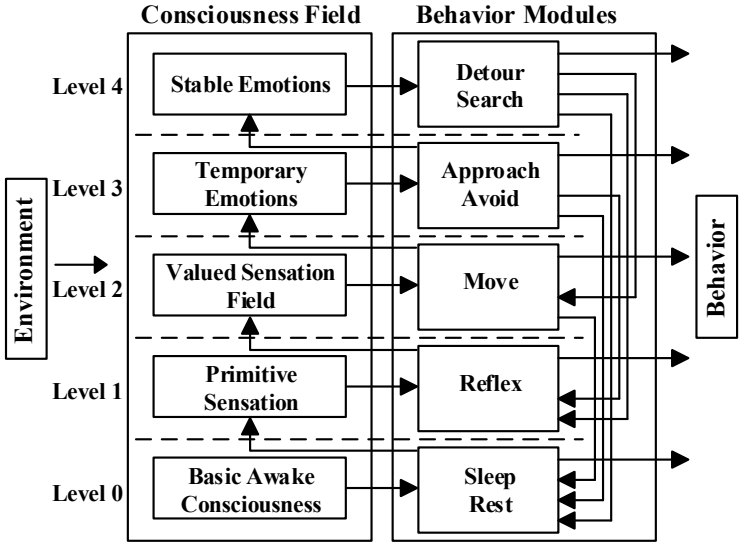


Fig. 1-2 Consciousness-Based Architecture (CBA) model

Therefore, Hayashi laboratory's works have focused on studying and investigating the application of brain-inspired technology by developing the robots with consciousness resembling that of animals and human beings [7] - [9]. Consciousness-Based Architecture (CBA) model (Fig. 1-2) is constructed to define the relationship between the consciousness field and behavior modules based on Tran Duc Thao's research [10]. Also, the CBA model introduces an evaluation function of action selection to realize a practical robot which is able to perform its action and adapt its environment. However, if the robot has only the behavior selection system which reflects cognitive information, it is limited capability and generating more human-like action [11]. Therefore, an artificial emotional system collaborates with an autonomous action selection system that has an important role in an intelligent system, decision-making process, perceptions memory and more

[12]. There are many studies in the literature focused on the concepts of emotion model for developing autonomous systems in various fields [13], [14]. Emotional expression skills are necessary in enabling friendly and understanding between human beings and robots [15] – [17].

1.2 Problem statement

As mentioned in the background. The developers of non-industrial robots have met with some problems in determining the motion strategies and the action selection. For instance, if the robot's movements have the same motion or repeated, the robot is a loss of interest to users. Although this problem was solved by the conventional model as Consciousness-Based Architecture (CBA) model, but the CBA model has limitation in managing and selecting the robot's behavior that only depends on the increase and decrease of the motivation levels. For a truly effective system, the robots should not only select the behavior, but also express the appropriate emotional state according to changing external stimuli. Moreover, if the robot can perform a behavior and express emotion how to define the relationship between the behavioral selection and emotional expression models. Consequently, all problems will be considered and solved by the proposed system in this thesis.

1.3 Research purpose

The general objective of this research project is to create the behavioral - emotional expression system for Conbe-I robot to enhance intelligent behavior and emotions, and to facilitate communication between users and robot. We seek to increase the robot's behavioral-emotional intelligence capabilities so that the robot can distinguish, adapt and react to changes in the environment. After reviewing the literature of previous researches done about the behavior selection, and emotional expression models based on an artificial intelligence neural network, the specific objectives are established as follows:

1) To accurately classify all behavior and emotions of robot and generate the behavior and emotion maps based on an unsupervised learning as a Self-Organizing Map method.
2) To create an emotional expression model, that can continuously express emotional state based on updating the affective factors and the previous emotional state.
3) To define and design the relationship between behavior selection system and emotional expression model.
4) To verify the behavioral-emotional selection model with a conscious behavior robot (Conbe-I), and confirm the effectiveness of the proposed system with experimental results in a realistic environment.

1.4 Overview of the thesis

The thesis consists of five chapters covering the background history, research objectives, modeling of the behavioral-emotional expression selection system, discussion of the results and conclusions.

Chapter 2 describes the fundamental theories for Conbe-I robot system, including a Self-Organizing Map (SOM) learning, Markov theory and Kinematics modeling, which are used to construct and develop the behavioral-emotional selection system.

Chapter 3 shows the system structure of Conbe-I robot, it is divided as the system configuration and overview of the software. At the end of this chapter, the computation of robotic arm posture is explained to perform the sequential movement.

Chapter 4, the overview of the proposed system is thoroughly described that consists of three major processes. The first process, the robot recognizes the external situation and generates the robot's motivation. In the second process, the cognitive process is used for clustering the input stimuli (the visual information and the internal motivation of the robot), which analyzes based on an unsupervised learning, then the affective factors and behavioral factors are calculated. The last

process is a behavioral-emotional expression that is modeled based on the Markov theory, the probabilities of emotional and behavioral state transitions are updated with affective and behavioral factors. And then, Conbe-I robot has been implemented to show the effectiveness of the proposed system.

At the end, chapter 5 is a conclusion that summarizes the thesis, and suggestion the future works, which has been mentioned in order to develop and improve the behavioral-emotional selection system in the future.

Chapter 2

Fundamental Theories for Conbe-I robot System

Nowadays, the new generation control architectures of autonomous robot systems have been contrived to be inspired from cognitive mechanism of the human brain. One of the main problems of autonomous robots is how to develop an intelligent system with a learning capability to acquire both varieties of knowledge and behavior through the interaction between humans and robots. Therefore, this chapter is proposed to explain the fundamental theories such as a Self-Organizing Map, Markov theory and Kinematics modeling, which are used to construct the behavioral-emotional selection system for the robot. Each fundamental theory can be described as follows.

2.1 Self-Organizing Map (SOM)

2.1.1 Introduction of SOM

Artificial Neural Networks (ANNs) have been widely studied and used to model the information processing systems based on inspired by biological neural networks. The neural network methods not only can provide solutions with improved performance when compared with traditional problem-solving methods, but also present an understanding of cognitive abilities of human. The architecture of neural networks and signal processing is used for modeling nervous systems can be classified into three categories. Feedforward networks transform the sets of input signals into the sets of output signals [18]. The objective of transformation is usually determined the supervised adjustment of the system parameters (Supervised learning) by a direction comparison between the actual and desired outputs. In the reinforcement learning [19], the input information defines as the states and rewards, this method can learn and relearn based on the actions and the effect (rewards). The last category, the neighboring cells in a neural network compete in their

activities, and develop adaptively into specific detectors of different signal patterns. In this category learning is called competitive learning, unsupervised learning, or self-organizing learning [20].

2.1.2 Background of SOM

There are many the neural network architectures and learning algorithms, but one of the most popular neural network models is a Kohonen's Self-Organizing Map (SOM) [21]. A Self-Organizing Map (SOM) is a conceptual mathematic model of topographic mapping from the visual information to the cerebral cortex. Modeling and analyzing of the mapping are important in understanding how the brain perceives, encodes, recognizes and processes the patterns. The SOM is able to map a high-dimensional signal onto a lower-dimensional network that implements a characteristic based on a nonlinear projection. This section will look into the evolution of relevant biological models, from two fundamental models involved in Hebbian learning and Von Malsburg & Willshaw's Self-Organization model to Kohonen's Self-Organizing Map.

2.1.2.1 Hebbian learning

Hebbian theory is a primary theory in the neurosciences that explains about the adoption of neurons in the brain during the learning process. This theory describes a basic mechanism for synaptic plasticity, or Hebbian learning. In 1949, a Canadian research named Donald Hebb, who proposed a mechanism by which neurons adapt the strength of their connections to other neurons. The rule [22] is:

"*When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.*"

Hebbian learning is one of the concepts, which is used for unsupervised learning, that implemented in neural network models through changes in the strength of connection weights

between units. The weight value between two neurons increases if the two neurons activate simultaneously. On the contrary, the weight value will be decreased if they activate separately. Neurons tend to be either both positive or both negative at the same time that have positive weights, while those that tend to be opposite have negative weights. Through such Hebbian learning, weights become to reflect statistical regularities in the environment, with networks self-organizing so that different units learn to represent different environmental regularities. Suppose there is a network of nodes N_i, each connected to other nodes and the strength of the connection between node *i* and node *j* is $w_{ij,Hebbian}$. The mathematical basis for Hebbian learning is described below:

$$y_i = \sum_j w_{ij,Hebbian} x_j \tag{2-1}$$

$$\Delta w_{ij,Hebbian} = \eta_{Hebbian} x_j y_i \tag{2-2}$$

Equation (2-1) shows the state of output nodes as y_i, that is the output from neuron *i*, x_j is equal to the input from neuron *j* (for all neurons *j* connected to *i*), multiplies by the weight of the connection between neuron *i* and neuron *j*. And Hebbian's rule is often generalized as Equation (2-2), where, $\Delta w_{ij,Hebbian}$ denotes the change in weight from node *i* to node *j*, y_i (post-synaptic) and x_j (pre-synaptic) denote the activation levels of the node *i* and *j* respectively, and $\eta_{Hebbian}$ denotes the learning rate. The learning rate is usually a small number ($0 < \eta_{Hebbian} < 1$) that can be decreased through time. If Equation (2-1) is substituted into Equation (2-2) then the result is expressed as Equation (2-3).

$$\Delta w_{ij,Hebbian} = \eta_{Hebbian} x_j \cdot \sum_k w_{ik,Hebbian} x_k \;=\; \eta_{Hebbian} \sum_k w_{ik,Hebbian} x_k x_j \qquad (2\text{-}3)$$

This sample rule is not enough, because it is unstable; repeated use can increase the weights of the connections without bounds, and the performance will degrade since all the neurons will be saturated to their maximum values. Due to the fact that is the positive feedback: the larger weights will result in a large output, which will result in a larger increase of weights. It's still difficult to develop in the nervous system since there is a limit on the number and efficiency of synapses per neuron.

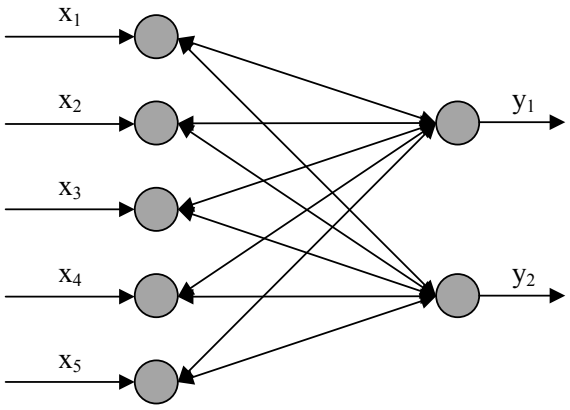


Fig. 2-1 The sample Hebbian network

Oja and other researchers [23], [24] proposed and developed the rules that have a decay time, which can be implemented using the negative feedback. The resulting network is shown in Fig. 2-1. The firing one time cycle is fed back to the next step, it can be expressed by Equation (2-4). And Equation (2-5) is represented the modified equation for updating the weight, where $x_j(t)$ and $x_j(t+1)$ differentiates between activation at time t and $t+1$.

$$x_j(t+1) \;=\; x_j(t) - \sum_{k=1}^{M} w_{kj,Hebbian} y_k \qquad (2\text{-}4)$$

$$\Delta w_{ij,Hebbian} = \eta_{Hebbian}(t) y_i x_j(t+1) = \eta_{Hebbian}(t) y_i \left(x_j(t) - \sum_{k=1}^{M} w_{kj,Hebbian} y_k \right) \quad (2\text{-}5)$$

The resulting of Oja learning algorithm is a so-call "Principal Component Analysis: PCA", which learns to extract the most variant directions among the data set [25].

2.1.2.2 Von Malsburg & Willshaw's Self-Organization model

The self-organizing learning behavior of brains has been studied for long time by many researchers such as: Hebb's learning, Marr's theory of the cerebellar cortex [26], Willshaw, Buneman and Longnet-Higgins's non-holographic associative memory [27], Von der Malsburg and Willshaw's self-organizing model of the retina-cortex mapping [28], [29], Amari's mathematical analysis of self-organization in the cortex [30] and Kohonen's self-organizing map. But Von der Malsburg and Willshaw first developed, in the mathematical forms by the self-organizing topographical mapping from pre-synaptic sheet (two-dimensional layer) to post-synaptic sheet (two-

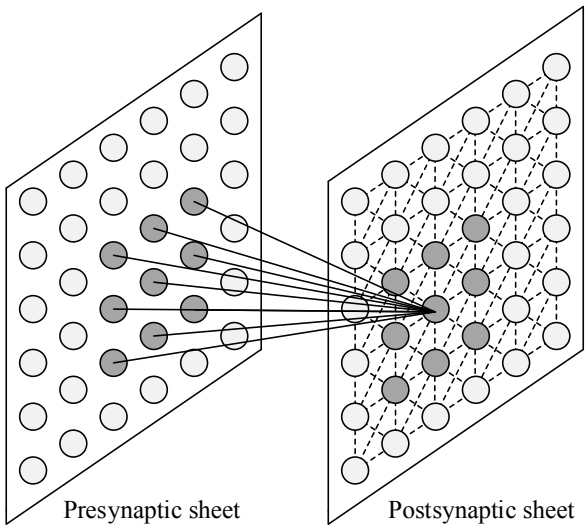


Fig. 2-2 Von der Malsburg's self-organizing model

dimensional layer) based on competitive learning. The competitive learning refers to the fact that will increase in strength to one neuron results in a relative and decrease in another, and this competition results in neurons that respond to the correlated inputs. If two inputs are activated by correlated activity, then they mutually reinforce their connections since they both work together to activate the target cell. Fig. 2-2 shows Von der Malsburg's self-organizing model and their basic idea was described as:

"*... The geometrical proximity of pre-synaptic cells is coded in the form of correlations in the electrical activity. These correlations can be used in post-synaptic sheet to recognize axons of neighboring pre-synaptic cells and to connect them to neighboring post-synaptic cells, hence producing a continuous mapping...* "

This model was the first to produce this pattern, and the first to use local connectivity that had short range excitatory connections and long range inhibitory connections in the sheet. The post-synaptic activities at time t, can be expressed by the nonlinear differential equation as shown in Equation (2-6), where, c is the decay constant. $w_{ij,Von}(t)$ is the synaptic weight between cell i and cell j in pre-synaptic and post-synaptic sheets respectively. The state of the pre-synaptic cell is $x_i(t); i = 1,2,...,N_x$, $x_i(t) = 1$; if cell i is active, $x_i(t) = 0$; otherwise. e_{ik} and b_{ik} are excitation and inhibition constants from other lateral cells. $y_k^*(t)$ is an active cell in post-synaptic sheet at time t. For the cells of post-synaptic fire of their activity by the threshold function as shown in Equation (2-7).

$$\frac{\partial y_j(t)}{\partial t} + c y_j(t) = \sum_j w_{ij,Von}(t) x_i(t) + \sum_k e_{ik} y_k^*(t) - \sum_{k'} b_{ik'} y_{k'}^*(t) \qquad (2\text{-}6)$$

$$y_j^*(t) = \begin{cases} y_j^*(t) - \theta, & if\ y_j^*(t) > 0 \\ 0 & otherwise \end{cases} \tag{2-7}$$

The modification of synaptic weights between pre-synaptic and post-synaptic sheets is determined by the Hebbian learning as expressed in Equation (2-8).

$$\frac{\partial w_{ij,Von}(t)}{\partial t} = \eta_{Hebbian} x_i(t) y_j^*(t) \tag{2-8}$$

2.1.2.3 Kohonen's Self-Organizing Map

Kohonen improved above self-organizing learning rule and proposed a simplified learning mechanism which incorporates the Hebbian learning and lateral interconnection rules. It is simplified and generalized model of the above self-organization process as explained in [31]:

"*...Kohonen's model of self-organizing maps represented an important abstraction of the earlier model of von der Malsburg and Willshaw; the model combines biological plausibility with proven applicability in a board range of difficult data processing and optimization problems...* "

Development of self-organizing maps is motivated by a distinct feature of the human brain that has been shown to be biologically plausible: The brain is organized in many places in primary sensory areas, are ordered according to some features dimensions of sensory signals such as tactile [32], visual information [33], [34] and acoustic [35]. One important point that emerges from the brief discussion of computerized maps in the brain is the principle of topographic map formation, which is commented by Kohonen as:

"*…The spatial location of an output neuron in a topographic map corresponds to a particular domain or feature of data drawn from the input space…*"

The principle has provided the neurobiological motivation for two basically different features-mapping models as von der Malsburg and Willshaw's model and Kohonen's model. The models differ from each other in the manner in which the input patterns are specified. The Kohonen model has received much more attention in the research than the von der Malsburg and Willshaw model. Because, integrating the neural field dynamics is usually every time consuming, and the replacement of the neural field representations with tuning curves reduces the computational burden dramatically. Moreover, topographic organizations are easily observable when the plotting centers of tuning curves.

Self-Organizing Map (SOM) also called Kohonen map or Kohonen network [36], [37]. The SOM is a neural network model and algorithm that implements a characteristic non-linear projection form the high-dimensional space of input signals onto a low-dimensional array of neurons. Kohonen network's properties both make them extremely useful in the visualization and exploration of data properties, and preserve the topology of input space during mapping. In Kohonen's model uses soft competitive learning in a post-synaptic like artificial neural network, the cells on the map become specifically tuned to various input signal patterns of classes of patterns through an unsupervised learning process. Fig. 2-3 shows a diagram of two-dimensional lattice of neurons commonly used. Individual neuron (output layer) is fully connected to all nodes in the input layer. The feed-forward structure is represented as the two-dimensional map with the neurons arranged into rows and columns.

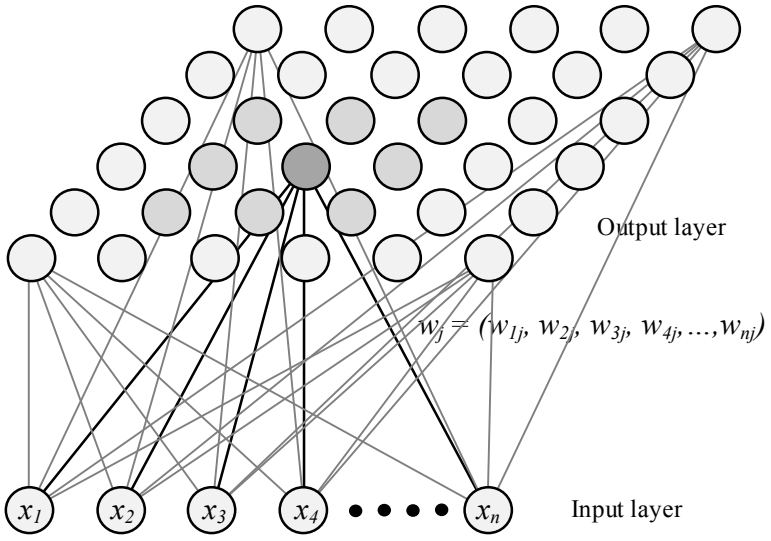


Fig. 2-3 Kohonen's Self-Organizing Map

2.1.3 The SOM algorithm

An algorithm for the formation of the self-organizing map proceeds first by initializing the synaptic weights in the network, which define them as small values and selected from a random number generator. Briefly, that has three essential processes involved in the formation of the self-organizing map as summarized below:

Competitive process: the neurons in each input pattern compute their particular values of the discriminant function, which performs for competition among the neurons of input patterns. The particular neuron with the largest value of the discriminant function is defined as the winner of the individual sample (input) pattern.

Cooperative process: the spatial location of a topological neighborhood of excited neurons is calculated by the winning neuron in individual group patterns.

Adaptation of synaptic weight process: the last process improves the excited neurons to increase their individual values of the discriminant function in relation to the input patterns by adjustment to their synaptic weights.

The descriptions of the processes of competition, cooperation and adaptation of synaptic weight processes are described below.

2.1.3.1 Competitive process

Assume the set of input variables $\{\bar{x}\}$ is selected from the input space, which is denoted by Equation (2-9), where, m denote the dimension of the input space.

$$\bar{x} = \left[x_1, x_2, x_3, ..., x_m\right]^T \tag{2-9}$$

Synaptic weight or weight vector of each neuron in the network has the same dimensions as the input space. Let the synaptic weight on cell j is defined by Equation (2-10), where, n is the total number of neurons in the network.

$$\bar{w}_j = \left[w_{j1}, w_{j2}, w_{j3}, ..., w_{jm}\right]^T, \quad j = 1,2,3,...,n \tag{2-10}$$

The simplest analytical measurement for matching $\bar{x}$ with $\bar{w}_j$ is the inner product as $\bar{w}_j^T \cdot \bar{x}$, based on maximizing inner product is mathematically equivalent to minimizing the Euclidean distance between vectors $\bar{x}$ and $\bar{w}_j$. Index c denotes the neuron that is the best matching input vector also called "the best matching unit or the winning neuron", which is able to determine by Equation (2-11).

$$c = \arg\min_j \left\|x - w_j\right\|, \quad j = 1,2,3,...,n \tag{2-11}$$

2.1.3.2 Cooperative process

This section explains the adjustment of the winning neuron, which locates at the center of a topological neighborhood and the lateral interaction among a set of excited neurons. In particular, a neuron is firing tends to excite the neurons in its immediate neighborhood more than those farther away from it. This observation, it makes the topological neighborhood around the winning neuron decay smoothly with lateral distance. In order to observe the model, that is suitable for the cooperative process. The topological neighborhood and the lateral distance are assumed, such that satisfies in two distinct requirements as:

- The topological neighborhood is symmetric about the maximum point, which is defined by the lateral distance is zero. Therefore, that point represents the maximum value at the winning neuron.

- The amplitude of the topological neighborhood decreases monotonically with increasing lateral distance, and decaying to zero when the lateral distance approaches to positive infinity.

The typical choice of the topological neighborhood that satisfies these requirements is the Gaussian function as expressed in Equation (2-12).

$$h_{j,c}(t) = \exp\left[-\left\| r_j - r_c \right\|^2 \Big/ 2\sigma(t)^2 \right], \tag{2-12}$$

where, $h_{j,c}(t)$ denotes the topological neighborhood centered on winning neuron c, and encompassing a set of excited neurons, a typical one of which is denoted by neuron j. $\left\| r_j - r_c \right\|$ is the lateral distance between winning neuron c and excited neuron j. r_j and r_c define the position of excited neuron *j* and the discrete position of winning neuron *c* respectively. Both positions are

measured in the discrete output layer. $\sigma(t)$ is called the neighborhood radius, this parameter is the effective width of the topological neighborhood as shown in Fig. 2-4.

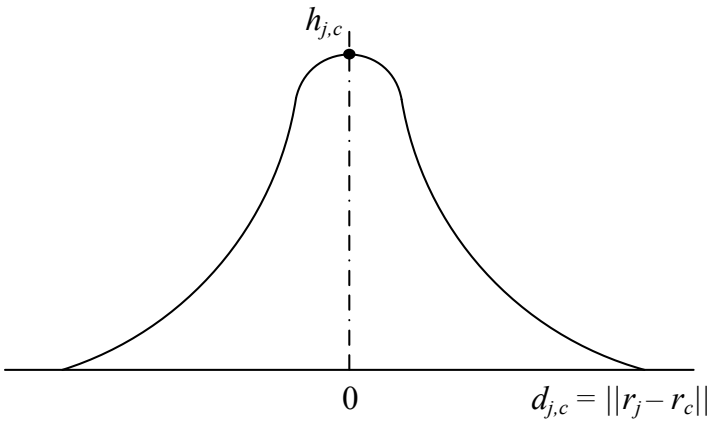


Fig. 2-4 The neighborhood function

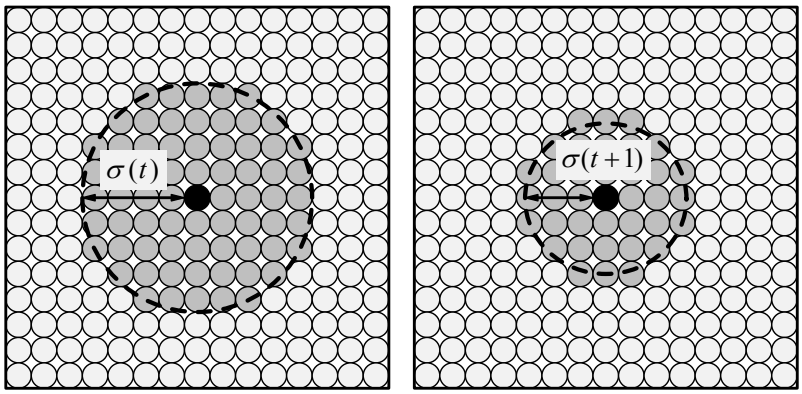


Fig. 2-5 The size of the topological neighborhood when it shrinks over time

The Gaussian topological neighborhood equation is more biologically appropriate than a rectangular one, and it also makes the SOM algorithm converge more quickly than a rectangular topological neighborhood function.

Another unique feature of the SOM algorithm is the size of the topological neighborhood that shrinks over time as displayed in Fig. 2-5. It can solve by the exponential decay function [38] as represented by Equation (2-13).

$$\sigma(t) = \sigma_0 \exp\left[{}^{-t}\!/_{\lambda}\right] \tag{2-13}$$

σ_0 denotes the width or radius of the topological neighborhood function at the initiation of the SOM algorithm. For a good global ordering, σ_0 should initially be large (up to half the size of the output space). t is the learning step and λ is a time constant. The value of time constant depends on the number of iterations (*T*), which estimates by Equation (2-14).

$$\lambda = {}^{T}\!/_{\log_e(\sigma_0)} \tag{2-14}$$

2.1.3.3 Adaptation of synaptic weights process

In the self-organized formation of the feature map, the synaptic adaptive process is the last process of the self-organizing map algorithm. The synaptic weight vector $\overline{w}_j$ of neuron *j* in the network is required to change in the relation to the input vector $\overline{x}$. For Hebbian learning's rule, it is well suited for associative learning, when a synaptic weight is increased with a simultaneous occurrence of pre-synaptic and post-synaptic activities. However, the Hebbian learning is not satisfying formation, because it changes in connectivity neurons occur in one direction only, which finally drive all the synaptic weights into saturation. In order to solve this problem, the Hebbian hypothesis is modified by including a forgetting term $g(y_j) \cdot \overline{w}_j$, where $g(y_j)$ is some positive function of the response y_j.

In order to avoid the saturation problem, it can express the change to the weight vector of neuron *j* in the lattice as follows in Equation (2-15).

$$\Delta w_j = \eta_{SOM} y_j \bar{x} - g(y_j)\bar{w}_j \tag{2-15}$$

η_{SOM} is called the learning rate of SOM learning. In Equation (2-15), the first term on the right-hand side is the Hebbian term and the second term is the forgetting term. In order to satisfy the requirement, that is the constant term in the Taylor series expansion of $g(y_j)$ be zero when y_j is zero. Therefore, a linear function is used for $g(y_j)$ as shown by Equation (2-16), and it is able to simplify Equation (3.15) by setting y_j as expressed in Equation (2-17).

$$g(y_j) = \eta y_j \tag{2-16}$$

$$y_j = h_{j,c}(t) \tag{2-17}$$

Finally, the formation that given the synaptic weight vector $w_j(t)$ of neuron *j* at time *t* , and the updated weight vector $w_j(t+1)$ at time *t+1* is defined by Equation (2-18),

$$w_j(t+1) = w_j(t) + \eta_{SOM}(t) h_{j,c}(t)\left[\bar{x} - \bar{w}_j(t)\right] \tag{2-18}$$

which is applied to all neurons in the lattice that lie inside the topological neighborhood of the winning neuron *c*. The learning rate parameter $\eta_{SOM}(t)$ should be time varying throughout the learning step. In particular, it should start at an initial value as $\eta_{0,SOM}$, thereafter it should decrease

gradually with increasing time *t*. This condition can be satisfied by choosing an exponential decay for the learning rate $\eta_{SOM}(t)$ as shown in Equation (2-19), where τ is another time constant of the SOM algorithm.

$$\eta_{SOM}(t) = \eta_{0,SOM} \exp\left[-t/\tau\right] \tag{2-19}$$

2.1.4 Quantifying the goodness of SOM

During SOM learning, the several of the SOM parameters, such as learning parameters, map size and map topology can influence the formation of the final map. Therefore, in order to guarantee the effectiveness of the mapping, the system should be sure the mapping parameters have been correctly chosen [39], [40]. The measuring SOM quality can be done in many ways. One most important feature is the ability to preserve the topology in the projection. Topology preservation is a property that is not easy to define and even harder to measure, since usually a major reduction of dimensionality is performed and the information is necessarily lost in the projection process. For example, the topographic product can be used to optimize the map size for any given dataset, the average quantization error can be reduced by simply increasing the number of the neurons in the map, and the topographic error and the trustworthiness both measure the projection quality.

This section presents the widely used quality measurement methods of the self-organizing map such as the average quantization error and the topographic error [41], which are described below.

2.1.4.1 The average quantization error

A commonly used measurement is the average quantization error (E_{qe}) that can be used to determine the quality of the map and helping in choosing the suitable learning parameters and map

sizes [42]. The average quantization error is computed by the average distance of the sample input patterns to its best matching unit as shown in Equation (2-20).

$$E_{qe} = \frac{1}{N}\sum_{i=1}^{N}\left\|x_i - w_c\right\|, \tag{2-20}$$

where, N is the number of the input patterns used for training the map, $\left\|x_i - w_c\right\|$ is the Euclidean distance between the weights of the BMU (w_c) and the input pattern (x_i).

The optimum map is expected to yield the smallest average quantization error, which means that the data vectors are close to their prototypes. The average quantization error can be reduced by increasing the number of neurons, because the data samples are distributed more sparsely on the map. And the SOM with a lower average error is more accurate than the SOM with higher average error.

2.1.4.2 The topographic error

The topographic error (E_{te}) is the most simple of the topology preservation measures. It is also one of the errors proposed by Kohonen. The topographic error represents the proportion of all data vectors for which first and second best matching units (1st BMU and 2nd BMU) are not adjacent vectors. This error indicates the accuracy of the mapping in the preserving topology. The topographic error is calculated as shown in Equation (2-21), where *N* is the number of the input samples, the function $u(\bar{x}_i)$ is one if $\bar{x}_i$ data vector's first and second BMUs are not adjacent and otherwise $u(\bar{x}_i)$ is zero.

$$E_{te} = \frac{1}{N}\sum_{i=1}^{N}u(\bar{x}_i) \tag{2-21}$$

The total error is normalized to a range from 0 to 1, therefore the lower the topographic error is the better the self-organizing map preserves the topology.

2.2 Markov model

Andrei Markov is the mathematician who proposed the term "Markov model". The Markov models refer to the mathematical models as a type of stochastic model (chance model), which consists of the random variables that are defined on a given probability space, and are indexed by parameter. An important property of Markov models is "memoryless" property, where the next state of the system depends only on its current state, not on it is the previous state. Markov models can analyze as discrete-state or continuous-state, and can be also classified as homogeneous or non-homogeneous time. In this thesis, the discrete-state homogenous Markov model is proposed, because this model is very suitable to form the human emotions that is explained by previous researches [43], [44]. Also, the memoryless property of Markov models is important in modeling humans.

2.2.1 Markov model fundamentals

A Markov model consists of a list of the possible states of that system, the possible transition paths between those states and the state transition probabilities of those transitions. For representing a basic Markov model, the nodes in the model represent certain states and arrows which denote the probabilities of movement between states or the probabilities of getting out of the states. In the basic model, Let X_n, $n = 0,1,2...$, is a discrete time stochastic process with a discrete state space *S*, if the state space could be assumed as $\{1,2,...,N\}$ *or* $\{0,1,...,N-1\}$ in the finite state, and either $\{0,1,...\}$ *or* $\{1,2,...\}$ in the countably infinite state. To understand the behavior of a process, Equation (2-22) shows the finite dimensional distributions, which allows for the calculation of any path probability, for every n and every finite sequence of states $i_0, i_1, ..., i_n \in S$.

$$P\{X_0 = i_0, X_1 = i_1, ..., X_n = i_n\} = P\{X_0 = i_0, X_1 \in S, X_2 \in S, ..., X_n = i_n\} \quad (2\text{-}22)$$

Above Equation, that is a common choice for such structure is the assumption that the processes satisfies the Markov property, that is to mention, the probability of next state depends only on the current state. It is called a discrete time Markov chain for any process $X_n, n \geq 0,$ satisfying the Markov property as expressed by Equation (2-23).

$$P\{X_n = i_n \mid X_0 = i_0, ..., X_{n-1} = i_{n-1}\} = P\{X_n = i_n \mid X_{n-1} = i_{n-1}\} \quad (2\text{-}23)$$

Furthermore, the one step transition probability of a Markov chain from state i to state j can be denoted as $a_{ij}(n)$ that forms by Equation (2-24).

$$a_{ij}(n) = P\{X_n = j \mid X_{n-1} = i\}, \quad 1 \leq i, j \leq N \quad (2\text{-}24)$$

For the initial probability distribution of the process by π , which shows in Equation (2-25).

$$\pi_j = P\{X_0 = j\}, \quad j \in S \quad (2\text{-}25)$$

Returning to Equation (2-22) and then the problem of computing probabilities has been converted to one of simple multiplication as shown in Equation (2-26).

$$
\begin{aligned}
&P\{X_0 = i_0,...,X_n = i_n\} \\
&\quad = \; P\{X_n = i_n \mid X_0 = i_0,...,X_{n-1} = i_{n-1}\} \cdot P\{X_0 = i_0,...,X_{n-1} = i_{n-1}\} \\
&\quad = \; a_{i_{n-1},i_n} \cdot P\{X_0 = i_0,...,X_{n-1} = i_{n-1}\} \\
&\quad = \; a_{i_{n-1},i_n} \cdot P\{X_{n-1} = i_{n-1} \mid X_0 = i_0,...,X_{n-2} = i_{n-2}\} \cdot P\{X_0 = i_0,...,X_{n-2} = i_{n-2}\} \\
&\quad = \; a_{i_{n-1},i_n} \cdot a_{i_{n-2},i_{n-1}} \cdot P\{X_0 = i_0,...,X_{n-2} = i_{n-2}\} \\
&\quad \vdots \\
&\quad = \; \pi_0 \cdot a_{i_0,i_1} \cdot a_{i_1,i_2},...,a_{i_{n-2},i_{n-1}} \cdot a_{i_{n-1},i_n}
\end{aligned}
\tag{2-26}
$$

The first order or one step transition probabilities are mostly expressed in matrix form as expressed in Equation (2-27), where the matrix A is the state transition probability matrix for a Markov chain with state space $S = \{1, 2, 3, ..., N\}$, and a_{ij} is one step transition probabilities in the $N\,x\,N$ matrix.

$$
A \;=\; \{a_{ij}\} \;=\; \begin{pmatrix} a_{11} & a_{12} & \cdots & \cdots \\ a_{21} & a_{22} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & \cdots \end{pmatrix} \tag{2-27}
$$

A summary of state transition probability matrix, each Markov chain can be defined as the transition probabilities, $a_{ij}(n)$ at step *n* arranged into the probability matrix according to the current states as rows and the future states as columns, and row sums to one as shown in Equation (2-28).

$$
\sum_{j \in S} a_{ij} \;=\; 1 \tag{2-28}
$$

2.2.2 Example of the simple Markov model

The above stochastic process can consider as a sample 3 state Markov model of the weather as illustrated in Fig. 2-6, where $S_1, S_2, S_3, \ldots, S_N$ a set of states, N is the number of state (for simplicity N = 3). The weather is assumed that once a day and observed as being one of the following: Sunny (State 1), Rainy (State 2) and Cloudy (State 3), and the probabilities of tomorrow's weather based on today's weather is expressed in Table 2.1.

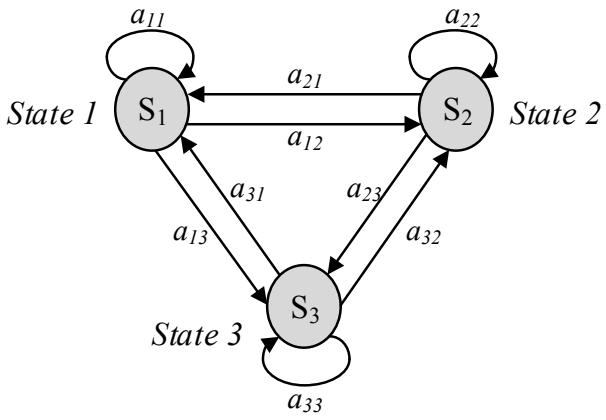


Fig. 2-6 Markov model with three states.

Table 2.1. The probabilities for tomorrow's weather based on today's weather

Tomorrow's weather				
Sunny (S1)	Rainy (S2)	Cloudy (S3)		
0.8	0.1	0.1	Sunny (S1)	**Today's weather**
0.1	0.6	0.3	Rainy (S2)	
0.2	0.3	0.5	Cloudy (S3)	

Let the weather on day 1 is Sunny (S1), ask the question: What is the probability that the weather for next 5 days will be "Sunny → Sunny → Rainy → Rainy → Cloudy …"? For the

formally, the observation sequence is defined as $O = \{S_1, S_1, S_1, S_2, S_2, S_3\}$ corresponding to the time (t). The probability of O can be expressed as below:

$$\begin{aligned} P(O \mid Model) &= P[S_1, S_1, S_1, S_2, S_2, S_3 \mid Model] \\ &= P[S_1] \cdot P[S_1 \mid S_1] \cdot P[S_1 \mid S_1] \\ &\quad \cdot P[S_2 \mid S_1] \cdot P[S_2 \mid S_2] \cdot P[S_3 \mid S_2] \\ &= \pi_1 \cdot a_{11} \cdot a_{11} \cdot a_{12} \cdot a_{22} \cdot a_{23} \\ &= (1) \cdot (0.8) \cdot (0.8) \cdot (0.1) \cdot (0.6) \cdot (0.3) \\ &= 1.152 x 10^{-2} \end{aligned}$$

An application of Markov models for the behavior selection system based on emotional variations is the Markovian emotional model, which will be described in the next chapter.

2.3 Motion Control system

2.3.1 Introduction

Definition: *A robot is a reprogrammable, multi-functional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks* [45]. That is the programmability, which gives a robot its utility and adaptability. the first successful applications of robot manipulators generally involved transfer of parts in the factories, or that could be programed to execute the sequential movements, such as moving to position A, closing a gripper, moving to position B, and opening a gripper, etc., and more complex applications, such as welding, grinding, deburring, and assembly that require not only more complex motion but also some form of the external sensing such as vision, tactile, distance, or force sensing, due to increase accurately the interaction of the robot with its environment.

Kinematics is the science of motion that studies the body movements without considering the forces or moments, size, shape and weight. The formulation of the suitable kinematics models for a

robot mechanism is very important for analyzing the part movements of industrial manipulators. There are commonly two different spaces used in kinematics modeling of manipulators such as, Cartesian space and Quaternion space. There are many ways to represent rotation, including the following: Euler angle, Roll-Pitch-Yaw angles, Cayley-Klein parameters, Pauli spin matrices, orthonormal matrices and Hamilton's quaternions. The general transformation between two joints that requires four parameters is Denavit & Hartenberg [46]. These parameters known as the D-H parameters have become the standard for describing robot kinematics models. For representing the rotation and transition in a compact form of transformation vector that can be determined as Dual quaternion. For example, if the body is represented nine elements in homogeneous transformations, the dual quaternion can reduce the number of elements to four parameters. It is accepted as an advantage in terms of computational robustness and storage efficiency for dealing with the kinematics of robot chains [47].

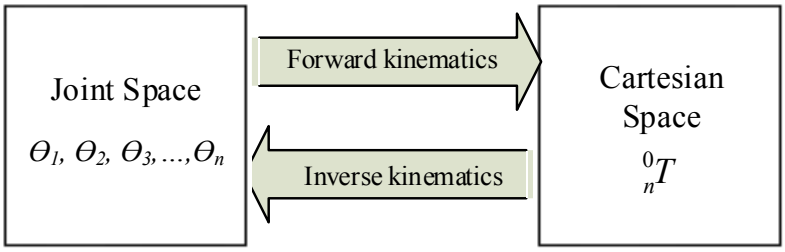


Fig. 2-7 Schematic representation of forward and inverse kinematics.

The robot kinematics can be divided into forward kinematics and inverse kinematics. Forward kinematics problem is static geometrical problem in solving the equations. Because the typically manipulator will be able to sense its own position by using internal sensors (position encoders) that can be directly measured the joint angles, and then it is determined the position and orientation of the end effector or tool frame, which relative to the based frame. But the inverse kinematics problem is much more difficult problem than forward kinematics. Singularities and nonlinearities that make the problem more difficult to solve the solution of the inverse kinematics

problem. Relationship between forward and inverse kinematics is shown in Fig. 2-7. There are two main solution techniques for solving the inverse kinematics problem that are analytical and numerical methods. In the first type is the analytical method, the joint variables are determined analytically according to given configuration data. The second type of solution is the numerical method, the joint variables are obtained based on the numerical technique methods.

2.3.2 Rigid motions and Homogeneous transformations

An essential part of the robot kinematics is involved with establishing various coordinate frames to represent the positions and orientations of the rigid objects and with transformations among these coordinate frames. In this section, the operations of rotation and transition are described, and introduce the notion of homogenous transformations, these transformations combine the operations of rotation and transition into a single matrix multiplication that are used to derive as forward kinematics equations of the manipulators.

2.3.2.1 Representing positions and rotations

For the representing positions, that schemes for points and vectors. There are two fundamental approaches to geometric reasoning as the synthetic approach and the analytic approach. The first approach is one reason directly about points or lines, while in the latter, one presents these entities using coordinates or equations, and is performed via algebraic manipulations. The latter approach requires the choice of the reference coordinate frame. In robotic systems, one typically uses analytic reasoning, since the robot tasks are often defined as Cartesian coordinates.

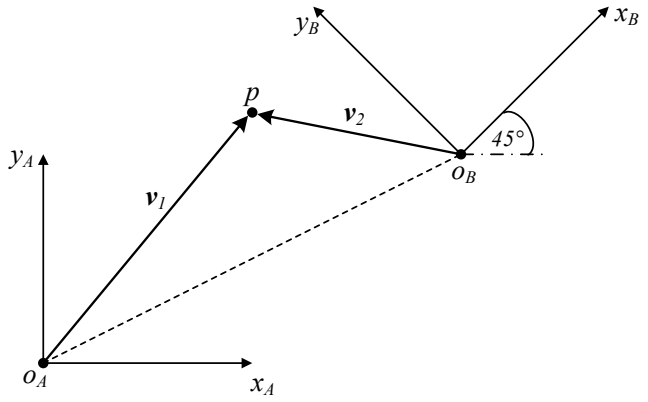


Fig. 2-8 Two coordinate frames, a point *p* and two vector $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$.

The Fig. 2-8 shows two coordinate frames that differ in orientation by angle of 45°. In the figure, the coordinates of the point *p* is specified with respect to either frame $o_A x_A y_A$ or frame $o_B x_B y_B$. The point *p* is assigned to the coordinate vector $p^A = [4,5]^T$, and in the latter case $p^B = [-3,4]^T$, that means a point in space, while both p^A and p^B are coordinate vectors that represent the location of this point in space with respect to coordinate frames $o_A x_A y_A$ and $o_B x_B y_B$, respectively. The coordinates that can be represented the position of the origin of one coordinate frame with respect to another as $o_B^A = [9,4]^T$ and $o_A^B = [-9,3]^T$. Therefore, a point corresponds to a specific location in space, a vector specifies a direction and magnitude. The vectors can be used to represent displacement or forces. Consider the Fig. 2-8, the displacement from origin o_A and origin o_B to point *p* is given by the vectors v_1 and v_2 respectively. Both vectors are geometric entities that are invariant with respect to the choice of coordinate frames, but the representation by coordinates of these vectors depends on the choice of the reference coordinate frames such as $v_1^A = [4,5]^T$, $v_1^B = [6,1]^T$, $v_2^A = [-5,1]^T$ and $v_2^B = [-3,4]^T$.

In representing rotations, to represent the relative position and orientation of one rigid body with respect to another, the coordinate frames are attached to each part, and then specify the relationships between these coordinate frames. In the case of the representing rotations that can be described as follows:

- ***Rotation in the plane***

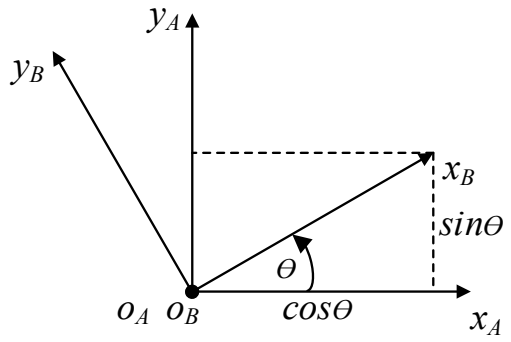


Fig. 2-9 Coordinate frame $o_B x_B y_B$ is oriented at angle θ with respect to $o_A x_A y_A$

Consider Fig. 2-9 that shows two coordinate frames, with frame $o_B x_B y_B$ being obtained by rotating frame $o_A x_A y_A$ by angle θ. And a way to define the orientation is to specify the coordinate vectors for the axes of frame $o_B x_B y_B$ with respect to coordinate frame $o_A x_A y_A$ as expressed by Equation (2-29).

$$R_B^A = [x_B^A \mid y_B^A] \tag{2-29}$$

Where, x_B^A and y_B^A are the coordinates in frame $o_A x_A y_A$ of unit vectors x_B and y_B respectively. This formation is called “Rotation matrix”, which have a number of special properties that discusses below.

In the two-dimensional case, it is easy to compute the entries of the rotation matrix as shown in Equation (2-30).

$$R_B^A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \text{ where } x_B^A = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \text{ and } y_B^A = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} \tag{2-30}$$

To summarize, R_B^A is a matrix whose column vectors are the coordinates of the unit vectors along the axes of frame $o_B x_B y_B$ expressed relative to frame $o_A x_A y_A$.

- ***Rotations in three dimensions***

The projection technique is used in three dimensions, each axis of the frame $o_B x_B y_B$ is projected onto coordinate frame $o_A x_A y_A$. The resulting rotation matrix is given by Equation (2-31).

$$R_B^A = \begin{bmatrix} x_B \cdot x_A & y_B \cdot x_A & z_B \cdot x_A \\ x_B \cdot y_A & y_B \cdot y_A & z_B \cdot y_A \\ x_B \cdot z_A & y_B \cdot z_A & z_B \cdot z_A \end{bmatrix} \tag{2-31}$$

For example, the frame $o_A x_A y_A$ is rotated through an angle θ about the z_A axis, in order to find the result of transformation matrix R_B^A. Form the right hand defines the positive sense for the angle θ to be such that rotation by θ about the z axis would advance a right-hand threaded screw along the positive z axis as illustrated in Fig. 2-10, where $x_B \cdot x_A = \cos\theta$, $y_B \cdot x_A = -\sin\theta$, $x_B \cdot y_A = \sin\theta$, $y_B \cdot y_A = \cos\theta$ and $z_B \cdot z_A = 1$. While all other dot products are zero. Thus, the rotation matrix R_B^A has a particularly simple form as expressed in Equation (2-32) and also is called a basic rotation matrix (about the z axis) $R_{z,\theta}$.

$$R_B^A = R_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2-32}$$

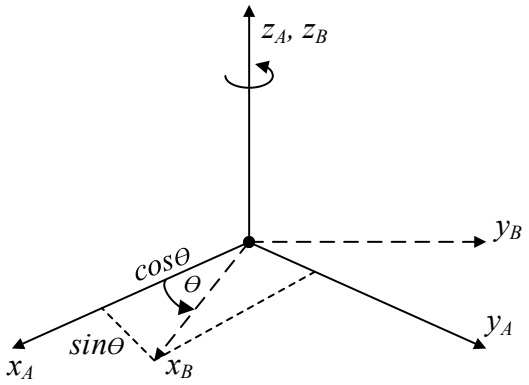


Fig. 2-10 Rotation about z_A by an angle θ

Similarity, the basic rotation matrices are able to represent rotation about the x axis and y axis are shown in Equation (2-33) and (2-34) respectively.

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{2-33}$$

$$R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{2-34}$$

2.3.2.2 Rotational transformations

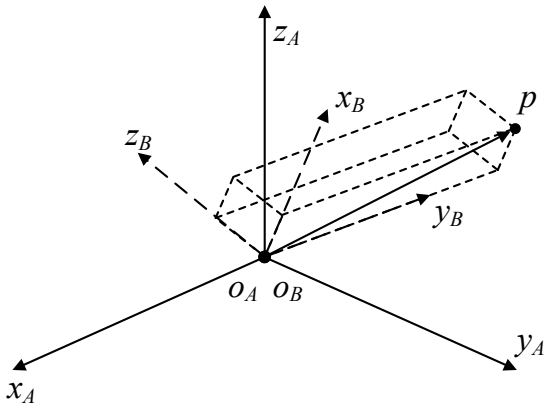


Fig. 2-11 The rigid object is attached in the coordinate frame

In Fig. 2-11 shows the rigid object that is attatched onto a coordinate frame $o_B x_B y_B$. Given the coordinate of point p with respect to the frame $o_B x_B y_B$ and the coordinates $p^B = [u, v, w]^T$ define as Equation (2-35). And a similar way, the coordinate p^A can be obtained by projecting the point p onto coordinate axes of the frame $o_A x_A y_A$ as shown in Equation (2-36).

$$p = [u x_B + v y_B + w z_B] \tag{2-35}$$

$$p^A = [p \cdot x_A, p \cdot y_A, p \cdot z_A]^T \tag{2-36}$$

In order to explain about the rotaional transformations, these two equations are combined. Finally, the final equation is solved as Equation (2-37).

$$
\begin{aligned}
p^A &= \begin{bmatrix} (ux_B + vy_B + wz_B)\cdot x_A \\ (ux_B + vy_B + wz_B)\cdot x_A \\ (ux_B + vy_B + wz_B)\cdot x_A \end{bmatrix} \\
&= \begin{bmatrix} x_B \cdot x_A & y_B \cdot x_A & z_B \cdot x_A \\ x_B \cdot y_A & y_B \cdot y_A & z_B \cdot y_A \\ x_B \cdot z_A & y_B \cdot y_A & z_B \cdot z_A \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \\
&= R_B^A p^B
\end{aligned}
\tag{2-37}
$$

Briefly, the rotation matrix R_B^A can be used both to represent the orientation of the coordinate frame $o_B x_B y_B$ with respect to frame $o_A x_A y_A$, and transform the coordinates of a point from one frame to another.

2.3.2.3 Composition of rotations

In this sub-topic, the composition of rotations is discussed that can be divided into two types as the rotation with respect to the current frame and the fixed frame. The details of both types are described below:

- ***Rotation with respect to the current frame***

Suppose, the third coordinate frame $o_C x_C y_C$ is added in the plane that relates to the frames $o_A x_A y_A$ and $o_B x_B y_B$ by rotational transformations. Given a point p is represented by coordinate specified with respect to any of these three frames, the relationship among the representations of p is expressed by Equation (2-38).

$$
\begin{aligned}
p^A &= R_B^A p^B \\
p^B &= R_C^B p^C \\
p^A &= R_C^A p^C
\end{aligned}
\tag{2-38}
$$

Where, R_B^A and R_C^A that represent rotations relative to the frame $o_A x_A y_A$ while R_C^B represents a rotation relative to frame $o_B x_B y_B$. Consider in the Equation (2-38) that can determine as the composition law for rotational transformations which shows in Equation (2-39).

$$R_C^A = R_B^A R_C^B \tag{2-39}$$

That means the resulting frame $o_C x_C y_C$ has orientation with respect to $o_A x_A y_A$ given by $R_B^A R_C^B$. It is called the current frame because the frame relative to which the rotation occurs the frame. For example, if a rotation matrix R represents a rotation of angle ϕ about the current y axis followed by a rotation of angle θ about the current z axis as shown in Fig. 2-12.

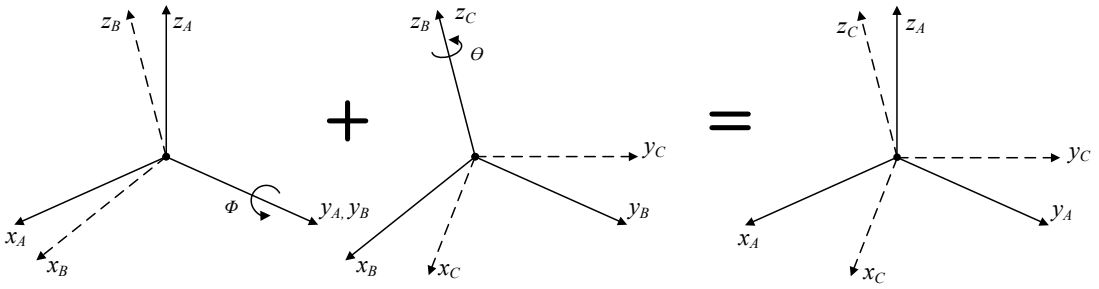


Fig. 2-12 Composition of rotations about current frame

Then the resulting matrix R is calculated as:

$$\begin{aligned} R &= R_{y,\phi} \cdot R_{z,\theta} \\ &= \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\phi\cos\theta & -\cos\phi\sin\theta & \sin\phi \\ \sin\theta & \cos\theta & 0 \\ -\sin\phi\cos\theta & \sin\phi\sin\theta & \cos\phi \end{bmatrix} \end{aligned}$$

- ***Rotation with respect to the fixed frame***

In many cases, it is to operate a sequence of rotations, each about a given fixed coordinate frames, rather than about successive current frames. For example, the coordinate frame $o_A x_A y_A$ is referred as the fixed frame, suppose the 3D plane has two frames $o_A x_A y_A$ and $o_B x_B y_B$ related by the rotational transformation R_B^A. From the previous section, the representation for rotational matrix in the current frame $o_B x_B y_B$ is given by $(R_B^A)^{-1} R R_B^A$. Therefore, the composition law for applying about the current axis as shown in Fig. 2-13 and expressed in Equation (2-40).

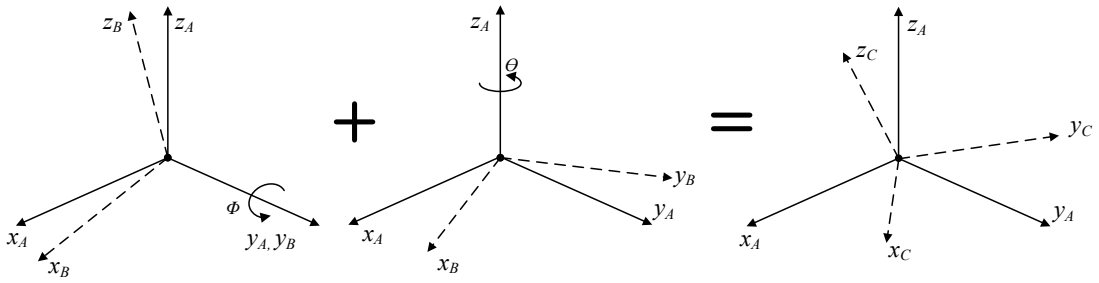


Fig. 2-13 Composition of rotations about fixed frame

$$R_C^A = R_B^A \left[(R_B^A)^{-1} R R_B^A \right] = R R_B^A \tag{2-40}$$

Thus, when a rotation is performed with respect to the world coordinate frame, the current rotation matrix is pre-multiplied by R to obtain the desired rotation matrix. Consider in Fig. 2-13 that can be formed the rotational equation as:

$$R = R_{y,\phi} \left[R_{y,-\phi} R_{z,\theta} R_{y,\phi} \right] = R_{z,\theta} R_{y,\phi}$$

Using the rule of composition of rotational transformations, it is an easy for determining the result of multiple sequential rotational transformations.

2.3.2.4 Homogeneous transformations

In order to represent both positions and orientations. This section, these two concepts are combined to define homogeneous transformations.

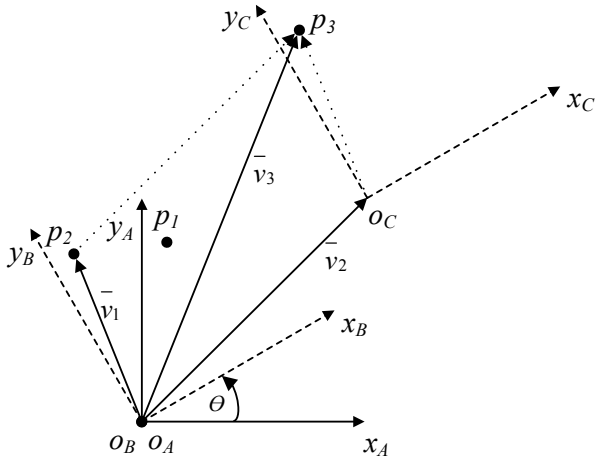


Fig. 2-14 Homogeneous transformations in two-dimensional

Consider Fig. 2-14, frame $o_B x_B y_B$ is obtained by rotating frame $o_A x_A y_A$ by angle θ, and frame $o_C x_C y_C$ is obtained by translating frame $o_B x_B y_B$ by the displacement $\bar{v}_2$. Suppose, the point p_1 is attached to coordinate frame $o_A x_A y_A$, then p_2 is the location of p_1 after the rotation, and p_3 is the location of p_1 after the translation. Note, the point p_3 is displaced by the vector $\bar{v}_3$ from the origin of frame $o_A x_A y_A$, and the vector $\bar{v}_3$ can be represented as $\bar{v}_3 = \bar{v}_1 + \bar{v}_2$. The coordinate for the vector $\bar{v}_1$ can be obtained by the rotation matrix to coordinates that represent p_2 in frame $o_B x_B y_B$ as expressed in Equation (2-41).

$$v_1^A = R_B^A p_2^B = R_C^A p_3^C \tag{2-41}$$

If $\bar{v}_2$ denotes by d_C^A that is the displacement of the origin of $o_C x_C y_C$, expressed relative to frame $o_A x_A y_A$, then the point p_3^A can determine by Equation (2-42).

$$p_3^A = R_C^A p_3^C + d_C^A \tag{2-42}$$

In definition, a transformation of the form given in Equation (2-42) is called to define a rigid motion if R is orthogonal. From Fig. 2-14, the point can be considered as two rigid motions as shown in Equations (2-43) and (2-44).

$$p^A = R_B^A p^B + d_B^A \tag{2-43}$$

$$p^B = R_C^B p^C + d_C^B \tag{2-44}$$

And the Equation (2-45) describe by substituting the expression for p^B from Equation (2-44) into Equation (2-43).

$$p^A = R_B^A R_C^B p^C + R_B^A d_C^B + d_B^A \tag{2.45}$$

Since the relationship between p^A and p^C is also a rigid motion, it can be described by Equation (2-46).

$$p^A = R_C^A p^C + d_C^A \tag{2-46}$$

Considering between Equations (2-45) and (2-46), the relationship is expressed by Equation (2-47), where R_C^A represents the orientation transformations and d_C^A shows the vector from the origin o_A to the origin o_C that has coordinates given by the sum of d_B^A.

$$\begin{aligned} R_C^A &= R_B^A R_C^B \\ d_C^A &= R_B^A d_C^B + d_B^A \end{aligned} \tag{2-47}$$

Modify the Equation (2-47) is to be the matrix identity as Equation (2-48).

$$\begin{bmatrix} R_B^A & d_B^A \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_C^B & d_C^B \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_B^A R_C^B & R_B^A d_C^B + d_B^A \\ 0 & 1 \end{bmatrix} \tag{2-48}$$

From the Equation (2-48) that forms as the transformation matrices or called homogenous transformations as expressed in Equation (2-49).

$$H = \begin{bmatrix} R & d \\ 000 & 1 \end{bmatrix} \tag{2-49}$$

The set of basic homogeneous transformations can be calculated by Equations (2-50), (2-51) and (2-52) for transitions and rotations about the x, y and z axes, respectively.

For x-axis

$$Trans_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \; Rot_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2-50}$$

For y-axis

$$Trans_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix};\ Rot_{y,\beta} = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2-51}$$

For z-axis

$$Trans_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 0 \end{bmatrix};\ Rot_{z,\gamma} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2-52}$$

2.3.3 Robot Kinematics

This section organizes in the following manner, the forward and inverse kinematics transformations for an open kinematics chain are described based on the homogeneous transformation.

2.3.3.1 Forward kinematics

A robotic manipulator is composed of serial links which are connected to each other revolute or prismatic joints from the base frame to the end-effector or tool frame. Determining the position and orientation of the end-effector in terms of the joint variables is called as forward kinematics. One method will be used for a suitable kinematics model that is Denavit-Hartenberg method. This method uses four parameters (D-H parameters) is the most common method for describing the robot kinematics. The parameters are the link length a_{i-1}, link twist α_{i-1}, link offset d_i and joint angle θ_i. In Fig. 2-15 that shows coordinate frame assignment for general manipulator. As description in the figure, the distance from Z_{i-1} to Z_i along X_i is assigned as a_{i-1}, the angle between Z_{i-1} to Z_i about X_i is assigned as α_{i-1}, the distance from X_{i-1} to X_i along Z_i is assigned as d_i and the angle

between X_{i-1} to X_i about Z_i is assigned as θ_i [48]. Thus, the general transformation matrix ${}^{i-1}_{i}T$ for a single link can be obtained as follows by Equation (2-53).

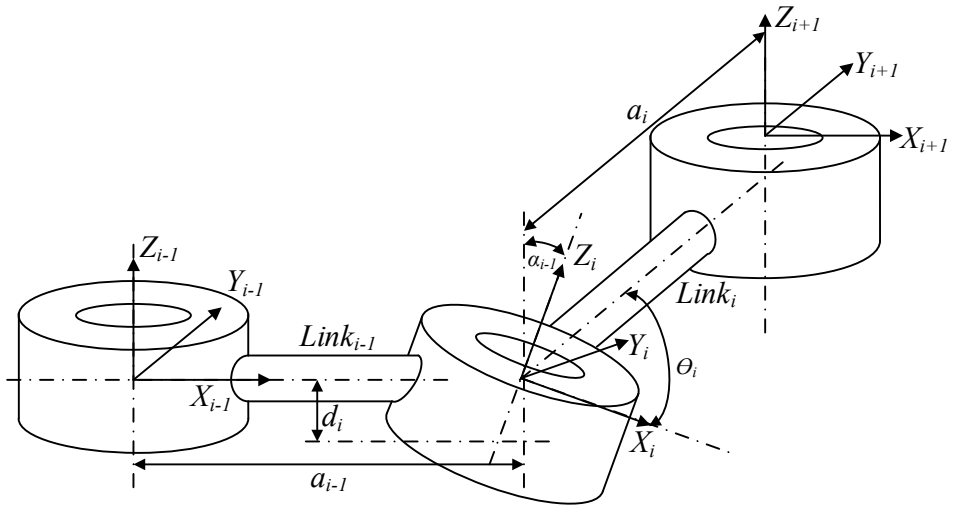


Fig. 2-15 Denavit-Hartenberg frame assignment

$$
\begin{aligned}
{}^{i-1}_{i}T &= R_x(\alpha_{i-1})D_x(a_{i-1})R_z(\theta_i)D_z(d_i) \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_{i-1} & -s\alpha_{i-1} & 0 \\ 0 & s\alpha_{i-1} & c\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

$$
{}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta c\alpha_{i-1} & c\theta c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta s\alpha_{i-1} & c\theta s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2\text{-}53)
$$

The forward kinematics of the end-effector with respect to the based frame determines by multiplying all transition matrices as expressed by Equation (2-54).

$$
\begin{aligned}
{}^{base}_{end-effector}T &= {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T \ldots {}^{n-1}_{n}T \\
&= \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{2-54}
$$

r_{ij} define as the rotational elements of the transformation matrix (i and j = 1, 2 and 3). The elements of the position vector denote with p_x, p_y and p_z.

2.3.3.2 Inverse kinematics

In the previous section, the forward kinematics is used to determine the end-effector's position and orientation in terms of the joint variables. But in this section, the inverse kinematics problem is solved in order to find the joint variables in terms of the end-effector's position and orientation. There are two solutions approaches namely geometric and algebraic that used for solving the inverse kinematics problem.

- ***Geometric Solution***

For the common kinematics assignments, the geometric solution approach is based on decomposing the spatial geometry of the manipulator into several plane geometry problems. 2-DOFs planar manipulator is the simple robot structures that is considered in order to derive the kinematics equation as shown in Fig. 2-16, where it has 2 revolute joints and link lengths are L_1 and L_2. And the components of point p are determined as follow Equations (2-55) and (2-56).

$$p_x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) \tag{2-55}$$

$$p_y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) \tag{2-56}$$

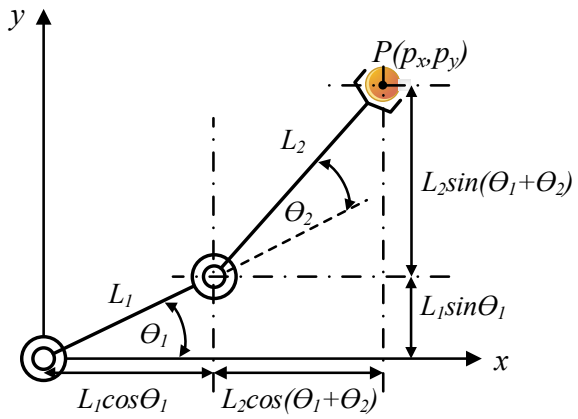


Fig. 2-16 The 2-DOF planar manipulator

The solution of θ_2 can be computed from summation of squaring both Equations (2-55) and (2-56) as follows:

$$\begin{aligned} {p_x}^2 + {p_y}^2 &= (L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2))^2 + (L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2))^2 \\ &= {L_1}^2(\cos^2\theta_1 + \sin^2\theta_1) + {L_2}^2(\cos^2(\theta_1 + \theta_2) + \sin^2(\theta_1 + \theta_2)) \\ &\quad + 2L_1L_2(\cos\theta_1 \cos(\theta_1 + \theta_2) + \sin\theta_1 \sin(\theta_1 + \theta_2)) \\ {p_x}^2 + {p_y}^2 &= {L_1}^2 + {L_2}^2 + 2L_1L_2 \cos\theta_2 \end{aligned}$$

And so

$$\cos\theta_2 = \frac{p_x^2 + p_y^2 - L_1^2 - L_2^2}{2L_1L_2};\ \sin\theta_2 = \pm\sqrt{1 - \left(\frac{p_x^2 + p_y^2 - L_1^2 - L_2^2}{2L_1L_2}\right)^2}$$

Finally, two possible solutions for θ_2 can be written as Equation (2-57).

$$\theta_2 = A\tan 2\left[\pm\sqrt{1-\left(\frac{p_x^2+p_y^2-L_1^2-L_2^2}{2L_1L_2}\right)^2}, \frac{p_x^2+p_y^2-L_1^2-L_2^2}{2L_1L_2}\right] \tag{2-57}$$

Next step, Find the solution of θ_1 in terms of link parameters and known variable θ_2 as follows. Let's multiply each side of Equation (2-55) by $\cos\theta_1$ and Equation (2-56) by $\sin\theta_1$.

$$\cos\theta_1 p_x = L_1\cos^2\theta_1 + L_2\cos^2\theta_1\cos\theta_2 - L_2\cos\theta_1\sin\theta_1\sin\theta_2$$
$$\sin\theta_1 p_y = L_1\sin^2\theta_1 + L_2\sin^2\theta_1\cos\theta_2 + L_2\sin\theta_1\cos\theta_1\sin\theta_2$$
$$\cos\theta_1 p_x + \sin\theta_1 p_y = L_1\cos^2\theta_1 + L_2\cos^2\theta_1\cos\theta_2 + L_1\sin^2\theta_1 + L_2\sin^2\theta_1\cos\theta_2$$

The simplified equation obtained as Equation (2-58).

$$\cos\theta_1 p_x + \sin\theta_1 p_y = L_1 + L_2\cos\theta_2 \tag{2-58}$$

Again consider in the Equations (2-55) and (2-56) by multiplying $-\sin\theta_1$ and $\cos\theta_1$ respectively, then adding the resulting equations produce as:

$$-\sin\theta_1 p_x = -L_1\sin\theta_1\cos\theta_1 - L_2\sin\theta_1\cos\theta_1\cos\theta_2 + L_2\sin^2\theta_1\sin\theta_2$$
$$\cos\theta_1 p_y = L_1\sin\theta_1\cos\theta_1 + L_2\cos\theta_1\sin\theta_1\cos\theta_2 + L_2\cos^2\theta_1\sin\theta_2$$
$$-\sin\theta_1 p_x + \cos\theta_1 p_y = L_2\sin\theta_2(\cos^2\theta_1 + \sin^2\theta_1)$$

The simplified equation obtained as Equation (2-59).

$$-\sin\theta_1 p_x + \cos\theta_1 p_y = L_2\sin\theta_2 \tag{2-59}$$

Then, multiply each side of Eqaution (2-58) by p_x and Equation (2-59) by p_y and add the resulting equations in order to find $\cos\theta_1$ as:

$$\begin{aligned}
\cos\theta_1 p_x^2 + \sin\theta_1 p_y p_x &= p_x(L_1 + L_2\cos\theta_2) \\
-\sin\theta_1 p_y p_x + \cos\theta_1 p_y^2 &= p_y L_2 \sin\theta_2 \\
\cos\theta_1 p_x^2 + \cos\theta_1 p_y^2 &= p_x(L_1 + L_2\cos\theta_2) + p_y L_2 \sin\theta_2
\end{aligned}$$

As a result, two possible solutions for θ_1 can be written as Equation (2-60).

$$\begin{aligned}
\cos\theta_1 &= \frac{p_x(L_1 + L_2\cos\theta_2) + p_y L_2 \sin\theta_2}{p_x^2 + p_y^2}; \\
\sin\theta_1 &= \pm\sqrt{1 - \left(\frac{p_x(L_1 + L_2\cos\theta_2) + p_y L_2 \sin\theta_2}{p_x^2 + p_y^2}\right)^2} \\
\theta_1 &= A\tan 2\left(\begin{array}{c} \pm\sqrt{1 - \left(\dfrac{p_x(L_1 + L_2\cos\theta_2) + p_y L_2 \sin\theta_2}{p_x^2 + p_y^2}\right)^2}, \\ \dfrac{p_x(L_1 + L_2\cos\theta_2) + p_y L_2 \sin\theta_2}{p_x^2 + p_y^2} \end{array}\right)
\end{aligned} \tag{2-60}$$

Although the planar manipulator has a very simple structure, as can be seen its inverse kinematics solution based on geometric approach, but it is very cumbersome for solving the inverse kinematics problem. And it is not suitable solutions when the manipulators that have many links and joints. Thus, the other method is an algebraic solution approach that is chosen for the inverse kinematics solutions.

- ***Algebraic Solution***

An algebraic solution approach is a suitable method for the manipulators with more links. Recall the Equation (2-54) to find the inverse kinematics solution for a 6-DOFs manipulator as shown in Equation (2-61).

$$ {}^{0}_{6}T = {}^{0}_{1}T(q_1)\,{}^{1}_{2}T(q_2)\,{}^{2}_{3}T(q_3)\ldots{}^{5}_{6}T(q_6) = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2-61} $$

To find the inverse kinematics solution for the first joint (q_1) as a function of the known of ${}^{0}_{6}T$, the link transformation inverses are pre-multiplied as Equation (2-62), where $\left[{}^{0}_{1}T(q_1)\right]^{-1}\,{}^{0}_{1}T(q_1) = I$ is an identity matrix,

$$ \begin{aligned} \left[{}^{0}_{1}T(q_1)\right]^{-1}\,{}^{0}_{6}T &= \left[{}^{0}_{1}T(q_1)\right]^{-1}\,{}^{0}_{1}T(q_1)\,{}^{1}_{2}T(q_2)\,{}^{2}_{3}T(q_3)\ldots{}^{5}_{6}T(q_6) \\ &= {}^{1}_{2}T(q_2)\,{}^{2}_{3}T(q_3)\ldots{}^{5}_{6}T(q_6) \end{aligned} \tag{2-62} $$

For the other variables, the following equations are obtained as a similar manner. All of them show in by Equation (2-63).

$$ \begin{aligned} \left[{}^{0}_{1}T(q_1)\,{}^{1}_{2}T(q_2)\right]^{-1}\,{}^{0}_{6}T &= {}^{2}_{3}T(q_3)\,{}^{3}_{4}T(q_4)\,{}^{4}_{5}T(q_3)\,{}^{5}_{6}T(q_6) \\ \left[{}^{0}_{1}T(q_1)\,{}^{1}_{2}T(q_2)\,{}^{2}_{3}T(q_3)\right]^{-1}\,{}^{0}_{6}T &= {}^{3}_{4}T(q_4)\,{}^{4}_{5}T(q_3)\,{}^{5}_{6}T(q_6) \\ \left[{}^{0}_{1}T(q_1)\,{}^{1}_{2}T(q_2)\,{}^{2}_{3}T(q_3)\,{}^{3}_{4}T(q_4)\right]^{-1}\,{}^{0}_{6}T &= {}^{4}_{5}T(q_3)\,{}^{5}_{6}T(q_6) \\ \left[{}^{0}_{1}T(q_1)\,{}^{1}_{2}T(q_2)\,{}^{2}_{3}T(q_3)\,{}^{3}_{4}T(q_4)\,{}^{4}_{5}T(q_3)\right]^{-1}\,{}^{0}_{6}T &= {}^{5}_{6}T(q_6) \end{aligned} \tag{2-63} $$

There are twelve sets of nonlinear equations to be solved. If the elements on the left hand side which are the function of q_1 are equated with the elements on the right hand side, then the joint variable q_1 can be solved as functions of $r_{11}, r_{12}, r_{13}, ..., r_{33}, p_x, p_y, p_z$ and the fixed link parameters. To find the suitable equation for the solution of the inverse kinematics problem, any equations can be used some trigonometric equations that are given in Table 2.2.

Table 2.2. Some trigonometric equations and solution can be used in inverse kinematics

	Equations	**Solutions**
1	$a\sin\theta + b\cos\theta = c$	$\theta = A\tan 2(a,b) \pm A\tan 2\left(\sqrt{a^2+b^2-c^2}, c\right)$
2	$a\sin\theta + b\cos\theta = 0$	$\theta = A\tan 2(-b,a) or \theta = A\tan 2\left(b,-a\right)$
3	$\cos\theta = a; \sin\theta = b$	$\theta = A\tan 2(b,a)$
4	$\cos\theta = a$	$\theta = A\tan 2\left(\pm\sqrt{1-a^2}, a\right)$
5	$\sin\theta = a$	$\theta = A\tan 2\left(a, \pm\sqrt{1-a^2}\right)$

2.4 Summary

In this chapter, the fundamental theories (an unsupervised learning as a Self-Organizing Map (SOM), Markov theory and Motion control system) have described above that will be used to generate and develop the behavior selection system based on the emotional variations for Conbe-I robot, which autonomously determines and outputs the most suitable behavior and emotional expression according the internal and external situations.

Chapter 3

System Structure of Conbe-I robot

Designing robots with cognition and consciousness resembling for humans and animal have become an important application of intelligent autonomous robots, in order to achieve a more effective human-robot interaction. Thus, in this thesis, the behavioral-emotional selection model is proposed based on the Self-Organizing Map (SOM) learning and the discrete stochastic state-space mathematical model (Markovian model) that mainly consider the issues of an autonomous action selection corresponds to the emotional state transition. The system structure of Conbe-I robot and the details of each part will be clearly described in this chapter.

3.1 System Configuration

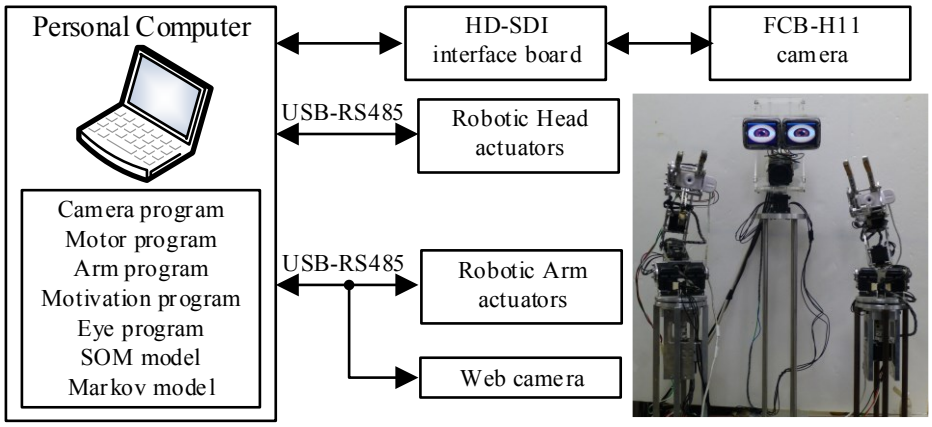


Fig. 3-1 The basic configuration of Conbe-I robot

Fig 3-1 shows the basic configuration of the robot, consisting of an autonomous control system (Camera program, Motivation program, Arm program, Eye program and Motor program), Self-Organizing Map module (Behavior map and Emotion map) and Markovian model in a personal computer and the actuator system (Arm and Head parts). The configuration of the Conbe-I is divided into 2 parts, namely an arm part and a head part.

3.1.1 Robot arm configurations

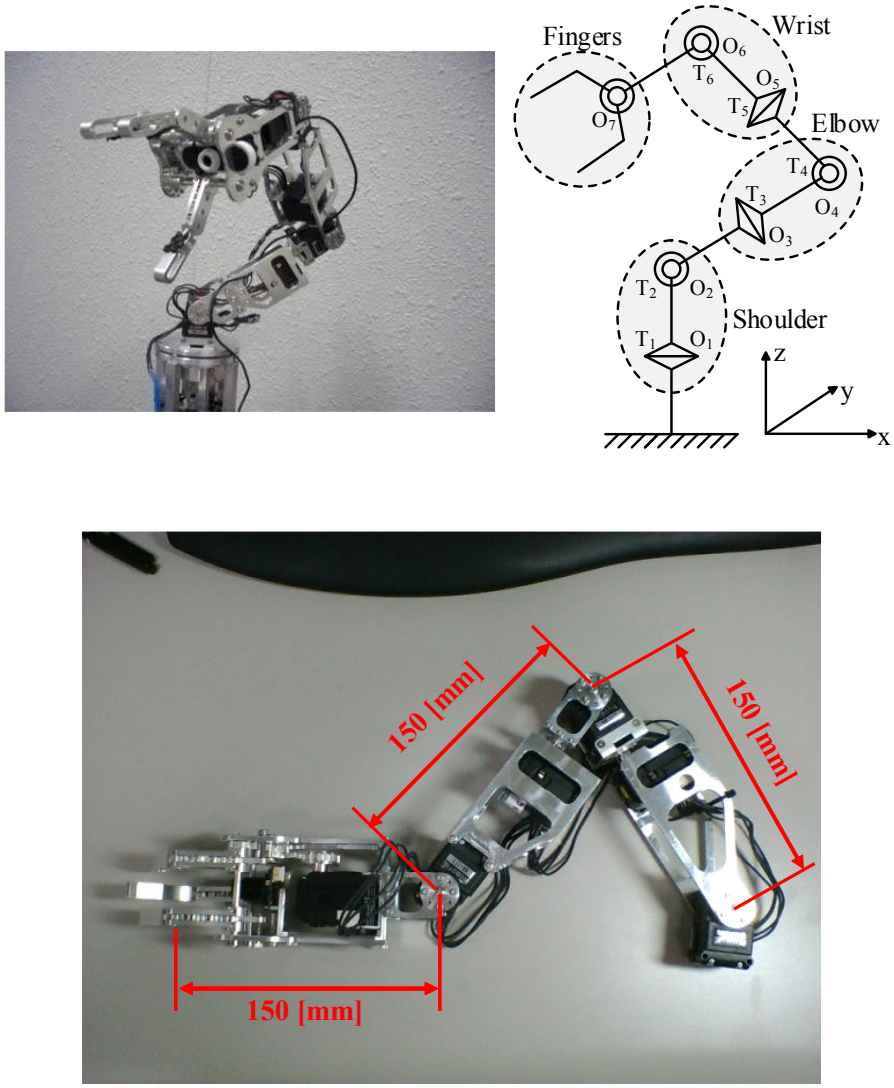


Fig. 3-2 Configurations of the robot arm

In Fig.3-2 that shows the configurations of the robot arm. The total length of each arm part is 450 mm, and the weight of the body is about 0.8 kg. Each arm has 7 levels of flexibility: the shoulder (Joint 1, Joint 2), the elbow (Joint 3, Joint 4), the wrist (Joint 5, Joint 6) and the hand part (Joint 7). The hand part has 3 fingers and a small web camera which is installed on the palm of the robotic hand in order to recognize the target objects and external situation.

3.1.1.1 Actuators in robot arm

An actuator for each joint uses a Dynamixel DX-117 manufactured by ROBOTIS, which has a decelerator and an angular sensor, and it is able to control the position and velocity using a target angle, torque limit, speed limit. In communicating, the RS-485 serial data communication is used to control multiple actuators with the personal computer. Fig. 3-3 shows the external dimension of Dynamixel DX-117 actuator and the main specifications are presented in Table 3.1.

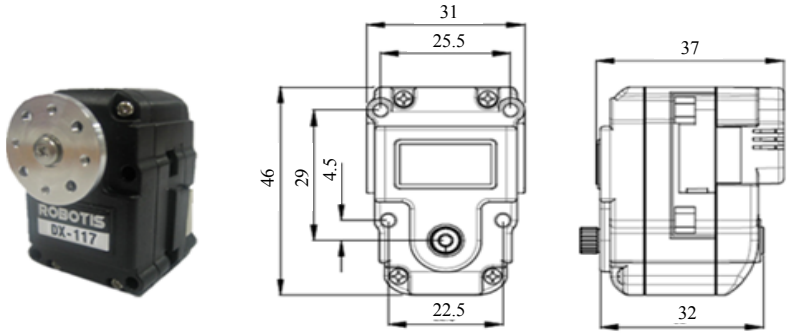


Fig. 3-3 Dimension of DX-117

Table 3.1 The specifications of Dynamixel DX-117 actuator

Weight	66 g
Dimension	31 mm x 46 mm x 37 mm
Resolution	0.29°
Gear Reduction Ratio	192.6 : 1
Stall Torque	3.7 N·m (at 18.5 V, 1.9 A)
No load speed	85 rpm (at 18.5 V)
Running Degree	0° ~ 300°
Running Temperature	-5°C ~ +80°C
Voltage	12 V ~ 18.5 V (Recommended Voltage : 14.8 V)
Link (Physical)	RS-485 Multi Drop (daisy chain type connector)
ID actuator	254 ID (0~253)
Communication Speed	7343 bps ~ 1 Mbps
Types of Feedback	Position, Temperature, Load, Input voltage, etc.
Material	Full Metal Gear, Engineering Plastic Body

3.1.1.2 Web camera

Fig. 3-4 UCAM-DLV3000T (a web camera)

Fig 3-4 shows an UCAM-DLV3000T web camera that is used to get the images from its surrounding environment. It is 3 million pixels of web camera realizing high-definition and high resolution of wonder. In correspondence with "UVC" (USB video class) which is standard of USB 2.0, it is not necessary to install driver from CD-ROM and is "blow connection" to only connect to the universal serial bus port of a desktop PC or all of notebook PC and to be usable immediately type.

3.1.2 Robot head configurations

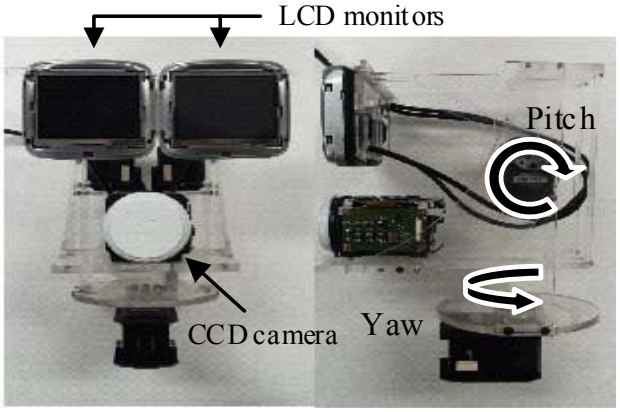


Fig. 3-5 Configurations of the robot head

For the robot head, it has two actuators, a CCD camera and 2 small LCD monitors. LCD monitors are installed that use to display the eye emotions of Conbe-I as shown in Fig. 3-5. The eye emotions are suitable for the intelligent expression of emotions. The eye movement simulator is created by the Open Graphics Library (OpenGL) software in order to express the basic emotions of the robot.

3.1.2.1 Actuators in robot head

The Conbe-I head has two degree-of-freedom robotic active head. Dynamixel RX-64 and DX-117 actuators are used to be yaw and pitch movements. For the RX-64 servo actuator is one of ROBOTIS most powerful smart actuator. It can provide a 888 oz*in of torque at 18 VDC, and it can traverse its entire 300° range in under 1 second. Each servo motor has the ability to track its speed, temperature, shaft position, voltage, and load. All of the sensor management and position control is handled by the servo motor's built-in microcontroller. The dimension of RX-64 servo actuator is shown in Fig. 3-6 and Table 3.2 illusates the specification of the RX-64 servo actuator.

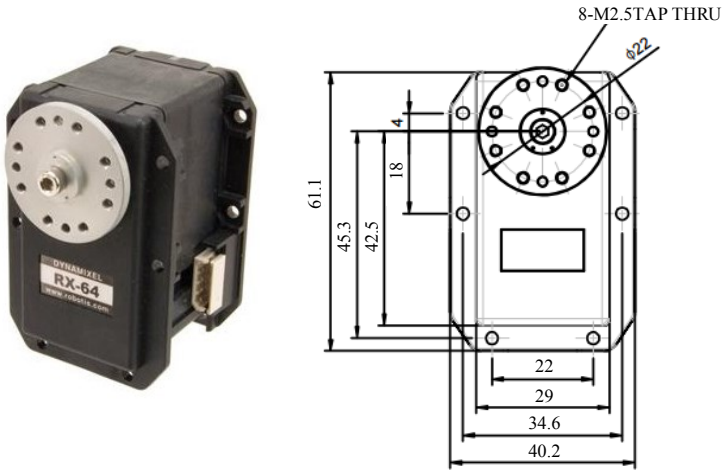


Fig. 3-6 Dimension of RX-64 motor

Table 3.2 The specifications of RX-64 servo actuator

Operating Voltage	18V	15V
Stall Torque	888 oz·in (64 kg·cm)	736 oz·in (53 kg·cm)
No-load Speed	0.162 sec/60°	0.198 sec/60°
Weight	125 g	
Size	40.2 mm x 61.1 mm x 41 mm	
Resolution	0.29°	
Reduction Ratio	1/200	
Operating Angle	300° or Continuous turn	
Max Current	1200 mA	
Standby Current	50 mA	
Operating Temperature	-5°C ~ +85°C	
Protocol	RS-485 Asynchronous serial	
Types of Feedback	Position, Temperature, Load voltage, Input voltage and Compliance/PID	
Module Limit	254 valid addresses	
Communication Speed	7343bps ~ 1Mbps	
Material	Metal Gears & Engineering Plastic Body	
Motor	Maxon RE-MAX	

3.1.2.2 FCB-H11 camera and Interface board

The SONY FCB-H11 camera is installed on the Conbe-I head, which is also used to recognize and observe the objects and external situation. The FCB-H11camera achieves a minimum illumination of 1.0 lx by dynamically removing the infrared cut filter and allowing the spectral responsivity range to extend into the near infrared. The FCB-11 incorporates a 1/3 type HD CMOS

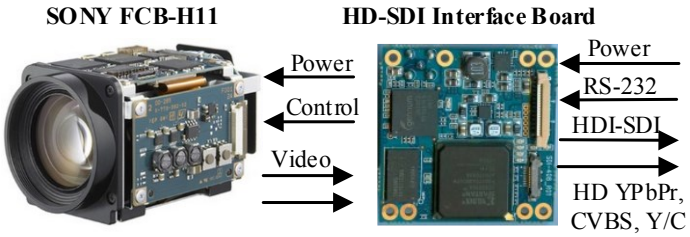


Fig. 3-7 Connection diagram for FCB-H11 camera

image sensor boasting approximately two million effective pixels and provides a 16x9 aspect ratio, making it ideal for use with wide-screen displays. The FCB-H11 inherits a multitude of functions with a 120x zooming capability picture freeze function, and slow shutter. For this system, the FCB-H11 camera is connected to the HD-SDI interface board for simplifying and operating in the image processing as shown in Fig. 3-7. And the specification of HD-SDI interface board is expressed by Table 3.3.

Table 3.3 The specifications of HD-SDI Interface Board

Video input	LVD for HD Digital video data, Analog video
Video output	HD-SDI (available only in HD mode, 720p, 1080i) HD Analog video (Bypass, available only in HD camera mode) SD Analog video (Bypass, available only in SD camera mode)
Control	RS-232, Remote Control Interface (VISCA protocol)
Power	HD-SDI Interface Module (Recommend input range : 6~12 VDC) Sony camera (Input range: 6~12 VDC, 4.8W)
Operation Conditions	0 to 45°C / 20 to 90 %RH
Dimensions	Width: 42mm, Height: 42mm, Thickness: Approx. 12.65mm (PCB 1.6mm)

3.2 Overview of software

In this section, the overview of software for the behavioral-emotional selection system based on Self-Organizing Map (SOM) and Markovian model is developed by Borland C++ Builder 5. The details of Conbe-I program are shown in Fig. 3-8 and described below.

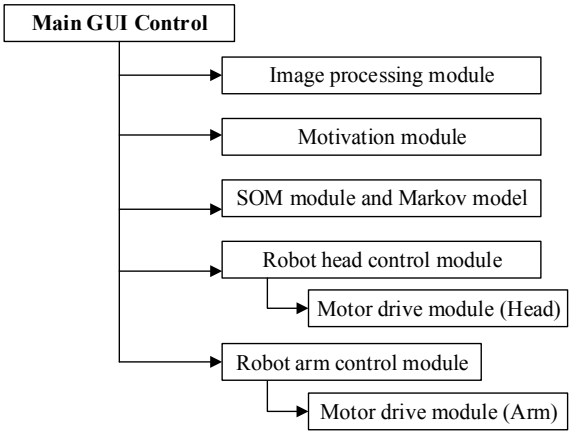


Fig. 3-8 The details of Conbe-I robot program

3.2.1 Main GUI control

Fig. 3-9 shows the Main GUI control program, which is the major program for controlling the sub-programs such as an image processing module, Motivation module, SOM module and Markov model, Robot head and arm control modules.

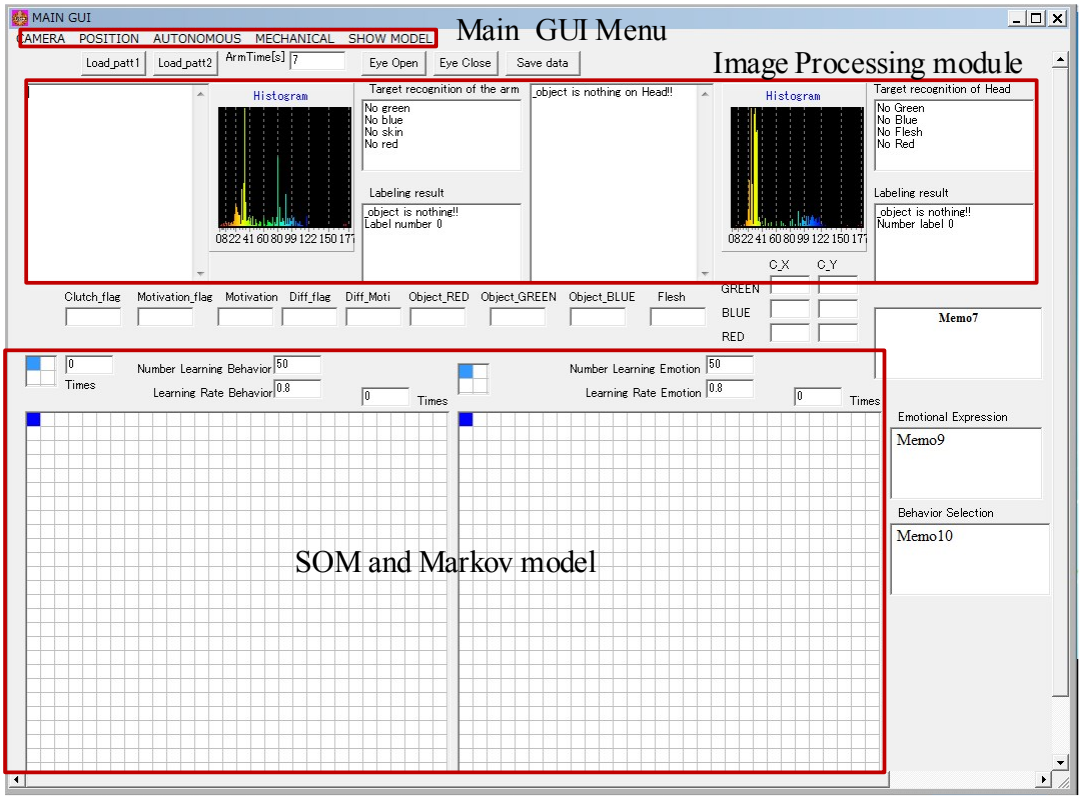


Fig. 3-9 Main GUI Control

3.2.2 Image processing module

Image processing is a form of signal processing where the input signals are images such as photographs or video frames. The output could be a transformed a version of the input image or a set of parameters related to the image. The computer revolution has taken place over the last 20 years that has led to great advancement in the field of the digital image processing. This has, in turn, opened up a multitude of applications in various fields. In this research, the images of the objects and surrounding environment are obtained using a small web camera and a CCD camera. The visual information is considered to determine the natural occurring dopamine waveform. And the example of image processing module is shown in Fig. 3-10, which consists of the capture image window for showing an image, the histogram of the image, the position of target objects and the labeling results.

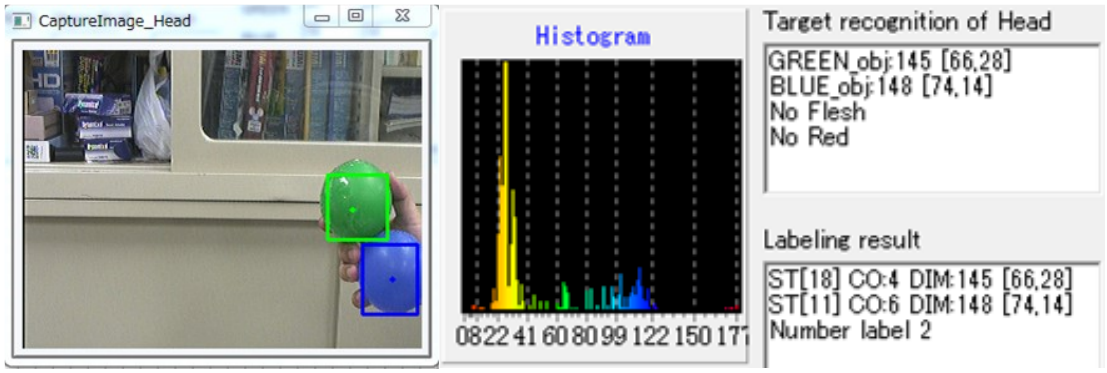


Fig. 3-10 The image processing module

3.2.3 Motivation module

The Motivation module shows the motivation of the robot, it is determined by using the total of all dopamine values (the positive and negative dopamine models). This module operates under an autonomous action. Fig. 3-11 illustrates the GUI Motivation module, the details of each part are explained as follows; 1. Start and end of the calculation of robot's motivation. 2. The motivation graph shows not only the robot's motivation, but also each dopamine waveform. In addition, the module is able to generate the typical pattern of dopamine waveforms manually by clicking the manual buttons as shown in the figure.

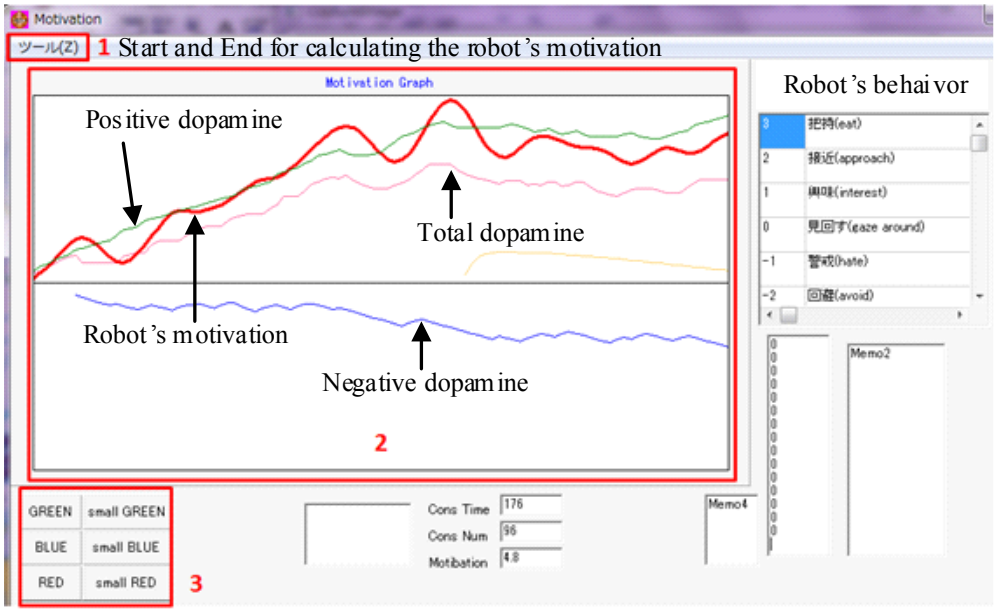


Fig. 3-11 GUI Motivation module

3.2.4 Self-Organizing Map module

For a Self-Organizing Map (SOM) module, it is used for clustering the behavior and emotions of the Conbe-I robot. The SOM technique is a type of neural network model that is can provide topologically preserved mapping from high-dimensional input data to a low-dimensional (typically two-dimensional) array. The SOM consists of an input layer and an output layer (competitive layer), which determines as an unsupervised learning. The GUI of SOM module is shown in Fig. 3-12, it has the behavior map and emotion map. This program can set the common parameters for computing the SOM learning such as the number of learning, the initial neighborhood radius, the initial learning rate, the input neurons, the output neurons and the map size. Moreover, the average quantization error is also used in this system in order to guarantee the effective mapping.

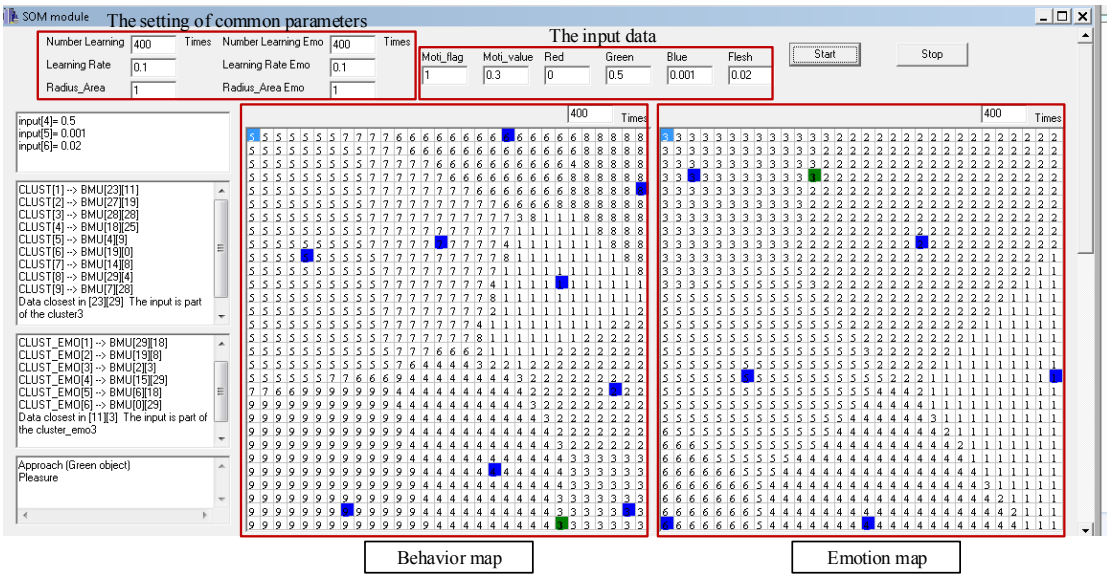


Fig. 3-12 The behavior and emotion maps in SOM module

3.2.5 Robot head and arm control modules

In Figs. 3-13 (a) and 3-13 (b) display the GUI of arm control module and GUI of head control module respectively. 1 and 8 are the open and close ports for communication between the program control and Conbe-I robot. 2 is the buttons for adjusting the degree of each joint in the arm part. 3 shows the actual angle and the estimated angle of each motor. 4 displays the position of the hand part and the target's position. 5 is created for adapting the hand's view about ±5 cm in x, y, z, directions. 6 is the group of buttons that is used for selecting the robot modes as a manual behavior mode and an autonomous behavior mode. The last group of the GUI arm control module is 7, each button is expressed the sub-behavior of the robot. For the GUI head control, 9 is the buttons that are used to adjust the yaw and pitch movements. And 10 displays the values of yaw and pitch rotations.

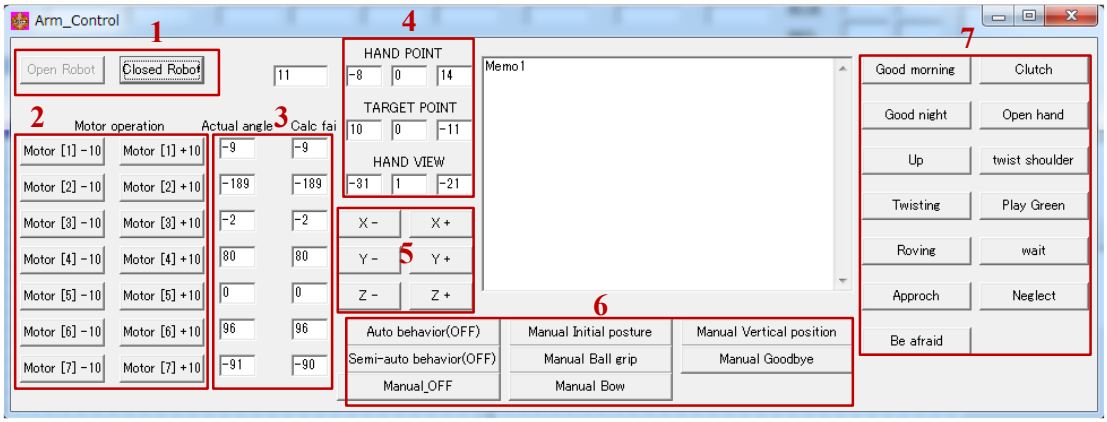


Fig. 3-13 (a) GUI of arm control module

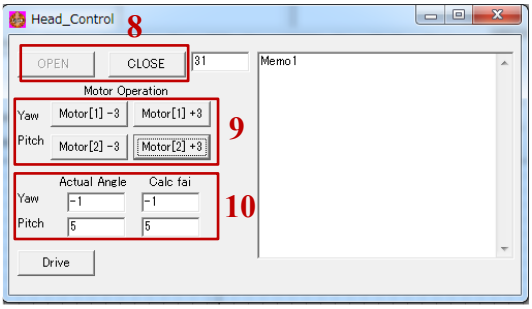


Fig. 3-13 (b) GUI of head control module

3.3 Computation of the robotic arm posture

Fig. 3-14 shows the robotic arm, which is divided into 4 parts: a shoulder, an elbow, a wrist and the fingers, because that is difficult to determine the angles of all joints from the target position by using inverse kinematics. Thus, in the research that uses the forward kinematics, which expresses using homogeneous coordinates and is able to calculate the posture from the joints of a shoulder to a wrist.

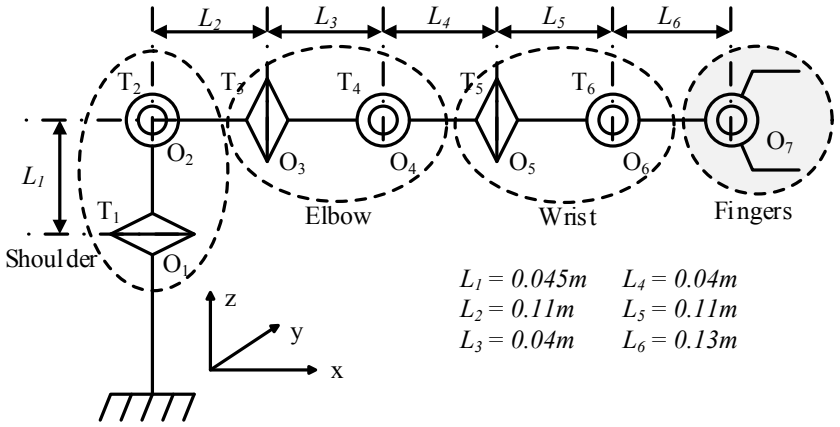


Fig. 3-14 Arrangement of degrees of freedom

3.3.1 Calculation of the position for each joint

Each joint is shown in Fig.3-14, the homogeneous transformations that described in the previous chapter is used for calculating the position of each joint, has a formation in Equation (2-49). Thus, the transformation matrices for each joint can be expressed by Equation (3-1) to Equation (3-6).

$$T_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-1}$$

$$T_2 = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 & L_2\cos\theta_2 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 & L_2\sin\theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-2}$$

$$T_3 = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & \cos\theta_3 & -\sin\theta_3 & 0 \\ 0 & \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-3}$$

$$T_4 = \begin{bmatrix} \cos\theta_4 & 0 & \sin\theta_4 & L_4\cos\theta_4 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_4 & 0 & \cos\theta_4 & L_4\sin\theta_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-4}$$

$$T_5 = \begin{bmatrix} 1 & 0 & 0 & L_5 \\ 0 & \cos\theta_5 & -\sin\theta_5 & 0 \\ 0 & \sin\theta_5 & \cos\theta_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-5}$$

$$T_6 = \begin{bmatrix} \cos\theta_6 & 0 & \sin\theta_6 & L_6\cos\theta_6 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_6 & 0 & \cos\theta_6 & L_6\sin\theta_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-6}$$

where, the position vector of the first joint is P_1, the positon vector of joint i can be considered in Equation (3-7), the local coordinate system of joint is calculate by using the inverse of the transformation matrix (T^{-1}) as shown in Equation (3-8).

$$P_i = T_{i-1}...T_4 \cdot T_3 \cdot T_2 \cdot T_1 \cdot P_1 \tag{3-7}$$

$$T^{-1} = \begin{bmatrix} R^{-1} & -p \\ 0 & 1 \end{bmatrix} \tag{3-8}$$

3.3.2 Methods of posture control

As described above, if the angle joints in the robotic arm are calculated by the inverse kinematic, that is difficult to solve the inverse kinematics problem and tend to take a long time. Therefore, in this study, the robotic arms are will be considered and determined all angle joints by dividing as the shoulder, the elbow and the wrist parts, each part that has 2 degrees of freedom. Then, the hand of the robotic arm is able to move to the target position without the inverse kinematics function. In order to create the movement patterns, the robotic arm can be divided into 3 steps according to the following sequence of step.

3.3.2.1 The wrist movement

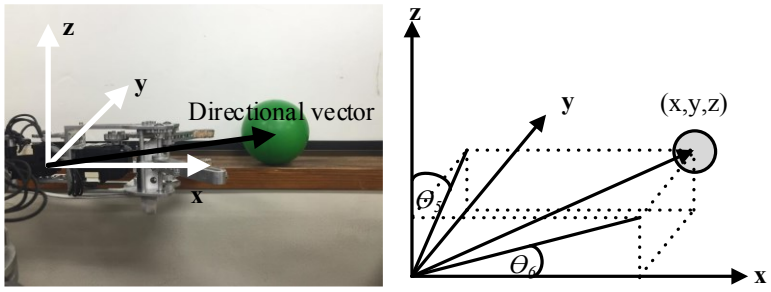


Fig. 3-15 The relationship between the robotic hand and the target position

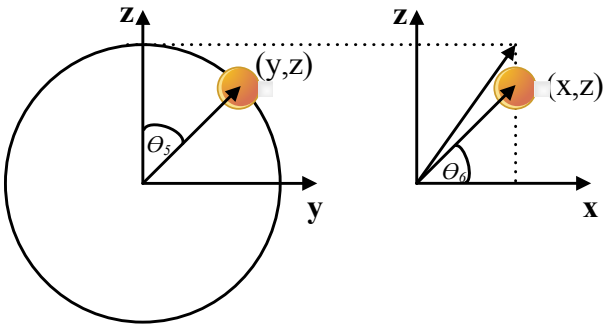


Fig. 3-16 Geometric diagram

Fig. 3-15 shows the relationship between the robotic hand and the target position. For the Fig. 3-16 illustrates Geometric diagram that can determine the local coordinates of the wrist part by Equations (3-9) and (3-10).

$$\theta_5 = \cos^{-1}\left(\frac{z}{\sqrt{y^2+z^2}}\right) \tag{3-9}$$

$$\theta_6 = \cos^{-1}\left(\frac{-x}{\sqrt{x^2+y^2+z^2}}\right) \tag{3-10}$$

By the above explanation, if there is the target object within the range of wrist movement, the robot hand can direct toward the target position at all times as shown in Fig. 3-17 (a). However, if the target position seems to be out of the range of wrist movement as illustrated in Fig. 3-17 (b), the robotic hand cannot approach to the target object, consequently the previous joints (an elbow) that are considered.

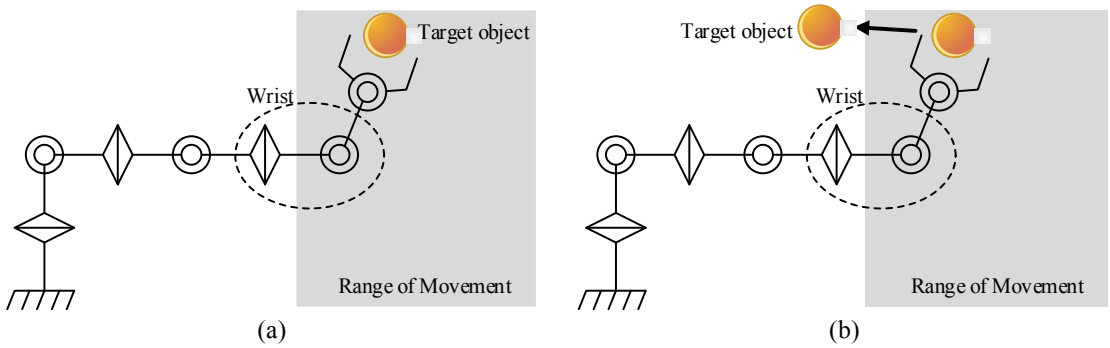


Fig. 3-17 The range of wrist movement

3.3.2.2 The elbow movement

If the target object is moved outside the operating range of the wrist movement, the robotic hand is impossible to capture the target position. The final posture is calculated using 2 DOF of the wrist part and 2 DOF of the elbow part. The elbow movement is based on pattern motions dependent on a deviation from the object, an adaptive posture can select the movement patterns based on the 9 ways of posing allowing the hand is to reach a position close to the target object. After that, the wrist movement is performed. The elbow and wrist movements are operated together as shown in Fig. 3-18.

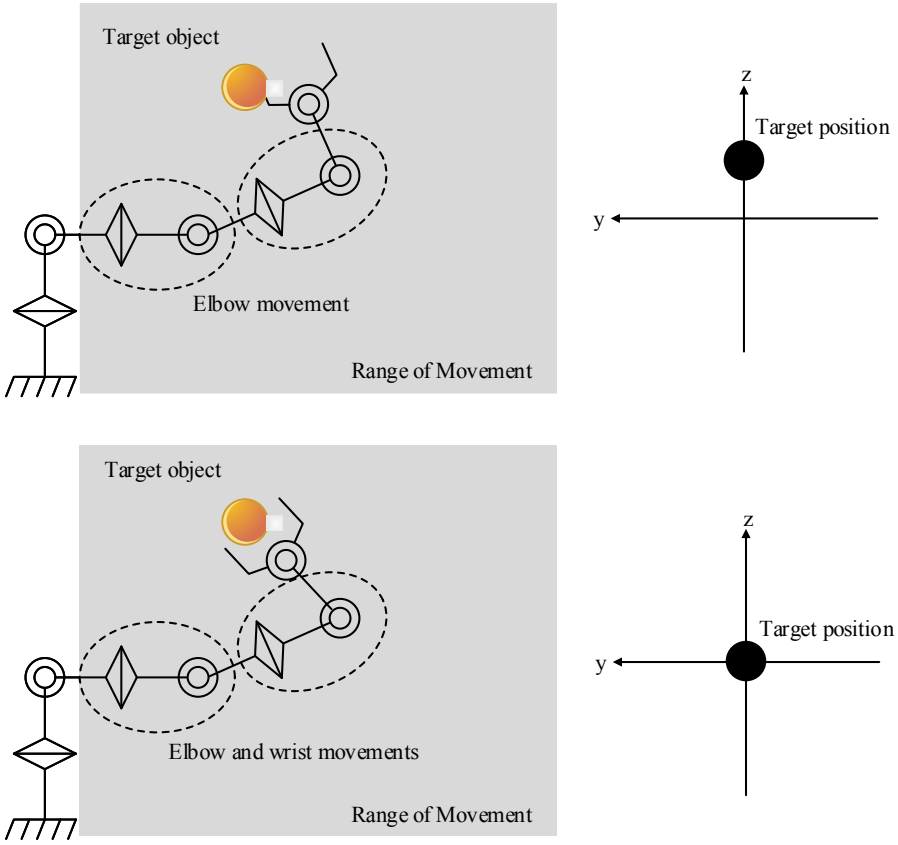


Fig. 3-18 The elbow and wrist movements

3.3.2.3 The combination of the elbow and shoulder movements

In this case, that uses 2 DOF of the shoulder part as a way of achieving the posture for gripping the target object. Therefore, the total of posing 81 ways is calculated, that is a combination of an elbow and a shoulder movements. In Fig. 3-19 (a), Fig. 3.19 (b) and Fig. 3.19 (c) show the sequential movement of the robotic arm that can continuously operate the movement, and toward to the target object without inverse kinematic solution.

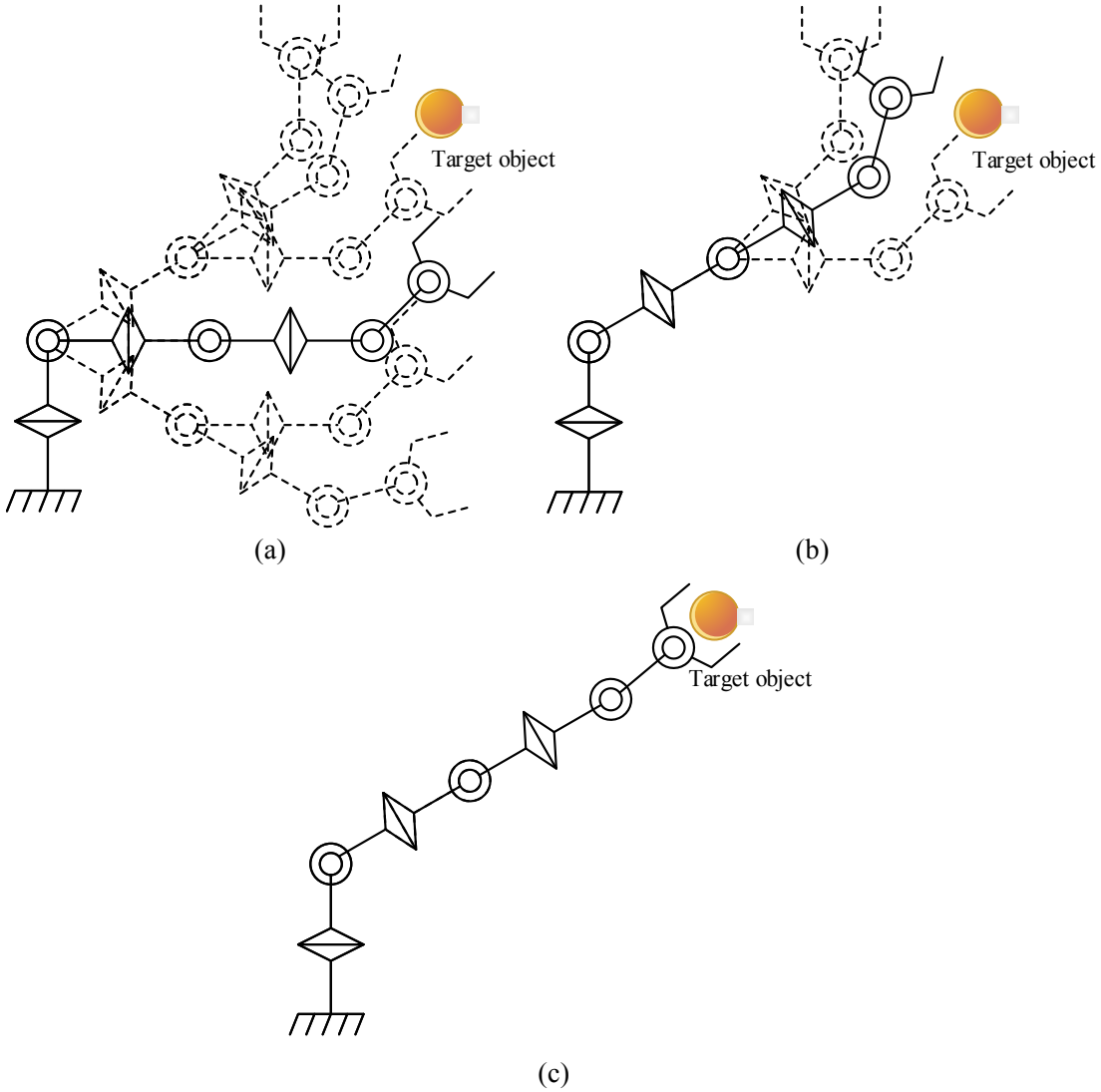


Fig. 3-19 The sequential movement of the robotic arm

3.4 Summary

In this chapter, the system structure of Conbe-I robot is described that has the system configuration and an overview of the software. The system configuration is explained such as a CCD camera and a web camera are installed for obtaining the images (target object and the external situation). Dynamixel servo motors are used at each joint in the arm part and head part. For the overview of software, Main GUI control is the main program, which can operate the sub-programs for controlling each servo motor in the robot. Finally, the computation of robotic arm posture is explained in order to perform the sequential movement.

Chapter 4

Behavioral-Emotional Selection System

Designing robot with cognition and consciousness resembling that of human beings or animals has become an important application of intelligent autonomous robot, in order to achieve a more effective human-robot interaction (HRI). In this thesis, the proposed method has been focused on considering and developing the primary structure of a conscious action, when an animal or human beings take an action that can be represented by the sequence of processes as follows: Recognition and Perception → Motivation → Behavior selection and Emotional expression.

In this chapter, the behavioral-emotional selection system is proposed that autonomously determines and outputs the most suitable behavior and emotional expression based on internal and external situations. From the primary structure as described above, the behavioral-emotional selection system can be divided into three processes. The first process, the robot recognizes the external situation and determines the robot's motivation. In the second process, the cognition module is generated and used for clustering the input stimuli (the intrinsic motivation and external situation) based on an unsupervised learning neuron network as a Self-Organizing Map (SOM), then the affective and behavior factors are calculated in order to update the elements of behavioral and emotional transition matrices. The last process is behavioral-emotional selection system. The emotion model of the proposed method is improved based on Plutchik's research. Robert Plutchik proposed the theory of emotion that is one of the most influential classification approaches for general emotional responses [49]. The emotional model consists of six basic emotions: Neutral, Hope, Happiness, Sadness, Fear and Disgust. The proposed emotional model is based on Markov modeling theory, which models emotional and dynamic states and uncertainly.

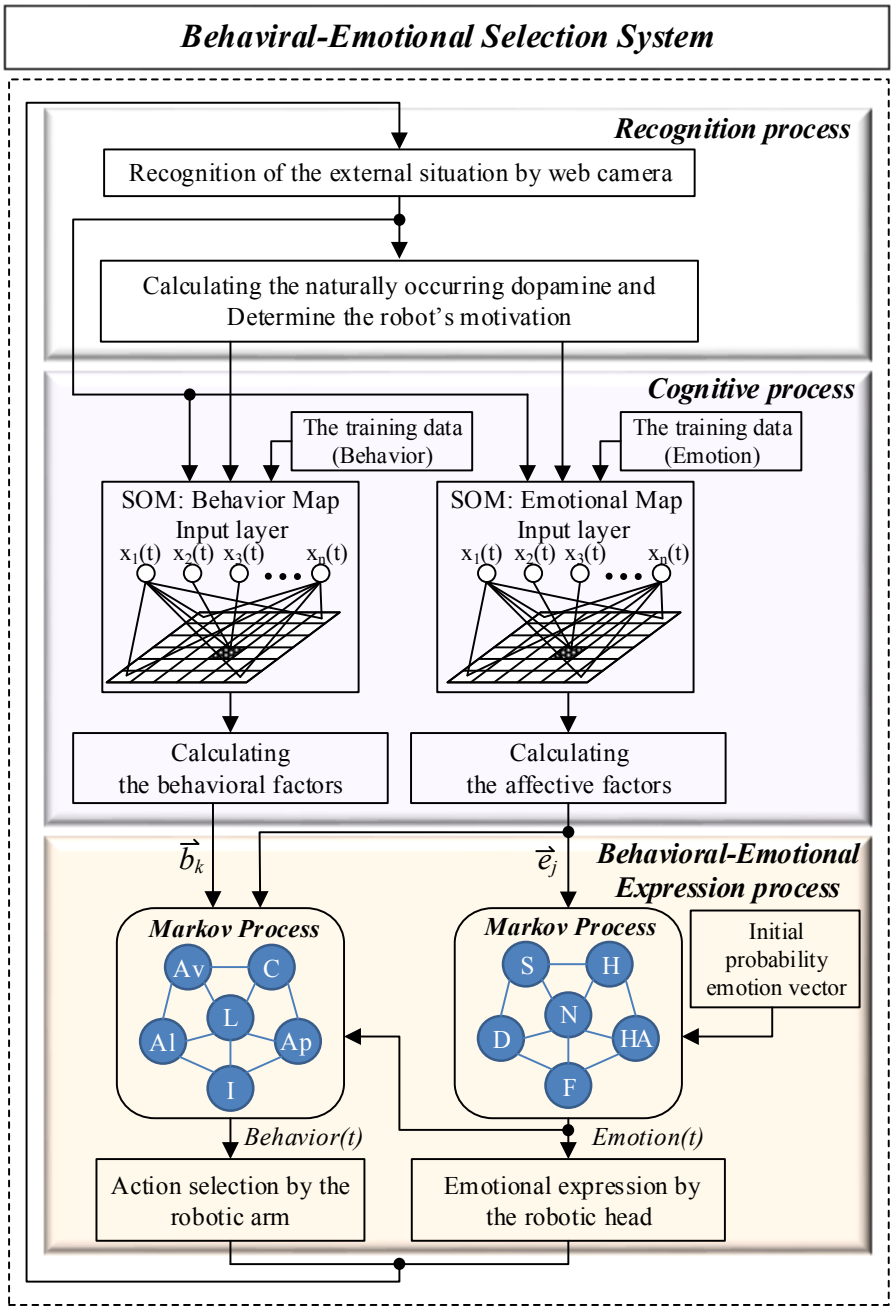


Fig. 4-1 Behavioral-emotional selection system for Conbe-I robot

The proposed method is the behavior selection system based on emotional variations as shown in Fig. 4-1. All processes are detailed below.

4.1 Recognition process

The recognition system has two fundamental parts as shown in Fig. 4-1. First part is the perception part which should recognize an external situation using the web camera and CCD camera, the visual information about the objects is corrected in terms of the shape, size, labeling and the central point of the target-color object. The second part is the calculation of the naturally occurring dopamine waveform and robot's motivation. Subsequently, the details of the recognition process will be described.

4.1.1 Recognition of the external situation

In this study, Conbe-I robot uses only the acquired images from the web camera and CCD camera for performing actions and emotional expression, the robot is not usable the other sensors such as the tactile sensor and laser range finder sensor. Thus, it is able to evaluate the rough position of the target object by without the other sensors. The simple image processing techniques for Conbe-I robot is described as follows.

4.1.1.1 The simplication of images

Typically, an important point of the robot control systems is an accurate recognition of the external environment. For example, an autonomous robots that are used in an indoor navigation task based on self-position recognition system and an obstacle recognition system by using the Laser Range Sensor (LRS) and visual methods. However, the most important in this study is to give a consciousness to our robot, is not to emphasize with high-precision formation control. Therefore, the system can simplify the acquired images from cameras by divided into 5 color groups: red, green, blue, flesh-color and the other colors, but only four colors (red, green, blue and flesh-color) that are used to recognize the target objects. And the acquired images are analyzed by using OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library.

In the simplification, it is composed of processes as:

- ***Reduce the image size***

The images obtained by the CCD camera (the robot head) and web camera (the robot arm) are read into the personal computer. However, the raw images from two cameras have the high-resolution and are difficult to process in the image processing. So, the original image size should be reduced to a lower resolution as 80x60 pixels, by using cvResize(); function and Bicubic interpolation method in OpenCV library.

- ***HSV color model***

The RGB color model is based on the theory that all visible colors can be created using the primary additive colors: red, green and blue (in the range of 0 to 255) as shown in Fig. 4-2. These colors are known as primary colors because when combined in equal amounts they produce white. But if two or three of them are combined in different amounts, other colors are produced.

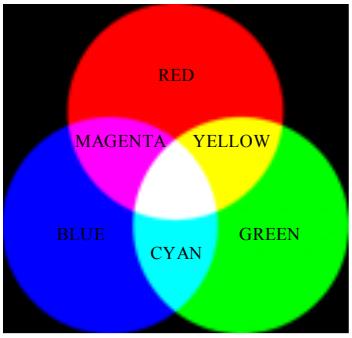


Fig. 4-2 RGB color model

The other model is HSV color model, this model defines a type of color space that is widely used to generate high quality computer graphic. It is similar to the RGB and CMYK color models. The HSV color space has three components: hue, saturation and value. Hue is expressed as a number from 0 to 360 degrees representing hues red (0°-60°), yellow (60°-120°), green(120°-180°), cyan(180°-240°), blue(240°-300°) and magenta(300°-360°). Saturation indicates the range of grey in the color space. It ranges from 0% to 100% or sometime the value is calculated from 0 to 1. A faded color is due to a lower saturation level, which means the color contains more grey. Value (or

Brightness) works in conjunction with saturation and describes the brightness or intensity of the color from 0% to 100%. When the value is '0' the color space will be totally black color. If the increase in the value, the color space brightness up and shows various colors. The HSV color model is illustrated in Fig. 4-3. Each component (Hue, Saturation and Value) can be determined from the RGB color model by the simple flowchart as shown in Fig. 4-4.

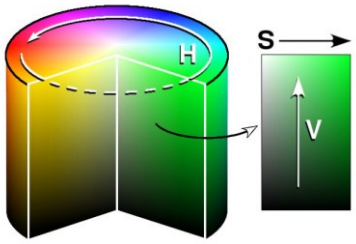


Fig. 4-3 HSV color model

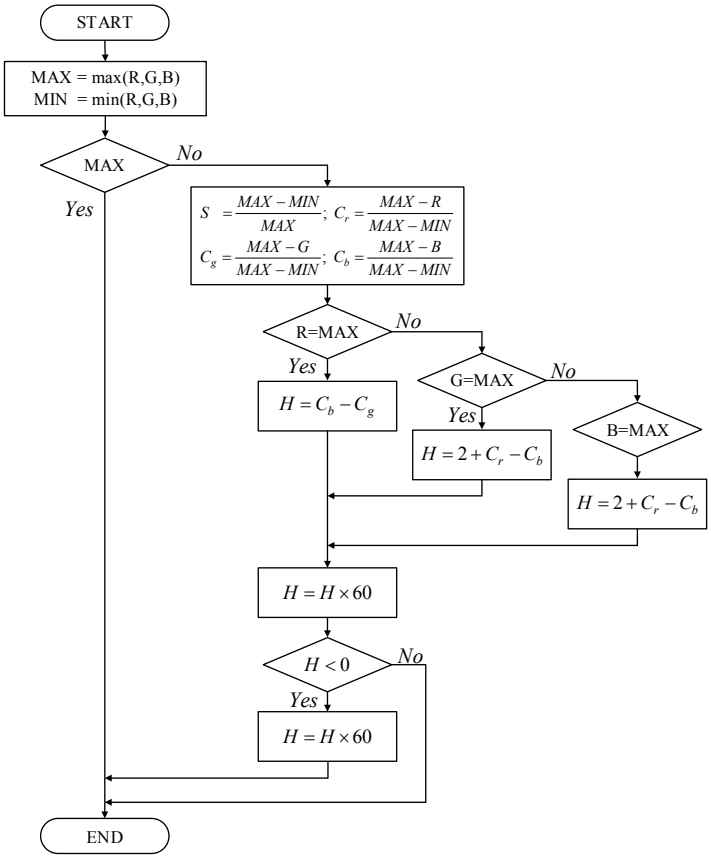


Fig. 4-4 RGB-to-HSV color algorithm

- ***Specification of images in HSV color model***

From the algorithm as shown in Fig. 4-4, the visual information of image (80x60 pixels) can be converted from the RGB color model to HSV color model. The range of each component (Hue, Saturation and Value) used to recognize the target object and human as shown in Fig. 4-5 and the threshold values of each color are defined in Table 4.1.

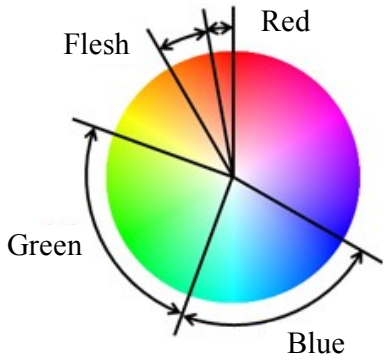


Fig. 4-5 The range of each component in HSV color model

Table 4.1 The threshold values for RGB-to-HSV color method

	Red	Flesh-color	Green	Blue
Hue [°]	0~10	10~30	70~160	160~240
Saturation [%]	59	10	18	39
Value [%]	20	20	20	39

4.1.1.2 Labeling process and landmark recognition

- ***Labeling process based on the color of visual information***

Typically, in order to extract specific features of the objects from the image, it is necessary to perform a segmentation process to original image. Therefore, an object labeling algorithm which is used for labeling the distinct objects from a binary (black and white) image is presented. This algorithm is useful for the separation of distinct objects for further analyses applied to each

individual object, it is possible to recognize the target object. Fig. 4-6 shows the simplified image and labeling image.

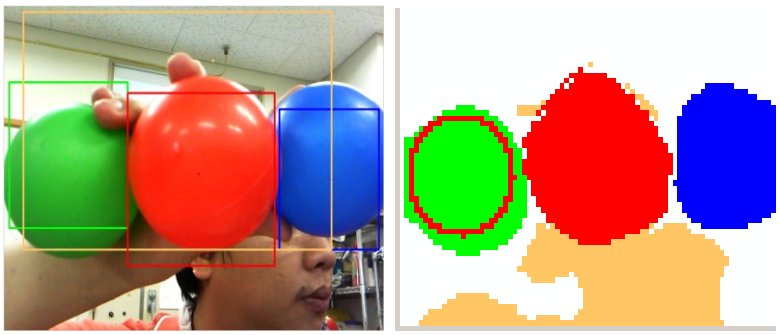


Fig. 4-6 Simplified image and labeling image

- ***Position recognition of target object***

The geometric center coordinates of the obtained color information that will be used to calculate approximately the position of the target object. However, it is very difficult to evaluate the depth perception using the camera. Consequently, in order to recognize the image obtained from the camera, the perspective projection plane is created for determining the position of the target object as described in Fig. 4-7.

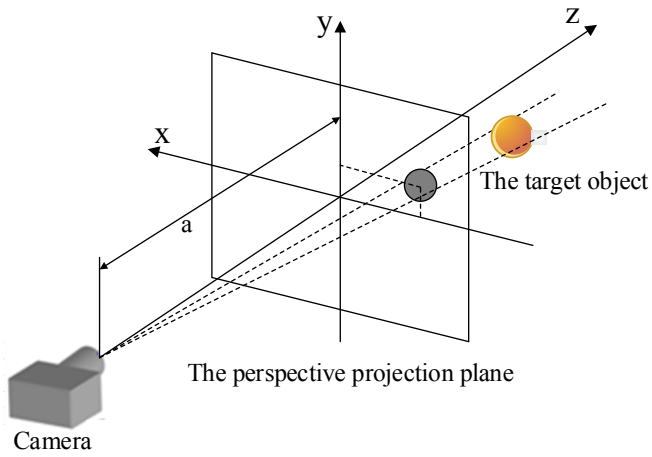


Fig. 4-7 Estimation of a target position

- ***Shape recognition***

However, only the color recognition is not enough to perceive the surrounding environment. Thus, the shape recognition method is considered and used in the recognition process. This method is the drawing circle from the results of the labelling process. Fig. 4-8 shows the example of the details of each element, how to determine the object frame.

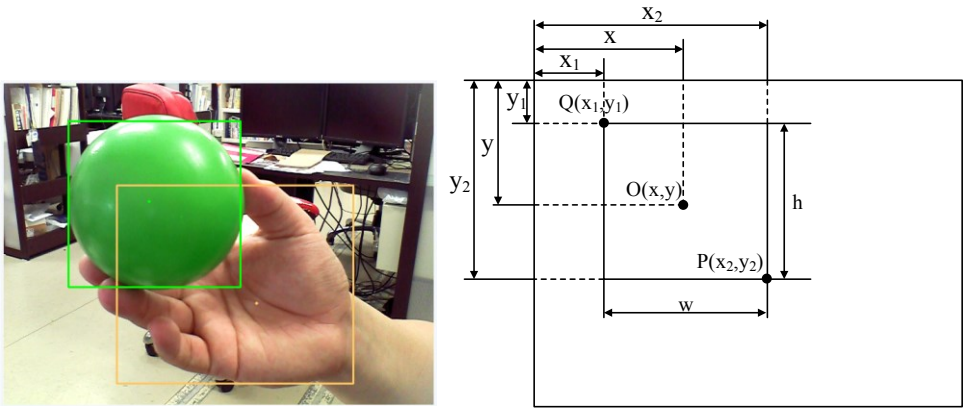


Fig. 4-8 The target object (green ball) frame

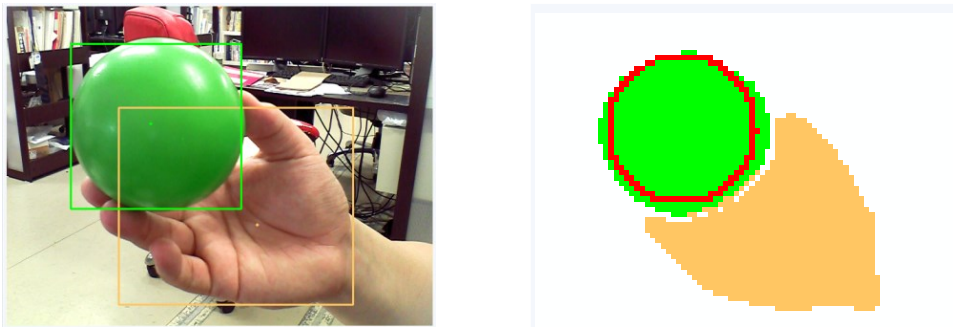


Fig. 4-9 The result of the recognition process

First, the results of the original image from labeling process are set the bounding rectangle. $Q(x_1,y_1)$ and $P(x_2,y_2)$ coordinates are used to calculate the size of a rectangle (height and width) and then O(x,y) is determined as the center point. Next step, the radius of the object is calculated in order to draw the circle by comparing the edge of the object frame as shown in Fig. 4-9.

4.1.1.3 The desired settings of the object for Conbe-I

When, the Conbe-I robot performs an autonomous behavior, it should recognize the important components of the object. The images are simplified by dividing into four color groups: red, green, blue and flesh-color that are distinguished and perception in terms of the shape, size, center-of-gravity position. In this study, the liking behavior is performed when the robot is able to recognize the red or green objects. On the other hand, the robot should perform disliking behavior if it faces of recognizes the blue object. And the sample color objects are shown in Fig. 4-10.

Fig. 4-10 The sample color objects. (Blue, Green and Red)

4.1.2 Computation of the naturally occurring dopamine

Next step, the visual information (the color, shape, size, labeling and distance of the target object) defines as the input data in order to determine and form the naturally occurring dopamine waveform. When animals and human beings take various actions, the dopamine is secreted in the brain [50]. Dopamine is a neurotransmitter, one of those chemicals that is responsible for transmitting signals between the nerve cells (neurons) of the brain, and plays vital roles in a variety of different behaviors. The major behavior of dopamine affects a movement, cognition, pleasure and motivation. Therefore, in this section, the computation of naturally occurring dopamine is explained.

4.1.2.1 A drug involves levels of the naturally occurring dopamine

In order to generate the typical pattern of dopamine secretion for determining the robot's motivation. The pattern of dopamine model is referred based on changing of amounts of drugs. For example, the p-chloroamphetamine is injected into the rat, the concentration is about 0.02 molar, and the dopamine releases in the rat's brain as shown in Fig. 4-11. Therefore, the waveform in the dotted box will be considered for generating the typical pattern of dopamine waveform in the next section.

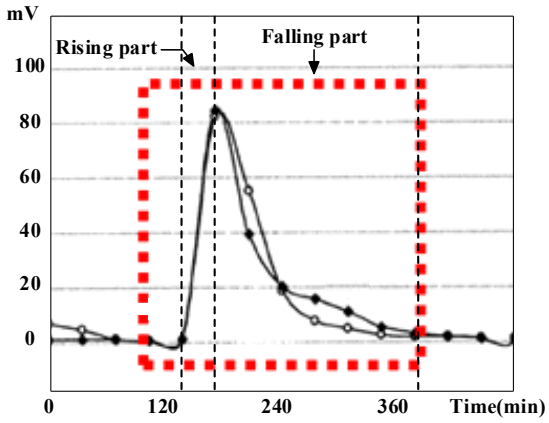


Fig. 4-11 The sample of dopamine waveform in the rat's brain

4.1.2.2 Imitation dopamine waveform

As mentioned above, the waveform of naturally occurring dopamine can be evaluated and divided into rise and fall portions as shown in Fig. 4-12.

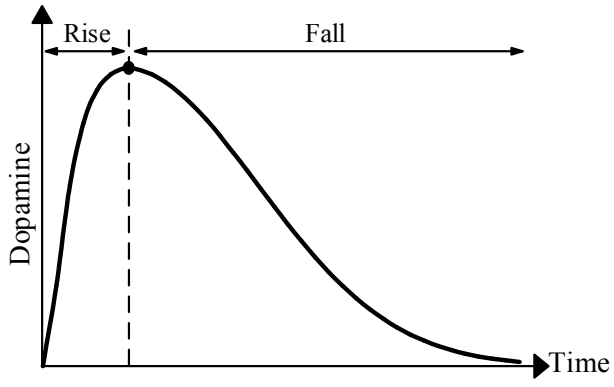


Fig. 4-12 Occurrence of dopamine model

- ***The rising part of dopamine waveform***

In order to create the rising part of dopamine waveform, the 2nd order system is used for calculating by Equation (4-1), where the input variable $x(t)$ is an accelerator of dopamine, the output variable $y(t)$ is a naturally occurring dopamine, ζ is the damping factor and ω_n is the natural frequency.

$$y''(t) + 2\zeta\omega_n y'(t) + \omega_n^2 y(t) = \omega_n^2 x(t) \tag{4-1}$$

At initial step (t=0), and the input variable is a unit step as $x(t) = 1$ and Equation (4-1) is able to be verified by depending on the values of damping factor as follows.

Case 1 ($\zeta < 1$)

From Equation (4-1) that is modified as Equation (4-2).

$$y(t) = 1 - e^{-\zeta\omega_n t}\left(\cos\sqrt{1-\zeta^2}\,\omega_n t + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin\sqrt{1-\zeta^2}\,\omega_n t\right) \tag{4-2}$$

Case 2 ($\zeta = 1$)

From Equation (4-1) that is modified as Equation (4-3).

$$y(t) = 1 - e^{-\omega_n t}\left(1 + \omega_n t\right) \tag{4-3}$$

Case 3 ($\zeta > 1$)

From Equation (4-1) that is modified as Equation (4-4).

$$y(t) = 1 - e^{-\omega_n t}\left(\cosh\sqrt{\zeta^2-1}\,\omega_n t + \frac{\zeta}{\sqrt{\zeta^2-1}}\sinh\sqrt{\zeta^2-1}\,\omega_n t\right) \tag{4-4}$$

Moreover, the time of peak (T_p) for $\zeta < 1$ and $\zeta \geq 1$ can be expressed as Equations (4-5) and (4-6) respectively.

$$T_{p,\zeta<1} = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}} \tag{4-5}$$

$$T_{p,\zeta\geq 1} = \frac{2\pi}{\omega_n} \tag{4-6}$$

- ***The falling part of dopamine waveform***

For the falling part of dopamine's waveform process, the peak value of the rising part (y_{peak}) defines as the initial input variable for determining by using the exponential equation as expressed in Equation (4-7), where $y(t)$ is the output of dopamine waveform, T_c is the time constant.

$$y(t) = e^{-t/T_c} \cdot y_{peak} \tag{4-7}$$

4.1.2.3 Controlling the amount of dopamine waveform

In this section, the stimulus variables are described for controlling the amount of dopamine's waveform such as the natural angular frequency (ω_n), the damping factor (ζ) and the time constant (T_c).

The first stimulus variable is the natural angular frequency (ω_n) that affects the speed of the rising part in the occurrence of dopamine model. The next stimulus variable is the damping factor (ζ), it has effect the peak value of the dopamine's waveform, and the last one is the time constant (T_c) it influences the decay of the falling part of dopamine's waveform. Therefore, in Fig. (4-13), Fig. (4-14) and Fig. (4-15) that show the waveforms of dopamine model when the stimulus variables (ω_n , ζ and T_c) are changed respectively.

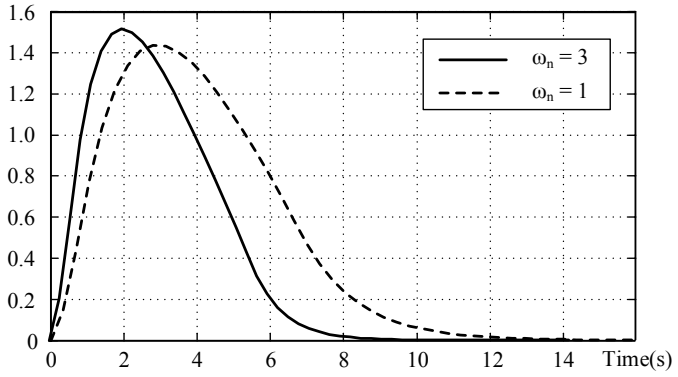


Fig. 4-13 Dopamine's waveform when ω_n is changed

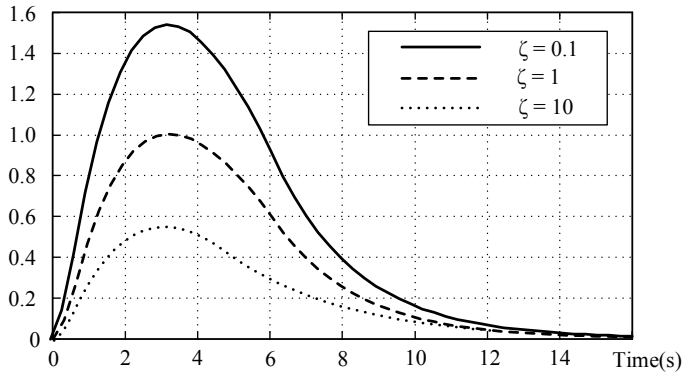


Fig. 4-14 Dopamine's waveform when ζ is changed

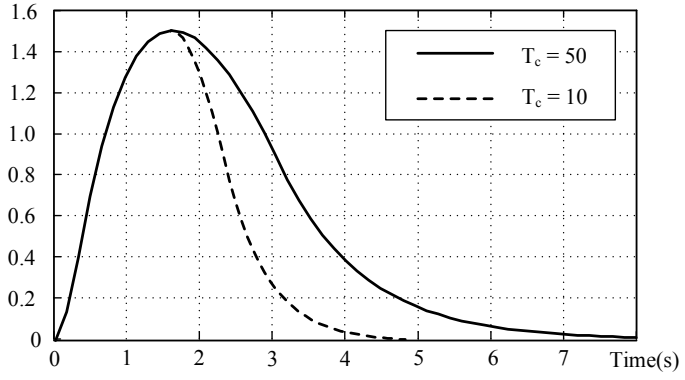


Fig. 4-15 Dopamine's waveform when T_c is changed

As described above, that is possible to set and generate the naturally occurring dopamine by controlling the stimulus variables according to the external situation.

4.1.2.4 Derivation of the feeling distance

The feeling distance (between the target object and the camera) is the important variable that is used for developing the robot's motivation. Therefore, this section will explain how to calculate the feeling distance. Suppose, the green ball (a favorite object) is recognized by a web camera. And the object's distance is changed from 0 cm to 50 cm. The result of the relationship between the number of pixels and the feeling distance is shown in Fig. 4-16.

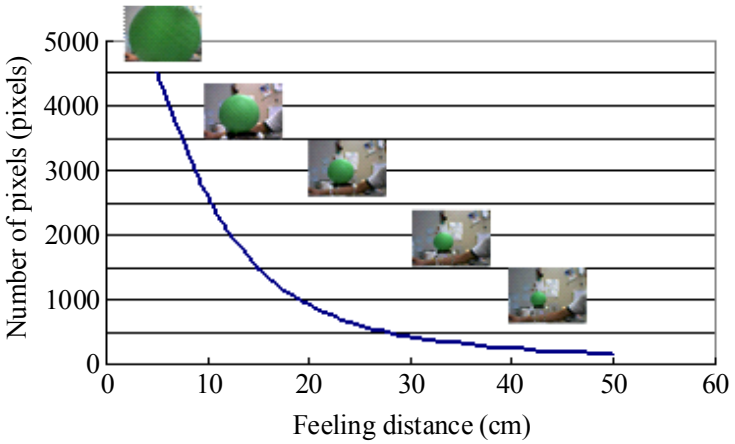


Fig. 4-16 Relationship between the number of pixels and feeling distance

In Fig. 4-16, the result of waveform seems as the exponential function, thus the relationship between the feeling distance and number of pixels can be expressed by Equations (4-8) and (4-9).

$$Dist = \frac{\log\left(Pixel/3983.0\right)}{-0.0682} ; if\ 0 < Pixel < 1500 \tag{4-8}$$

$$Dist = \frac{\log\left(Pixel/7796.0\right)}{-0.1099} ; if\ 1500 < Pixel < 4800 \tag{4-9}$$

The feeling distance will be used as the input variable for calculating naturally occurring dopamine and determining the stimulus variables.

4.1.2.5 Determination of the stimulus variables

In this section, the stimulus variables are specified by the conditions of the liking object (green object) and disliking object (blue object) recognitions.

- ***Condition of the liking object (green object)***

Here, the setting parameters are assigned in the conditions of the favorite object when the robot is able to recognize the liking object. In this situation can divide into 4 conditions for generating the dopamine's waveform as:

- If the robot can recognize the green object at the first time.

The movable range of the robot arm and the feeling distance are used to set the stimulus variables as expressed by Equations (4-10), (4-11) and (4-12), where $Feeling\,dist$ is the feeling distance of the robot and $Movement\,dist$ is the movable range of the robot arm.

$$\omega_n = 10.0 \tag{4-10}$$

$$\zeta = 0.1 + (Feeling\,dist/10) - (5Movement\,dist) \tag{4-11}$$

$$T_c = 60.0 + 60.0\{(50 - Feeling\,dist)/100 + Movement\,dist\} \tag{4-12}$$

- If the distance between the green object and the robot's hand has changed.

In this case, when the distance is changed between previous time and current time, which interprets as the shrinking of the dopamine's waveform or the expanded waveform. $diff\ vallue$ is the variable that presents the different value of the feeling distance as expressed by Equation (4-13), where $Feeling\,dist\,back$ is the feeling distance at the previous time and $Feeling\,dist$ is the feeling distance at the current time.

$$diff\ value = e^{\left\{(Feeling\,dist\,back - Feeling\,dist)/50.0\right\}} \tag{4-13}$$

Therefore, in this case, the setting parameters are calculated by Equations (4-14), (4-15) and (4-16).

$$\omega_n = 10.0 \tag{4-14}$$

$$\zeta = {30.0}/{diff\ value} \tag{4-15}$$

$$T_c = diff\ value \tag{4-16}$$

- If the green object is unmoved.

The stationary state is defined by the center of gravity point is not changed. In this state, the stimulus variables will be assigned as $\omega_n = 20.0, \zeta = 15.0$ and $T_c = 0.05$. But if the green object is not the same position, T_c will be increased in order to decrease the dopamine level dramatically correspond to the Equation (4-16).

- If the green object is a ball

In this situation, that is similarly the previous condition (if the green object is unmoved), it means the robot can recognize the green ball for a long time, the dopamine is continuously increasing. And the setting parameters is also $\omega_n = 20.0, \zeta = 15.0$ and $T_c = 0.05$.

- ***Condition of the disliking object (blue object)***

The other condition is described when the robot recognizes the blue object (disliking object). In this case, it can divide as 2 conditions for generating the dopamine's waveform:

- If the robot can recognize the blue object.

In this study, the robot should perform disliking behavior or negative emotion when it can recognize the blue object. And in this situation, the robot doesn't need to consider the movable range of the robot. The dopamine's waveform is represented as the negative value and the all setting parameters are also indicated by Equations (4-14), (4-15) and (4-16).

- If the feeling distance of green object and blue object are different

In this case that describes about the recognition of the green and blue objects at the same time. Equations (4-8) and (4-9) are used again for calculating the feeling distance of blue object ($Dist\,Blue$), in order to determine the ratio of the feeling distance between the green object and blue object ($Ratio\,of\,GtoB$) as illustrated in Equation (4-17).

$$Ratio\,of\,GtoB = \frac{Dist\,Green}{Dist\,Blue} \tag{4-17}$$

If $Ratio\,of\,GtoB \geq 1$ that means the blue object is near the camera, the negative dopamine is increasingly created. The setting parameters are set by $\omega_n = 20.0, \zeta = 15.0$ and $T_c = 0.05$. On the other hand ($Ratio\,of\,GtoB < 1$), the time constant will be modified by multiplying with the ratio of the feeling distance as expressed in Equation (4-18).

$$T_c = T_c \cdot Ratio\,of\,GtoB \tag{4-18}$$

4.1.3 Calculation the intrinsic robot's motivation

From the computation of the naturally occurring dopamine model as described above, the total sum of their positive (the green object) and negative (the blue object) values that is used as the input variable for calculating the robot's motivation shown in Fig. 4-17 and the motivation waveform is estimated by the 2nd order system of linear differential equation as expressed by Equation (4-19).

$$\begin{aligned} Robot's\,motivation(t) = {} & 2 \cdot Total\,of\,dopamine(t) \\ & - \frac{1}{\omega_n{}^2} \frac{d^2}{dt^2} Robot's\,motivation(t) \\ & - 2\frac{\zeta}{\omega_n{}^2} \frac{d}{dt} Robot's\,motivation(t) \end{aligned} \tag{4-19}$$

Here, $Total\ of\ dopamine(t)$ is the total of the naturally occurring dopamine that is described in the above section, $Robot's\ motivation(t)$ is the output variable, ω_n and ζ are considered by the outside environment and the internal state.

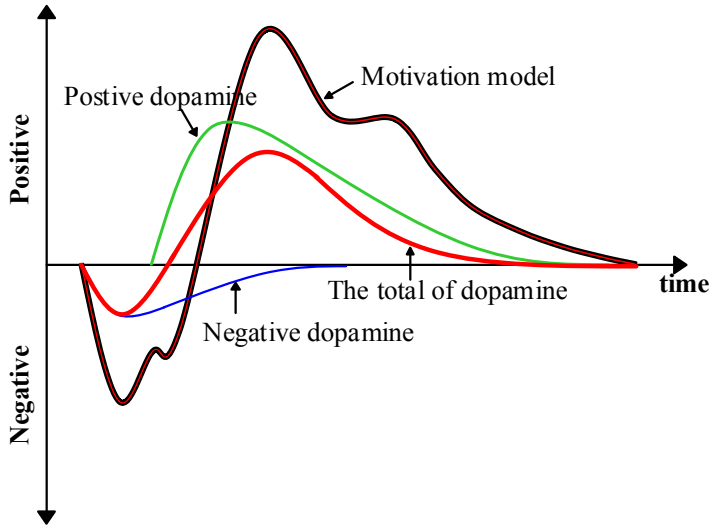


Fig. 4-17 Robot's motivation model

4.2 Cognitive process

After recognition process, the robot's motivation and the visual information are used to specify the input parameters of the Cognitive model which analyses by a Self-Organizing Map (SOM) learning in order to generate the behavior and emotion maps. In this process, the nine behaviors and six emotional expressions are set up for the Conbe-I robot as shown in Table 4.2.

Table 4.2 The behavioral and emotional states of Conbe-I robot

Liking behavior	Look around (green object), Interest (green object), Approach (green object), Catch (green object) and Interest (flesh-color)
Disliking behavior	Look around (blue object), Be alert (blue object), Avoid (blue object) and Be alert (flesh-color)
Emotion expressions	Neutral, Hope and Happiness (Positive emotion) Sadness, Fear and Disgust (Negative emotion)

In order to classify and select the most appropriate behavior and emotion correspond to the surrounding environment of the robot, the results of the SOM learning will be verified.

4.2.1 Formation of behavior and emotion maps by SOM

The theory of the SOM learning is explained in Chapter 2 that can summarize in this section. The Self-Organizing Map (SOM) is a neural network architecture. It is well known as an effective pattern classifier that uses the unsupervised learning method. The SOM is composed of map units called nodes or neurons, which connect to adjacent neurons by a neighborhood relation that provided topology-preserving mapping from high-dimensional input data onto a low-dimensional array of neurons (usually two dimensional), as illustrated in Fig. 4-18.

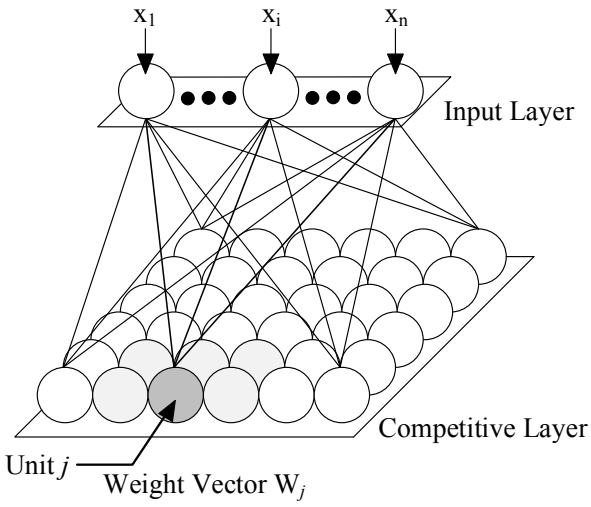


Fig. 4-18 Structure of the SOM model

Each neuron in the output map (the competitive layer) is associated with a vector representing the "weight vector", which are updated toward a center of gravity of the input vectors weighted by neighborhood function. And the procedure of the basic SOM learning can explain by the following steps:

Step 1. Each node must be initialized with the weight value. Typically the weights will be set to small standardized random value. The value of $w_j(0)$ are initialized in [0, 1].

Step 2. At each training step, the Euclidean distance at node j on the map between the weight vector and input vector is calculated as expressed by Equation (4-20), where x_i is the value of parameter i in the input vector, $w_{ij}(t)$ is the weight value between parameter i and node j on the SOM, and N is the number of parameters.

$$d_j(t) = \sqrt{\sum_{i=0}^{N-1}\left(x_i - w_{ij}(t)\right)^2} \tag{4-20}$$

Step 3. The best-matching unit (BMU), whose weight vector is closest to the input vector (denoted here by c), is determined, as shown in Equation (4-21).

$$c = \arg\min_j \{d_j\} \tag{4-21}$$

Step 4. After finding the BMU, the weight vectors of the BMU, and its topological neighbors are moved closer to the input vector. The new weight vectors are updated by Equation (4-22), where t is the training step index, $\eta(t)$ is the learning rate, and $h_{j,c}(t)$ defines as the neighborhood kernel function around the winner nodes; there are expressed by Equations (4-23) and (4-24), respectively. All variables that are described in Chapter 2.

$$w_j(t+1) = w_j(t) + \eta(t)h_{j,c}(t)\left[\bar{x} - \bar{w}_j(t)\right] \tag{4-22}$$

$$\eta(t) = \eta_0 \exp\left[-t/\tau\right] \tag{4-23}$$

$$h_{j,c}(t) = \exp\left[-\left\|r_j - r_c\right\|^2 / 2\sigma(t)^2\right] \tag{4-24}$$

Step 5. The training steps increase to $t+1$. Steps 2 to 4 are repeated with decreasing the width of the neighborhood function.

To verify the experimental results in the SOM learning, the data normalization is necessary for preparing the input data. The neural network training could be made more efficient by performing certain preprocessing steps in the networks input and target. The normalization process for the raw inputs has great effect on preparing the data to be suitable for training. Without this normalization, training the neural networks would be very slow. There are many types of data normalization. It can be used to scale the data in the same range of values for each input feature. Data normalization can also speed up the training time by starting the training process for each feature within the same scale. It is particularly useful for modeling applications where the inputs are generally at widely different scales. Different techniques can use different rules such as Statistical normalization, Min-Max normalization, Median normalization, Sigmoid normalization, Mean, Standard Deviation normalization and so on. In this study, the Min-Max normalization is used, which can calculate by the following Equation (4-25), where x_i is the original data values (the motivation value, pixels of green color, blue color and flesh color), $x_{\min}$ and $x_{\max}$ are the minimum and maximum of feature X.

$$x_{i,new} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \cdot \left(x_{\max,new} - x_{\min,new} \right) + x_{\min,new} \tag{4-25}$$

This function is used to scale into a new interval [0, 1], so define $x_{\min,new} = 0$ and $x_{\max,new} = 1$ in the above Equation (4-25). Now the simplified formula of normalization is shown in Equation (4-26). Each feature will lie within the new range of values and remain the same. Min-Max normalization has the advantage of preserving exactly all relationships in the data.

$$x_{i,0to1} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{4-26}$$

For instance, the motivation values, the slope of motivation values, the pixels of colors are normalized using Equation (4-26), and they are defined as the details of input data for testing the SOM learning as illustrated in Table 4.3. And Table 4.4 shows the parameters of SOM learning.

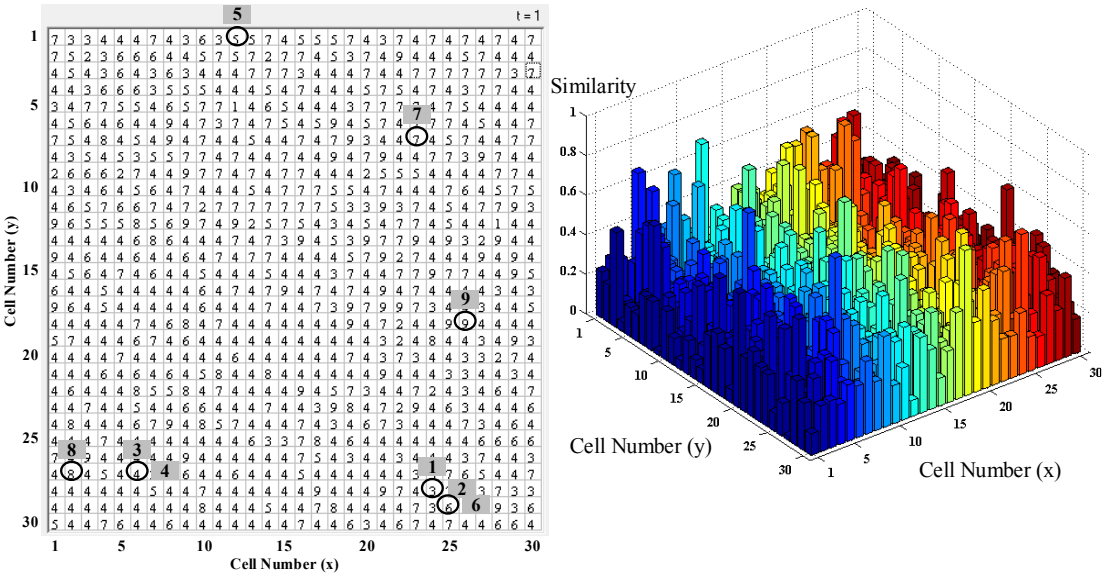
Table 4.3 The details of the input data for SOM learning

	Value	Description
Moti_flag	True or False	The positive motivation value is True = 1, and the negative value is False = 0.
Moti_value	0 to 1	The motivation value
Moti_slope_flag	True or False	The positive slope of motivation value is True = 1, the negative slope is False = 0.
Moti_slope_value	0 to 1	The slope of motivation value
Green	0 to 1	Pixels of green color
Blue	0 to 1	Pixels of blue color
Flesh	0 to 1	Pixels of flesh color

Table 4.4 The parameters of SOM learning

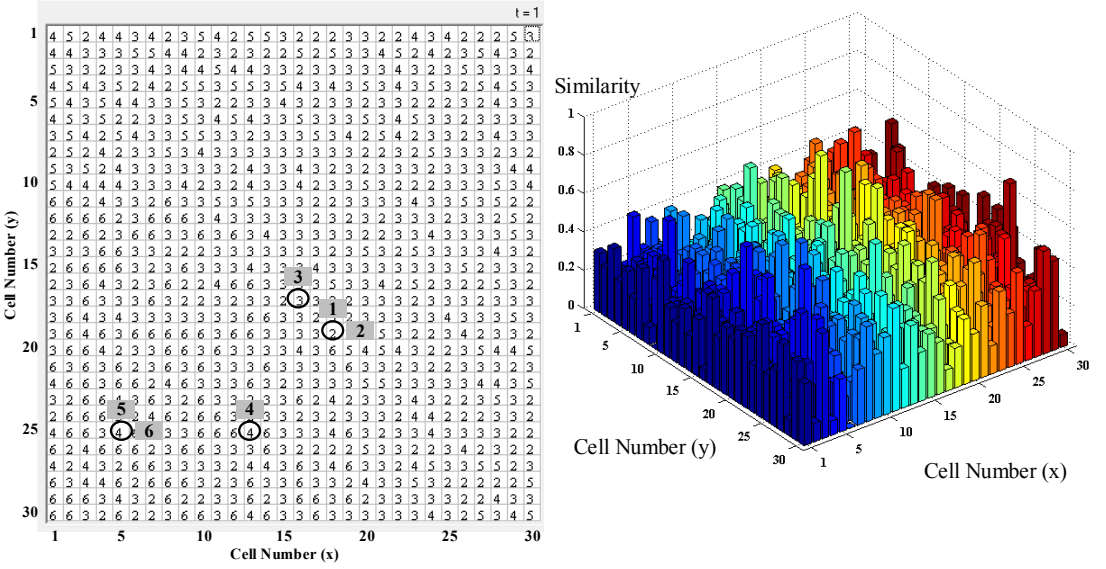
	Value	Description
T	200	The number of learning
η_0	0.08	The initial learning rate
σ_0	15	The initial neighborhood radius
Map size	30 x 30	The rectangular SOM topology size

For example, if the robot recognized a green object and the target object was near the robot's hand, the testing input data were the motivation flag = 1 (positive level), the motivation value = 0.45, the slope of motivation value = 0.003, the motivation waveform is a positive slope (Moti_slope_flag = 1), the green pixels = 0.6, the blue pixels = 0.02 and the flesh-color pixels = 0.2. During the learning of SOM process, we haven't only observed the changing position of the winner nodes of each output pattern in the behavior and emotion, but also considered the evolution of the form of both maps as shown from Figs. 4-19 to Figs. 4-26. For the input patterns (Behavior and emotion patterns) are shown in Appendix A.



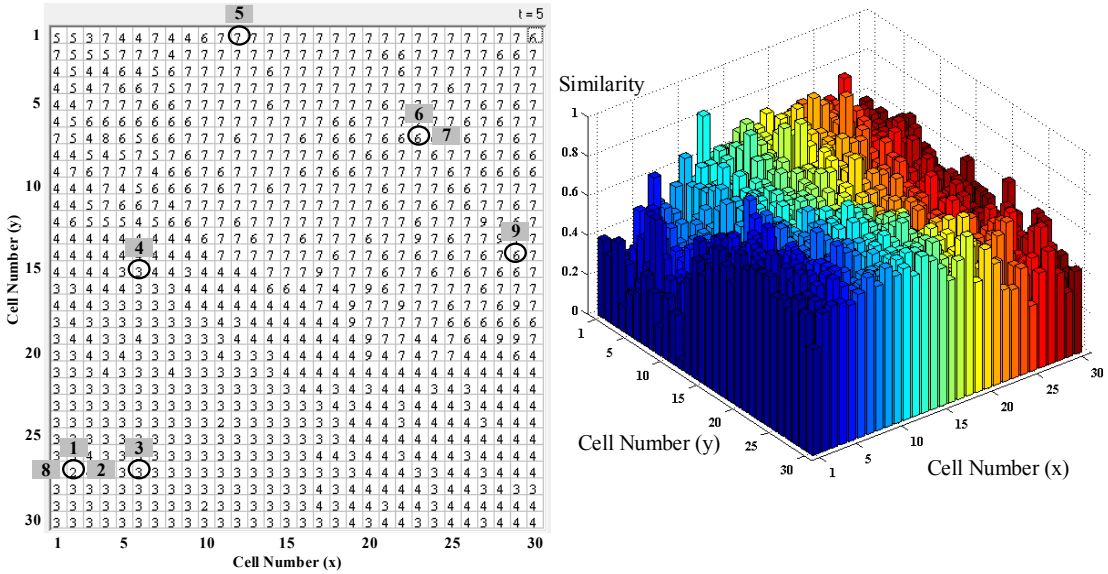
(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-19 (a) During SOM learning in behavior map (t = 1)



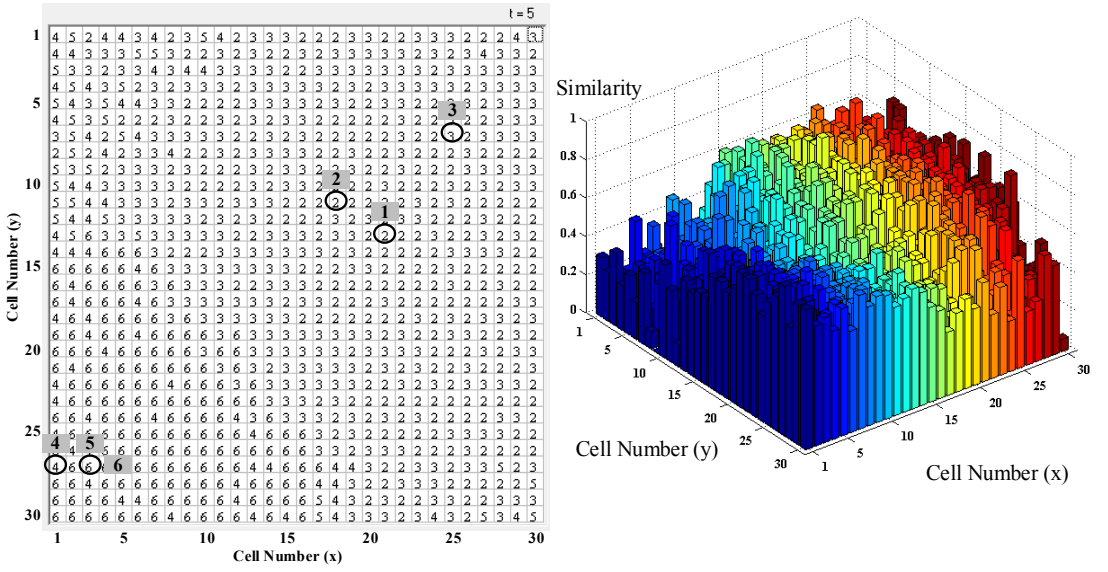
(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-19 (b) During SOM learning in emotion map (t = 1)



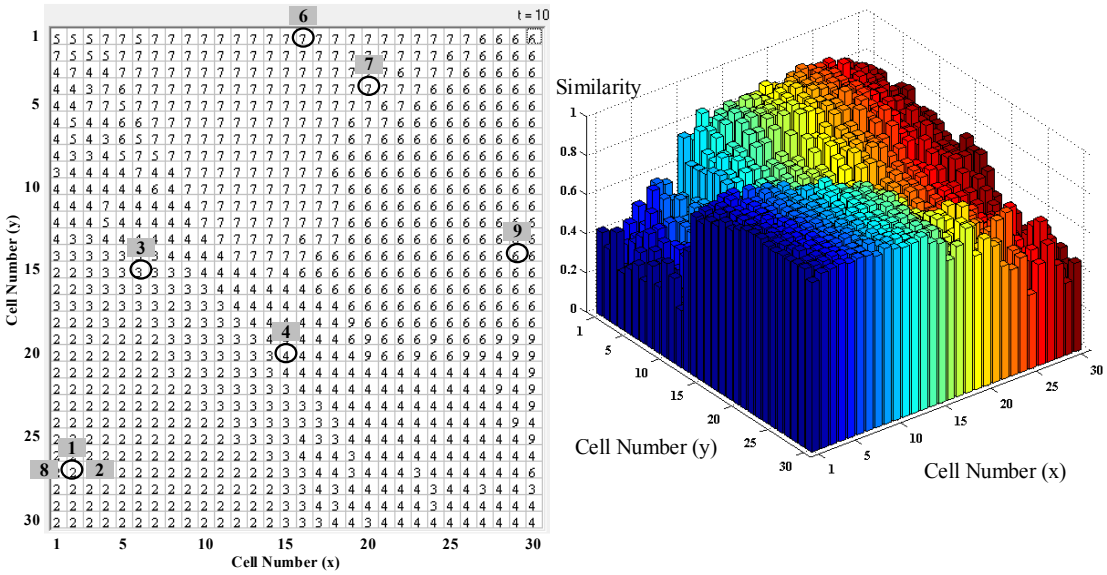
(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-20 (a) During SOM learning in behavior map (t = 5)



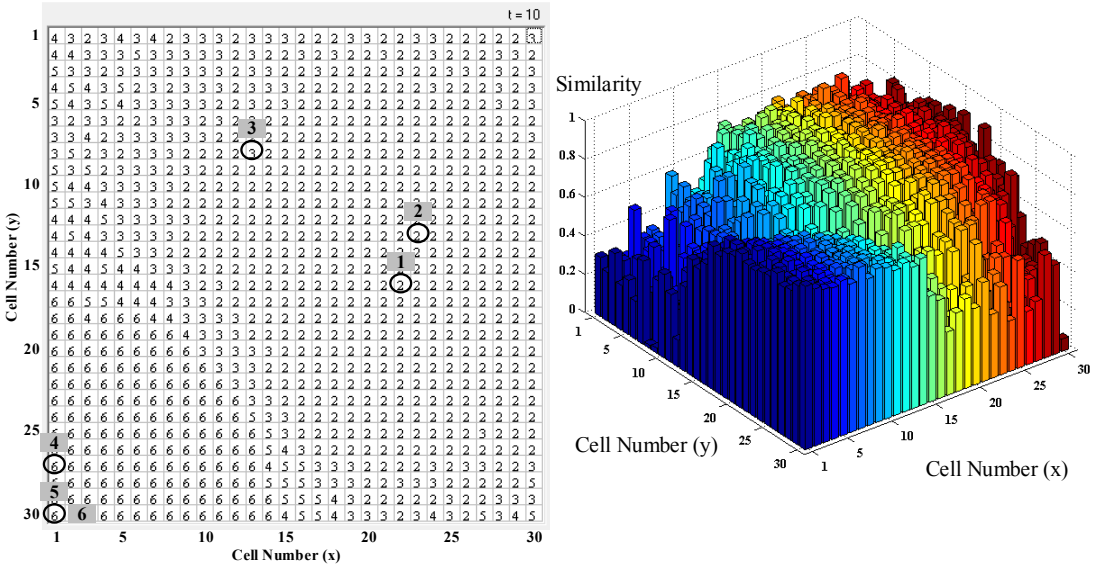
(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-20 (b) During SOM learning in emotion map (t = 5)



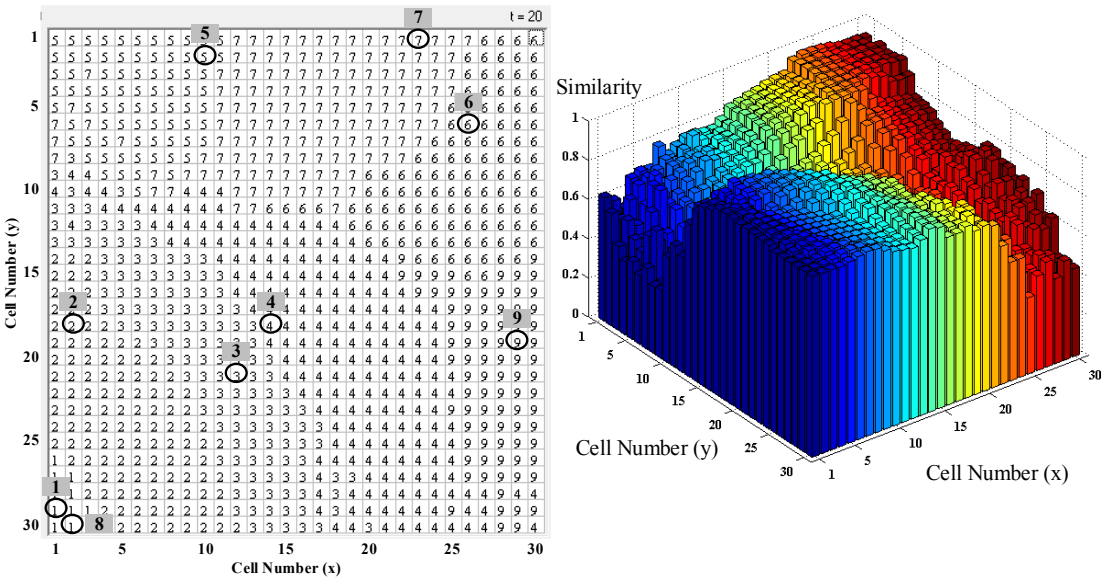
(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-21 (a) During SOM learning in behavior map (t = 10)



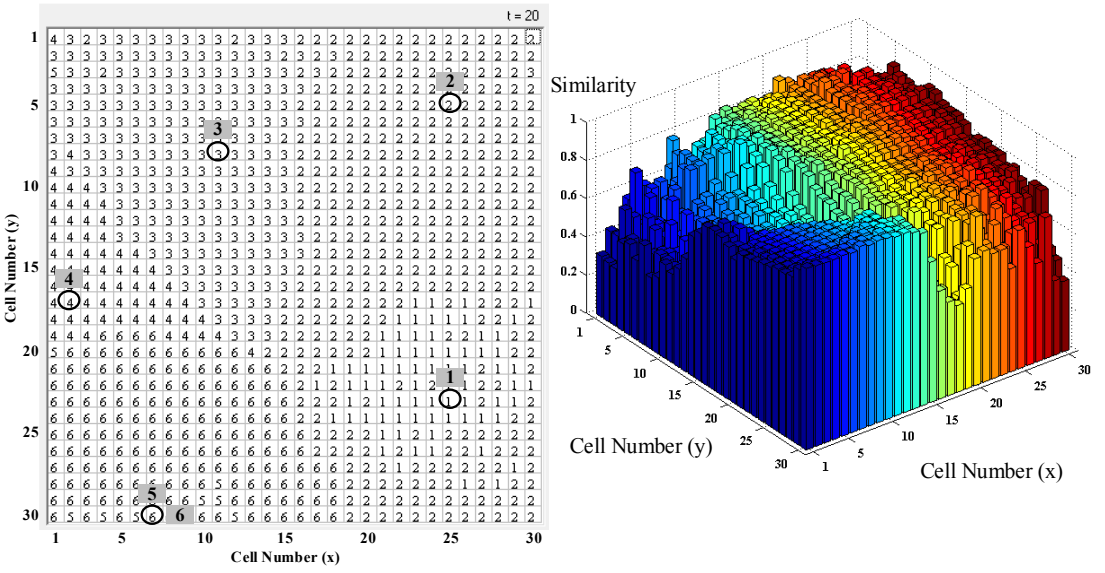
(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-21 (b) During SOM learning in emotion map (t = 10)



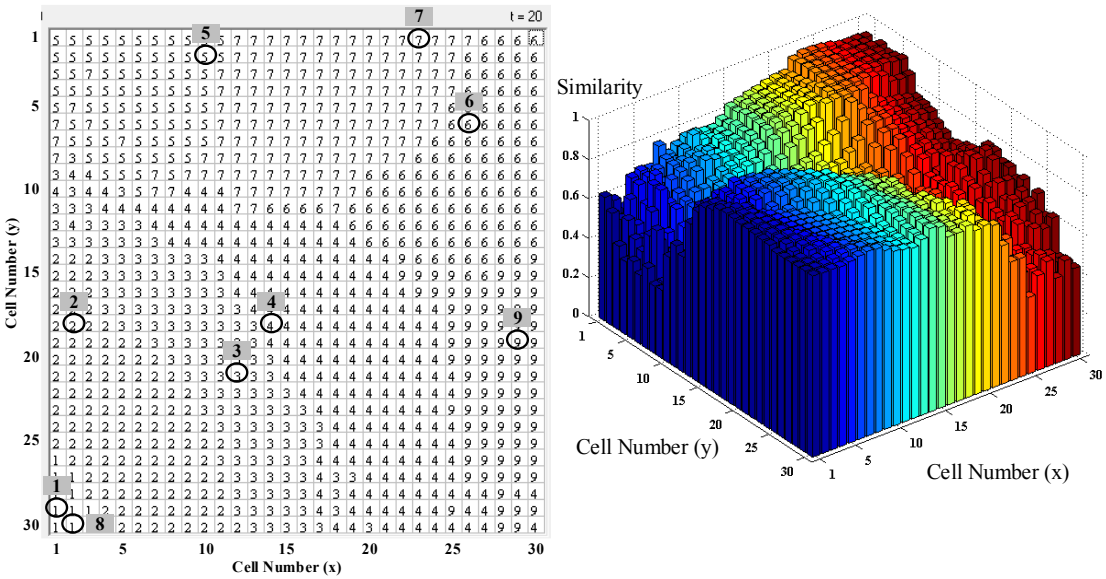
(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-22 (a) During SOM learning in behavior map (t = 20)



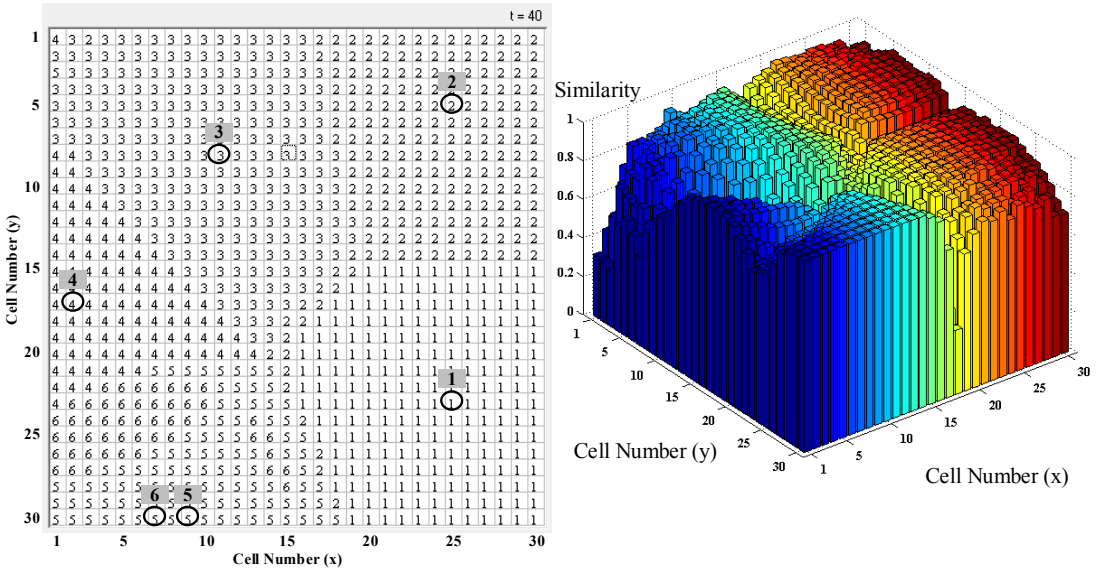
(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-22 (b) During SOM learning in emotion map (t = 20)



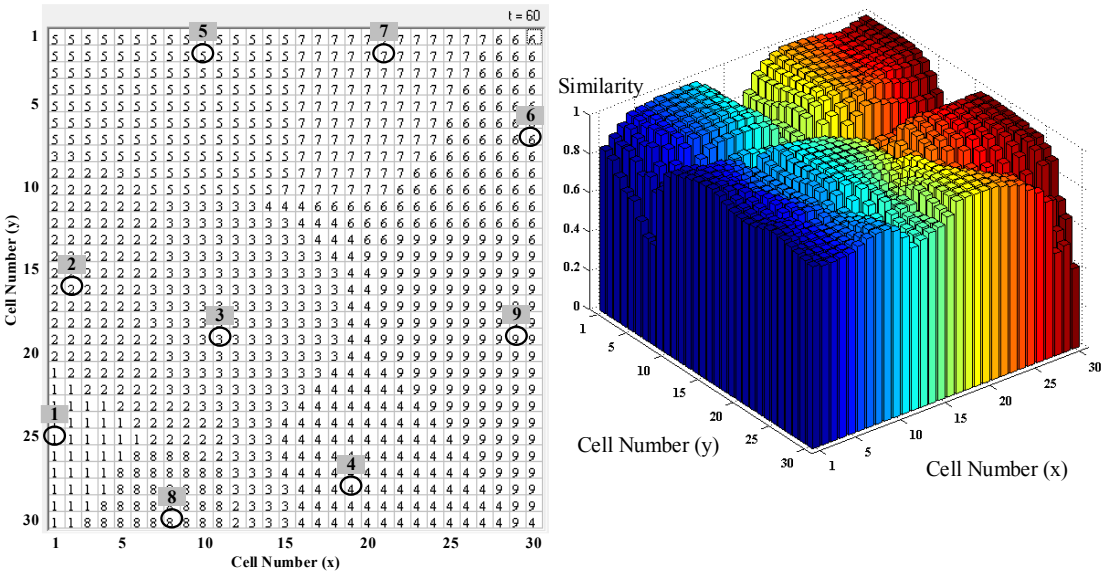
(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-23 (a) During SOM learning in behavior map (t = 40)



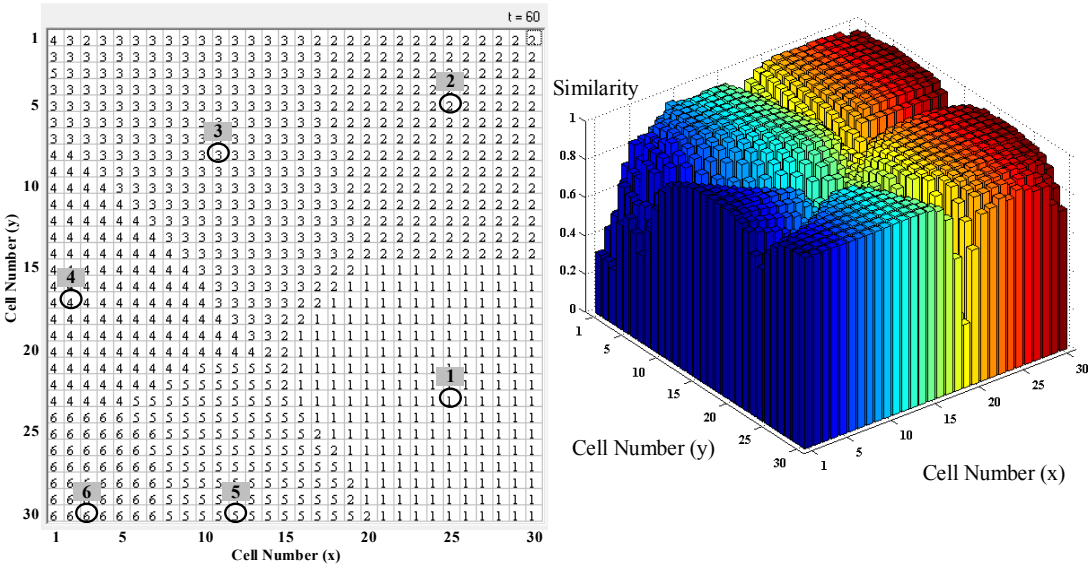
(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-23 (b) During SOM learning in emotion map (t = 40)



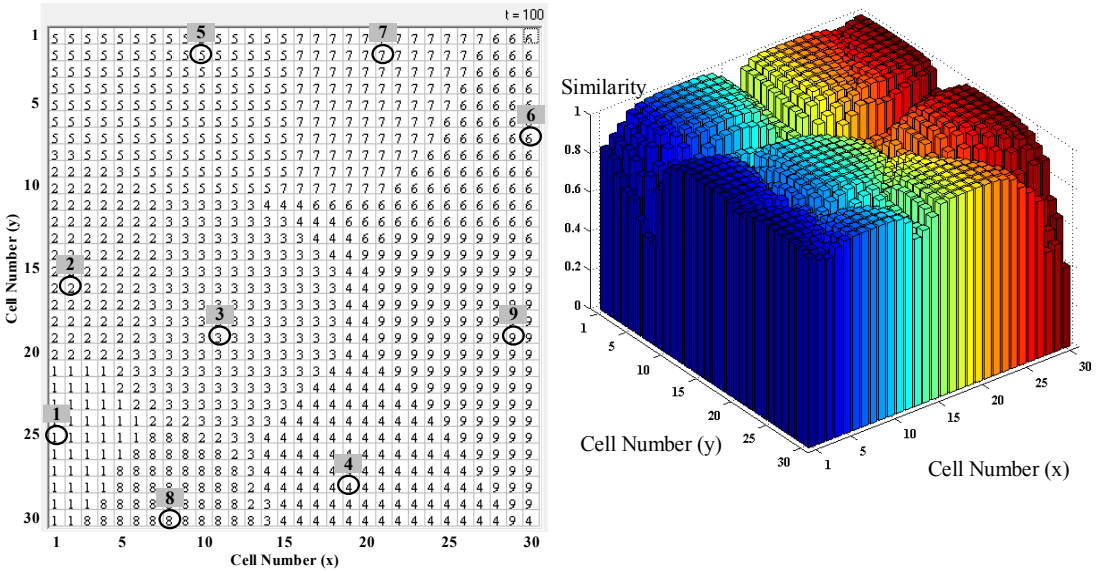
(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-24 (a) During SOM learning in behavior map (t = 60)



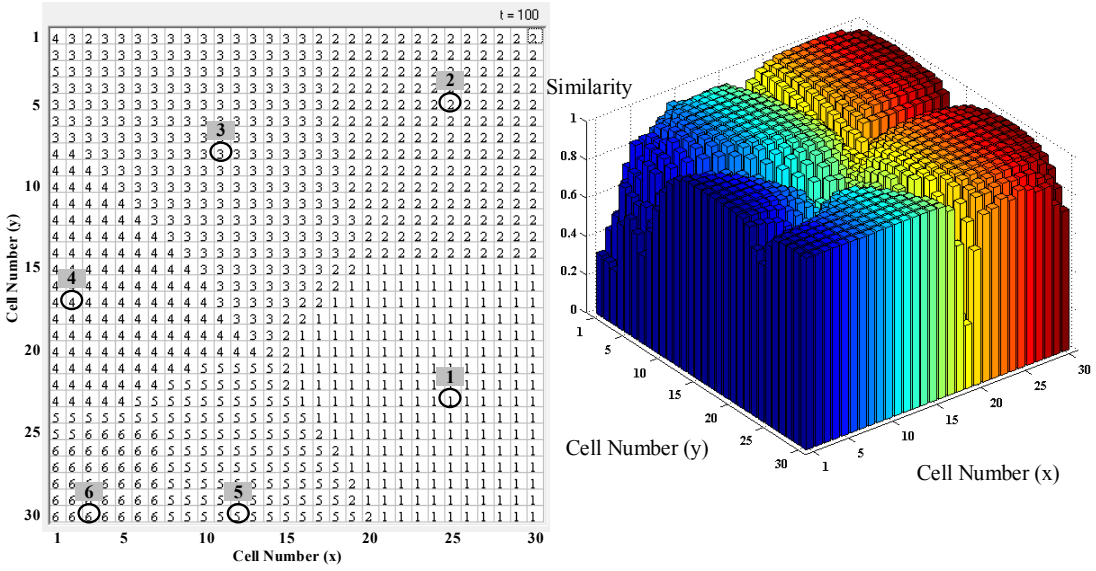
(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-24 (b) During SOM learning in emotion map (t = 60)



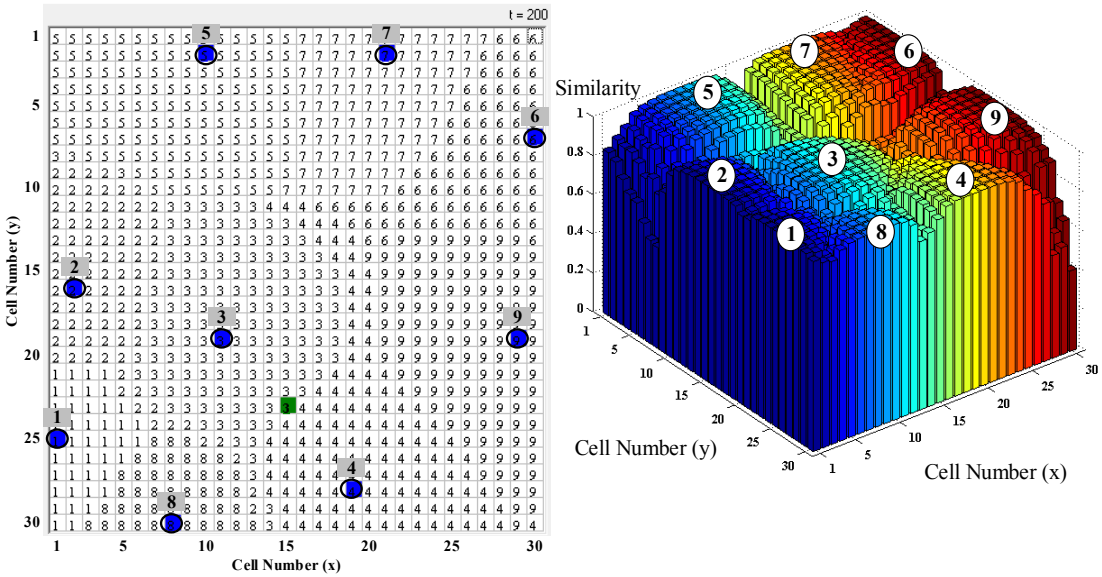
(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-25 (a) During SOM learning in behavior map (t = 100)



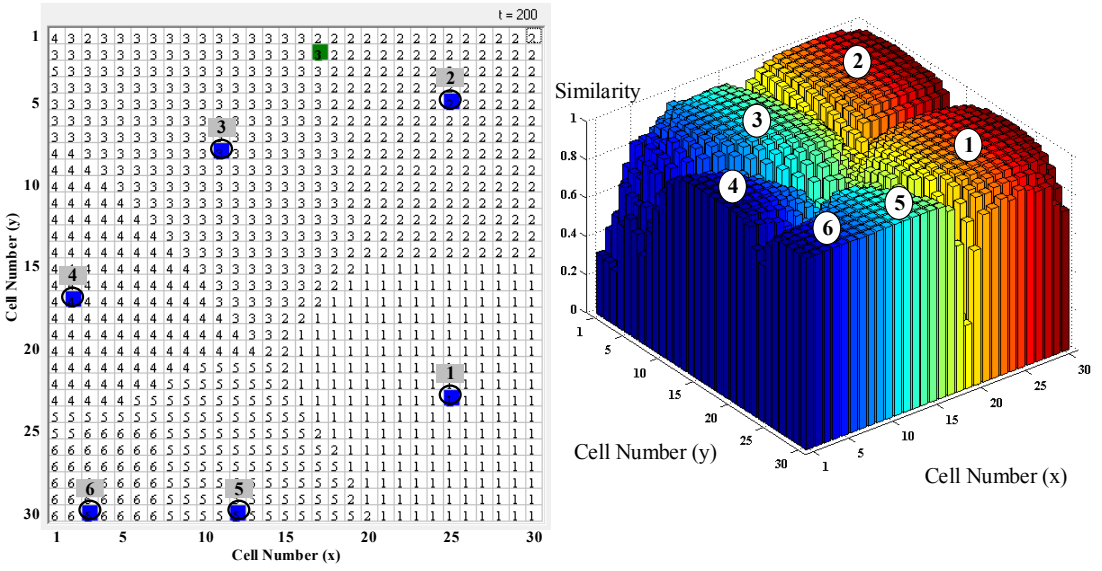
(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-25 (b) During SOM learning in emotion map (t = 100)



(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-26 (a) During SOM learning in behavior map (t = 200)



(i) Output map (two dimensional) (ii) the weight value of each node

Fig. 4-26 (b) During SOM learning in emotion map (t = 200)

After SOM learning (t = 200 times), the blue cells were shown in the behavior and emotion maps, which were the winner nodes of each behavior and emotion of the robot. The green cells that showed the response action according to the input data. In addition, to make it easy to understand,

the similarity of the mapped actions and emotions is presented which is determined by the weight values of each action and emotion group.

In Fig.4-27 showed the winner node of “Approach” action and the response action, and expressed the similarity of approach action when the robot recognized the green object respectively. For the emotion map that presented the winner node of “Happiness” emotion and the response emotion, the similarity of “Happiness” emotion is shown in Fig. 4-28.

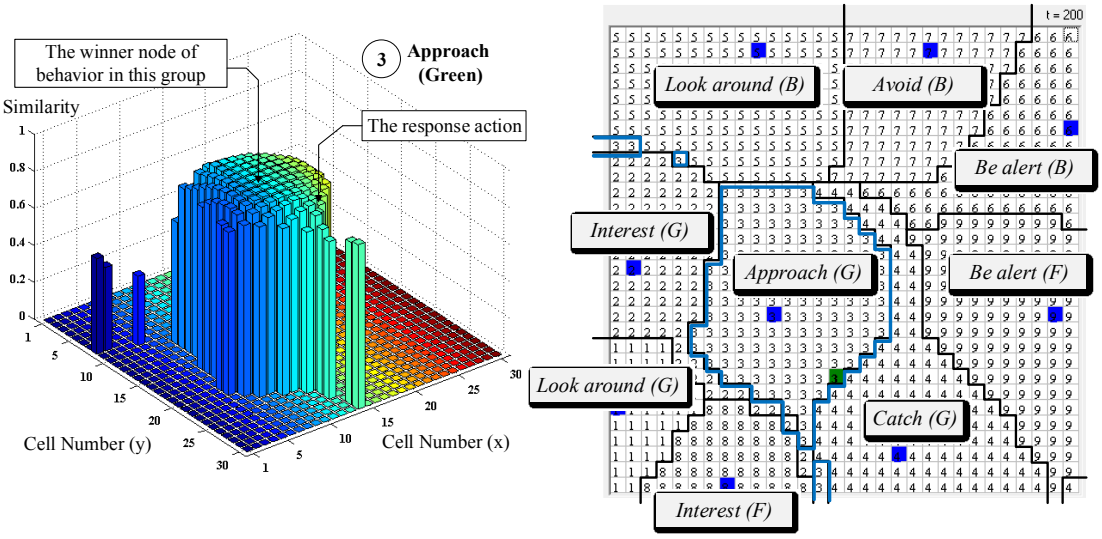


Fig. 4-27 Behavior map and the weight values of “Approach” action

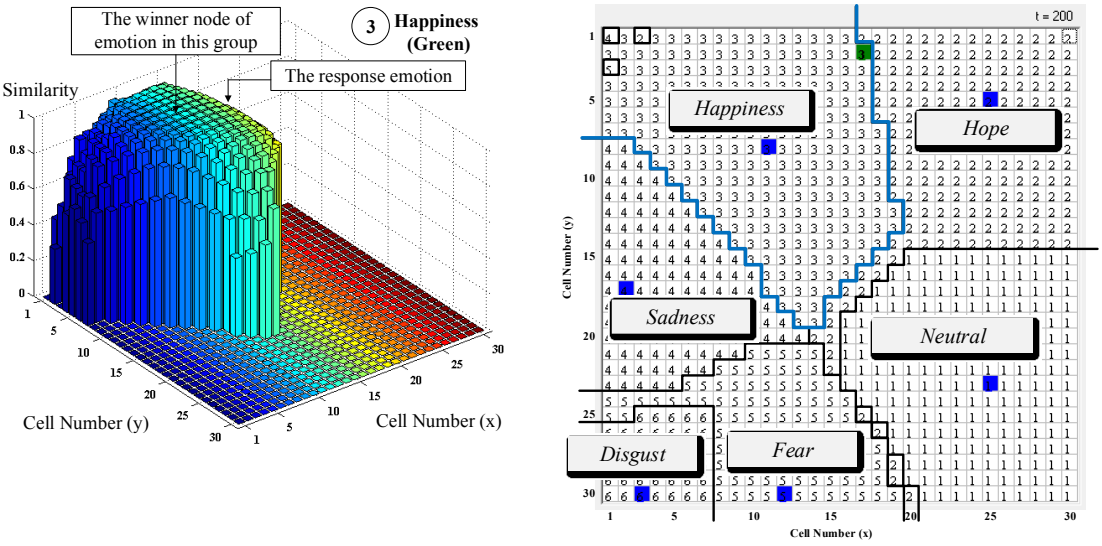


Fig. 4-28 Emotion map and the weight values of “Happiness” emotion

Moreover, during SOM learning, the several of the SOM parameters, such as learning parameters, map size and map topology can influence the formation of the final map. Therefore, in order to guarantee the performance of the map, that has to be sure the mapping parameters have been chosen correctly. Several measures have been used to evaluate the quality of a SOM. A widely used measurement is the average quantization error, this error shows the quality of learning and fitting of the map. It can calculate by the average of all Euclidean distances between each input pattern and its BMU as expressed by Equation (2-20) in Chapter 2.

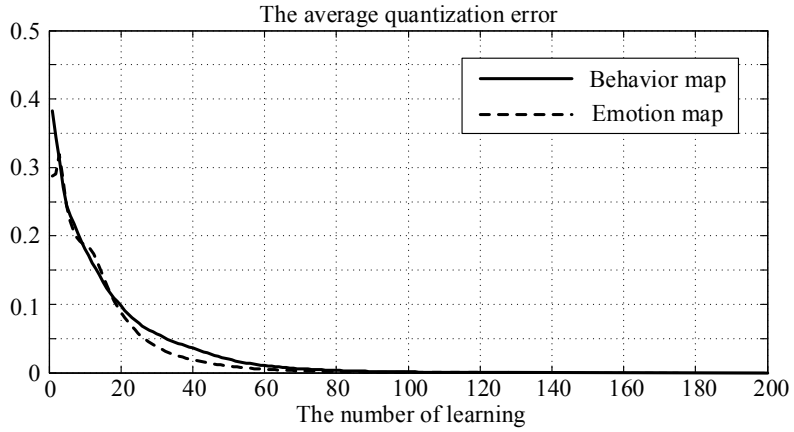


Fig. 4-29 Average quantization error of behavior and emotion maps (Map size 30x30)

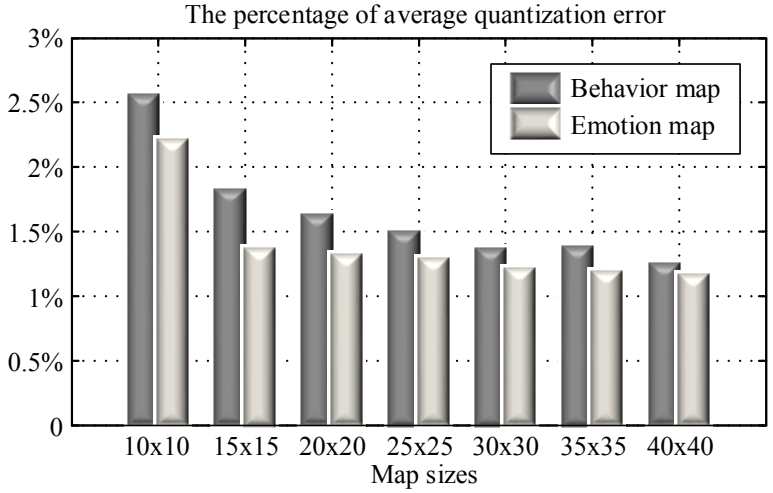


Fig. 4-30 Average quantization error of the different map sizes

The results of the average quantization error of behavior and emotion maps in Fig. 4-27 and Fig. 4-28 can be verified and shown in Fig. 4-29. The optimum map is expected to yield the smallest average quantization error, which means that the data vectors are close to their prototypes. The average quantization error can be reduced by simply increasing the number of map neurons, because the data samples are then distributed more sparsely on the map. Fig. 4-30 shows the average quantization error of the different map sizes, and how the average quantization error of behavior and emotion maps decreases when the map's size increases.

4.2.2 Affective factors and Behavioral factors

After SOM learning, the results from behavior and emotion maps are defined as the behavioral and affective factors, which provide for updating the state transition matrices in the next process. The behavioral and affective factors are determined by averaging the sum of the weights of each group in the behavior and emotional patterns as expressed by Equations (4-27) and (4-28), respectively, where, b_k is the behavioral factor value of each k behavior class, e_j is the affective factor value of each j emotion, m and n are the number of the behavior and emotion weights the contained in each class.

$$b_k = \frac{1}{m}\sum_{i=1}^{m} w_i^k \tag{4-27}$$

$$e_j = \frac{1}{n}\sum_{i=1}^{n} w_i^j \tag{4-28}$$

The resulting value of each affective and behavioral factor is in the range of [0, 1]. For example, $e_{Happiness} = \left\{ w_1^{Happiness}, w_2^{Happiness}, ..., w_i^{Happiness} \right\}$ is the set of weights of affective factor that are

contained in “Happiness” emotion, $w_i^{Happiness}$ is the weight of the i^{th} affective factor contained in “Happiness”. From the emotion map in Fig. 4-28, the weight values in the emotion map are used to calculate the affective factor values as shown in Fig. 4-31.

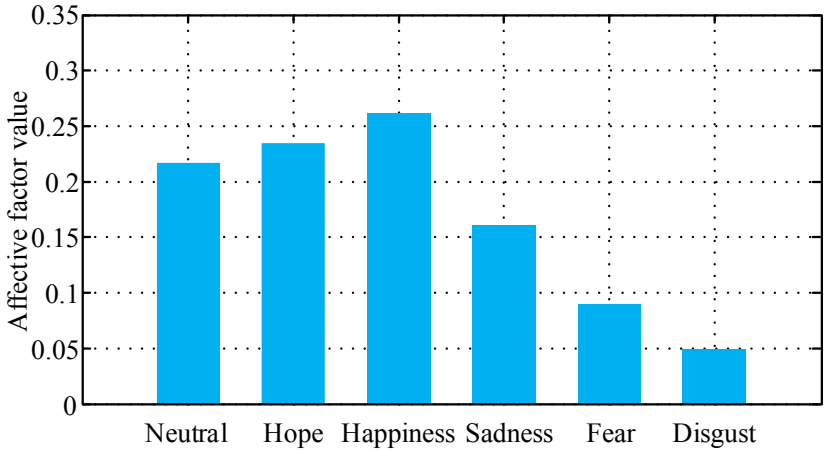


Fig. 4-31 Affective factor values after SOM learning

4.3 Behavioral-Emotional expression process

In recent years, there has been increasing interest in the field of intelligent robots in the study of emotion-generation schemes in order to give the robots more human-like behavior. Several different types of emotion models are available for application level model, task level model, Circumplex model [51], Markovian property [52], [53], PAD model [54], FLAME [55] and so on.

In this study, the Markov model is proposed, because the computational model of emotion based on the Markov theory that adapts emotions in a dynamic and uncertain environment. Therefore the overview of the Behavioral-Emotional expression process can express by Fig. 4-32. It has the emotional state transition module and behavior selection module based on emotional variations, the details of both modules and some the experimental results will be described and verified.

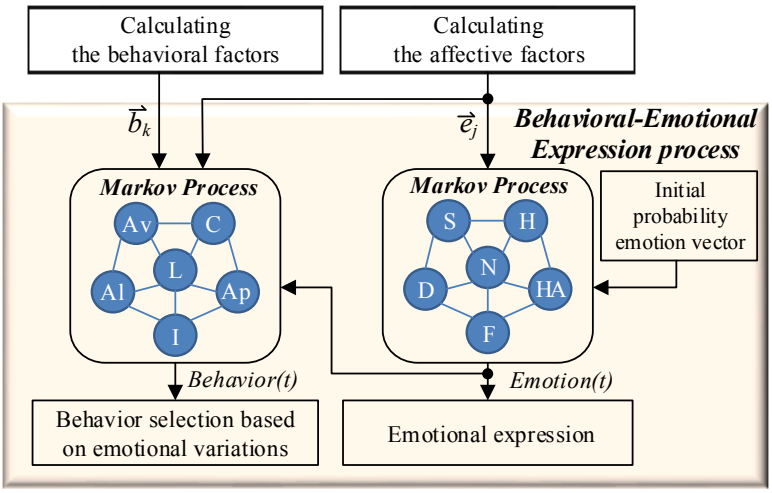


Fig. 4-32 Overview system of Behavioral-Emotional expression process

4.3.1 Emotional state transition

From the theory of the Markov theory is explained in Chapter 2 that can summarize the in this section. A Markov model is a state-space representation of a stochastic process, which can be used to model a random system and changes the states according to a transition rule that depends only on the current state. An application of the modeling theory of this proposed system is described by the Emotional Markovian model as shown in Fig. 4-33.

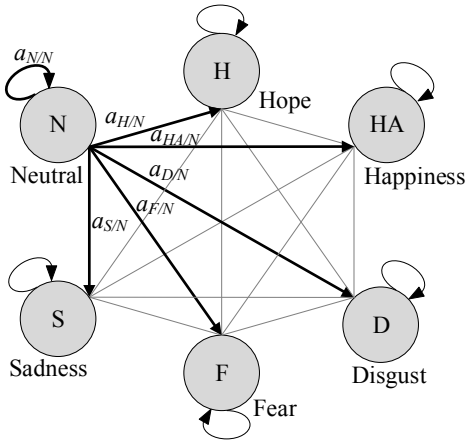


Fig. 4-33 Topology of the Emotional Markovian model

The topology of the Emotional Markovian model consists of the nodes representing the six basic emotional states. The arcs in the model represent the probabilities of getting out of the states. The emotional states transition at time k is given by Equation (4-29), where $Emotion_k$ is the current emotional state vector, and A is the emotional state transition matrix as expressed in Equation (4-30).

$$Emotion_{k+1} = A \cdot Emotion_k \tag{4-29}$$

$$A = \begin{bmatrix} a_{N/N} & a_{N/H} & a_{N/HA} & a_{N/S} & a_{N/F} & a_{N/D} \\ a_{H/N} & a_{H/H} & a_{H/HA} & a_{H/S} & a_{H/F} & a_{H/D} \\ a_{HA/N} & a_{HA/H} & a_{HA/HA} & a_{HA/S} & a_{HA/F} & a_{HA/D} \\ a_{S/N} & a_{S/H} & a_{S/HA} & a_{S/S} & a_{S/F} & a_{S/D} \\ a_{F/N} & a_{F/H} & a_{F/HA} & a_{F/S} & a_{F/F} & a_{F/D} \\ a_{D/N} & a_{D/H} & a_{D/HA} & a_{D/S} & a_{D/F} & a_{D/D} \end{bmatrix} \tag{4-30}$$

Here, $a_{A/B}$ is the probability of a transition from state B to state A, and the arc values are set to initial values as $q_1, q_2, q_3, ..., q_{36}$, which give the initial state transition matrix of Markov model. The values can be changed later by the influence of the affective factors: $e_{Neutral}$, e_{Hope}, $e_{Happiness}$, $e_{Sadness}$, e_{Fear} and $e_{Disgust}$. For example, the probability of the emotional state transition from Neutral to other states can be expressed by Equation (4-31). And all elements in transition matrix are update by using similar equations.

$$\left.\begin{aligned} a_{H/N} &= q_2 + \left(e_{Hope} - e_{Neutral}\right) q_2 \\ a_{HA/N} &= q_3 + \left(e_{Happiness} - e_{Neutral}\right) q_3 \\ a_{S/N} &= q_4 + \left(e_{Sadness} - e_{Neutral}\right) q_4 \\ a_{F/N} &= q_5 + \left(e_{Fear} - e_{Neutral}\right) q_5 \\ a_{D/N} &= q_6 + \left(e_{Disgust} - e_{Neutral}\right) q_6 \\ a_{N/N} &= 1 - \left(a_{H/N} + a_{HA/N} + a_{S/N} + a_{F/N} + a_{D/N}\right) \end{aligned}\right\} \tag{4-31}$$

4.3.2 Behavior selection based on emotional variations

The second stochastic forms a behavior selection module based on its emotional expression. The behavior selection probability vector $Behavior_{k+1}$ is determined by Equation (4-32), where, $Emotion_{k+1}$ is the predicted emotion, C is the behavioral state transition matrix and the elements of $Behavior$ vector are the selection probability of each behavior of the robot.

$$Behavior_{k+1} = C \cdot Emotion_{k+1} \tag{4-32}$$

The elements of the behavioral state transition mean a probability of transition from emotional state to behavioral state as represented by $c_{behavior/emotion}$ and shown in Equation (4-33).

$$C = \begin{bmatrix} c_{L(G)/N} & c_{L(G)/H} & c_{L(G)/HA} & c_{L(G)/S} & c_{L(G)/F} & c_{L(G)/D} \\ c_{In(G)/N} & c_{In(G)/H} & c_{In(G)/HA} & c_{In(G)/S} & c_{In(G)/F} & c_{In(G)/D} \\ c_{App(G)/N} & c_{App(G)/H} & c_{App(G)/HA} & c_{App(G)/S} & c_{App(G)/F} & c_{App(G)/D} \\ c_{Ca(G)/N} & c_{Ca(G)/H} & c_{Ca(G)/HA} & c_{Ca(G)/S} & c_{Ca(G)/F} & c_{Ca(G)/D} \\ c_{L(B)/N} & c_{L(B)/H} & c_{L(B)/HA} & c_{L(B)/S} & c_{L(B)/F} & c_{L(B)/D} \\ c_{Al(B)/N} & c_{Al(B)/H} & c_{Al(B)/HA} & c_{Al(B)/S} & c_{Al(B)/F} & c_{Al(B)/D} \\ c_{Avo(B)/N} & c_{Avo(B)/H} & c_{Avo(B)/HA} & c_{Avo(B)/S} & c_{Avo(B)/F} & c_{Avo(B)/D} \\ c_{In(F)/N} & c_{In(F)/H} & c_{In(F)/HA} & c_{In(F)/S} & c_{In(F)/F} & c_{In(F)/D} \\ c_{Al(F)/N} & c_{Al(F)/H} & c_{Al(F)/HA} & c_{Al(F)/S} & c_{Al(F)/F} & c_{Al(F)/D} \end{bmatrix} \tag{4-33}$$

All element values in the behavioral state transition matrix can be modified later on with behavioral factors and affective factors, for example, if the element values of the behavioral state transition matrix will be updated corresponding to the emotional state transition, the probability of the emotional state from “Neutral” to other behavioral states can be expressed by Equation (4-34).

$$
\left.
\begin{aligned}
c_{L(G)/N} &= q'_1 + \left(b_{Look\,around(Green)} - e_{Neutral}\right) q'_1 \\
c_{In(G)/N} &= q'_2 + \left(b_{Interest(Green)} - e_{Neutral}\right) q'_2 \\
c_{App(G)/N} &= q'_3 + \left(b_{Approach(Green)} - e_{Neutral}\right) q'_3 \\
c_{Ca(G)/N} &= q'_4 + \left(b_{Catch(Green)} - e_{Neural}\right) q'_4 \\
c_{L(B)/N} &= q'_5 + \left(b_{Look\,around(Blue)} - e_{Neutral}\right) q'_5 \\
c_{Al(B)/N} &= q'_6 + \left(b_{Alert(Blue)} - e_{Neutral}\right) q'_6 \\
c_{Avo(B)/N} &= q'_7 + \left(b_{Avoid(Blue)} - e_{Normal}\right) q'_7 \\
c_{In(F)/N} &= q'_8 + \left(b_{Interest(Flesh)} - e_{Neutral}\right) q'_8 \\
c_{Al(F)/N} &= q'_9 + \left(b_{Alert(Flesh)} - e_{Neutral}\right) q'_9
\end{aligned}
\right\} \qquad (4\text{-}34)
$$

Where, the initial values of the behavioral state transition matric are defined as $q'_1, q'_2, q'_3, \ldots, q'_{54}$, and these values can be changed later by the influence of the affective factors and behavioral factors.

4.4 Experimental results

The complete system for an intelligent behavioral-emotional capability of the robot was tested and evaluated. Tests confirmed its effectiveness in the realistic environment. The Conbe-I robot was performed the autonomous behavior and expressed the basic emotions with the robotic arm and the eye movement simulator. In this thesis, the experimental results of the behavioral-emotional selection system were divided into two parts as described below.

4.4.1 Experiment I

The purpose of this experiment, the robotic system was observed the emotions of the robot, which changed depending on the emotional state transition model with affective factors. The situation for testing that defined the green object was near the robot's hand as shown in Fig. 4-34,

the emotion map was generated by the SOM learning. The weight values in the emotion map were used to determine the affective factor values as illustrated in Fig. 4-31.

In this situation, suppose, the Conbe-I robot expressed the current emotional state was "Hope" emotion. Then after computing the emotional state transition model using Markov theory, the robot was able to predict the next emotional state as "Happiness" emotion, corresponding to the intensity of emotion as shown in Fig. 4-35.

Fig. 4-34 Situation for testing in Experiment I

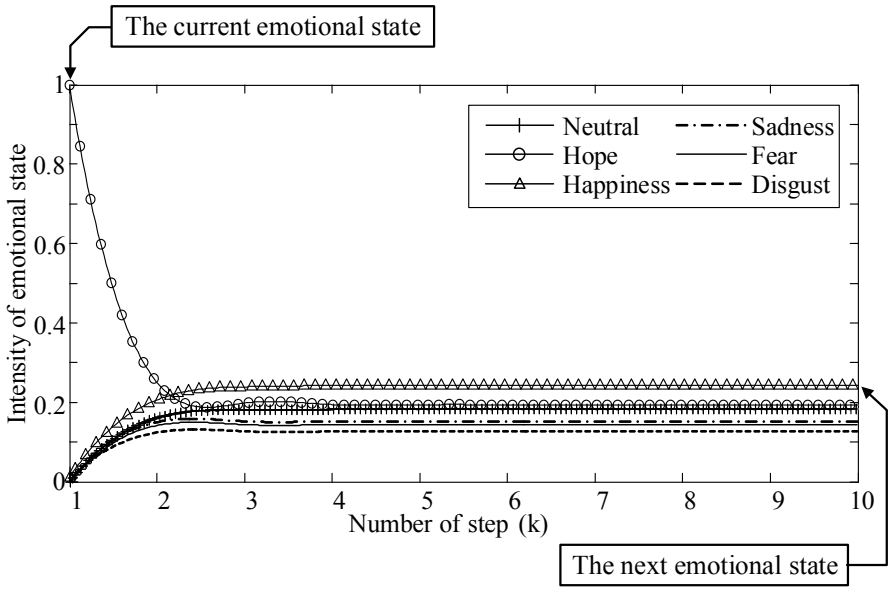


Fig. 4-35 Emotional state with step

4.4.2 Experiment II

The robotic system was observed the behavioral selection based on the emotional variations throughout the task period, which can divided into 2 conditions as liking and disliking behavior.

- Condition 1 (Liking behavior)

The Conbe-I robot was verified its behavior and emotional expression when it recognized the favorite object (the green object) in the realistic environment. The individual input data for testing that consist of robot's motivation values and the visual information as illustrated in Fig. 4-36. And all the input parameters were normalized into a new interval [0, 1].

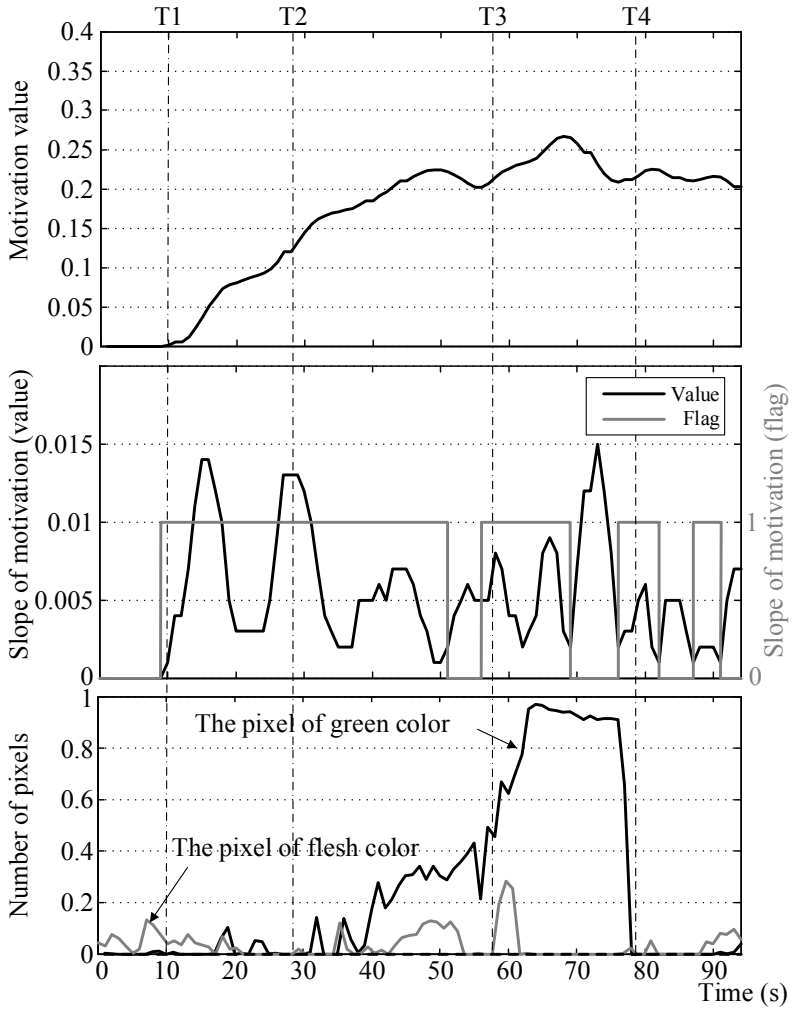


Fig. 4-36 Input data for testing in condition 1

Throughout the task period of condition 1 (T1-T4), the Conbe-I was able to perform the appropriate behaviors and emotional expressions continuously as shown in Fig. 4-37.

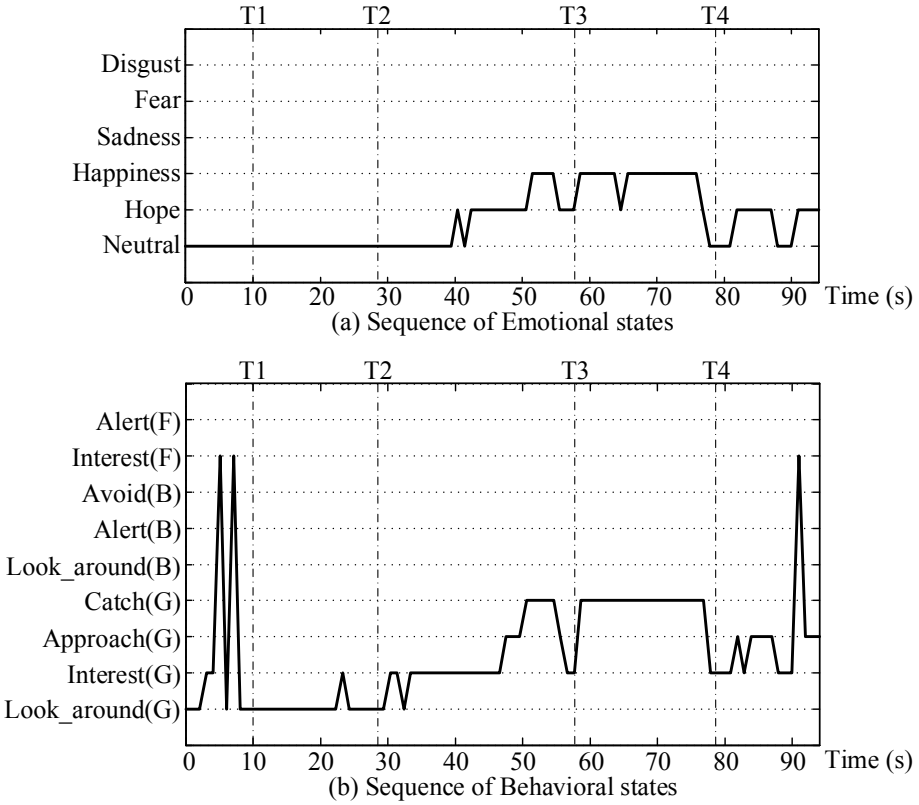


Fig. 4-37 Emotional states and behavioral states (Condition 1)

After the robot was started, it searched around its environment, the robot selected the action as “Look around” and “Interest (Flesh)” when it recognized the flesh-color that means the robot is possible to communicate with humans. At T1, the robot recognized a favorite object (green object). The robot’s motivation value increased and the robot performed the action between “Look around” and “Interest (Green)”; the robot expressed the emotion as “Neutral” (shown in Fig. 4-38).

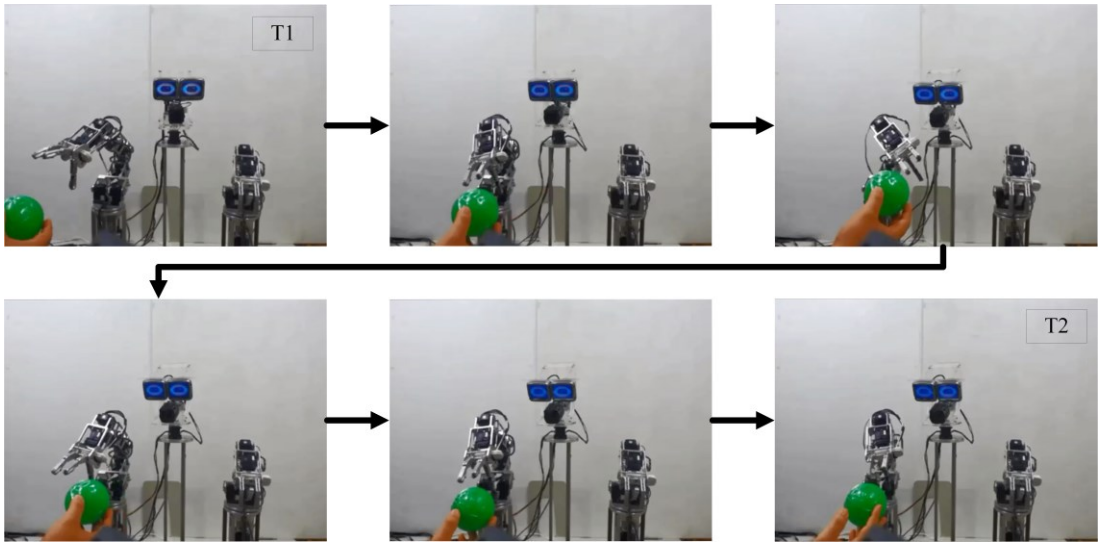


Fig. 4-38 Behavioral and Emotional selections at period time T1-T2

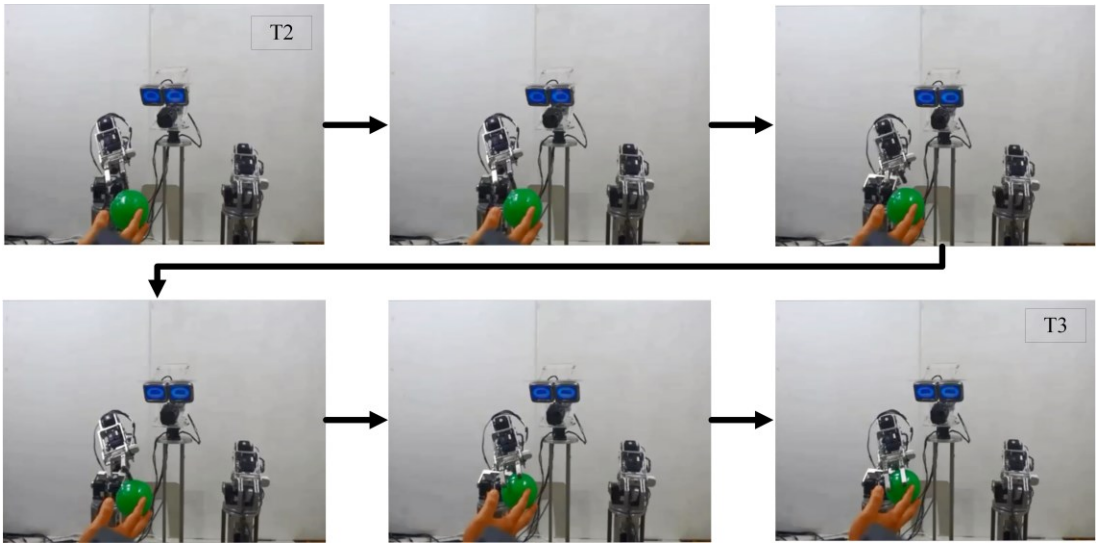


Fig. 4-39 Behavioral and Emotional selections at period time T2-T3

From T2-T3, the robot expressed “Hope” with “Happiness” emotions and performed the behavior as “Interest (Green)” and “Approach” continuously, because the robot tried to get close to the target object until it was able to catch the green ball as represented in Fig. 4-39. Then the time period between T3-T4, the robot was able to possess the favorite object, and it expressed “Happiness” emotion as shown in Fig. 4-40.

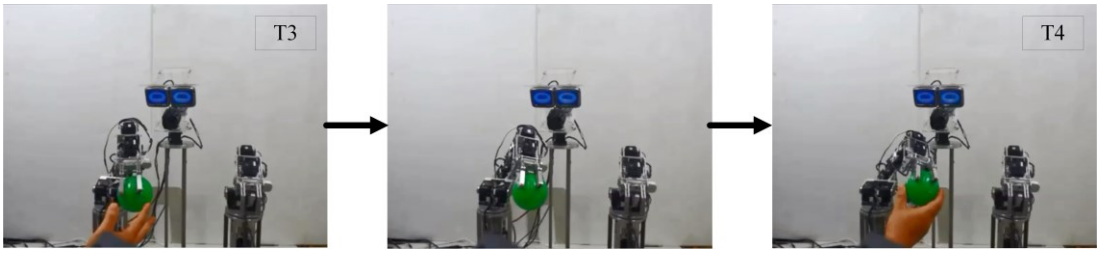


Fig. 4-40 Behavioral and Emotional selections at period time T3-T4

- Condition 2 (Disliking behavior)

In this condition, a disliking (blue) object was presented in order to decrease the robot's motivation. And the input data for testing in condition 2 represented in Fig. 4-41.

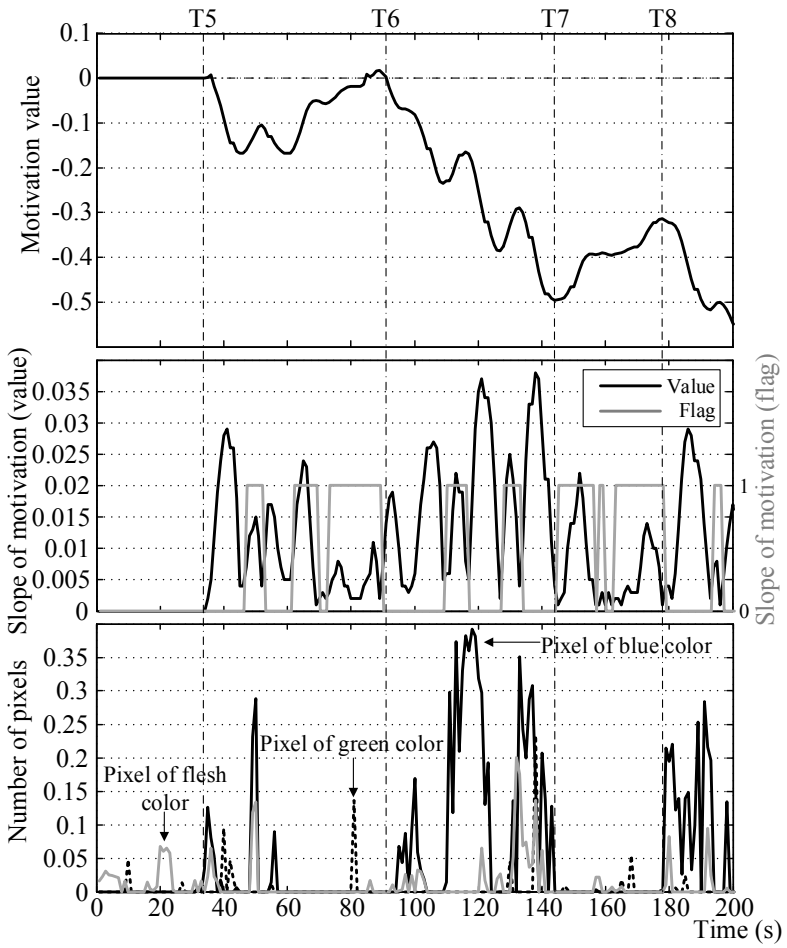


Fig. 4-41 Input data for testing in condition 2

The task period in condition 2 (T5-T8), the Conbe-I was able performed and adapted its behaviors and emotions correspond to the several situations as shown in Figs. 4-42 and 4-43.

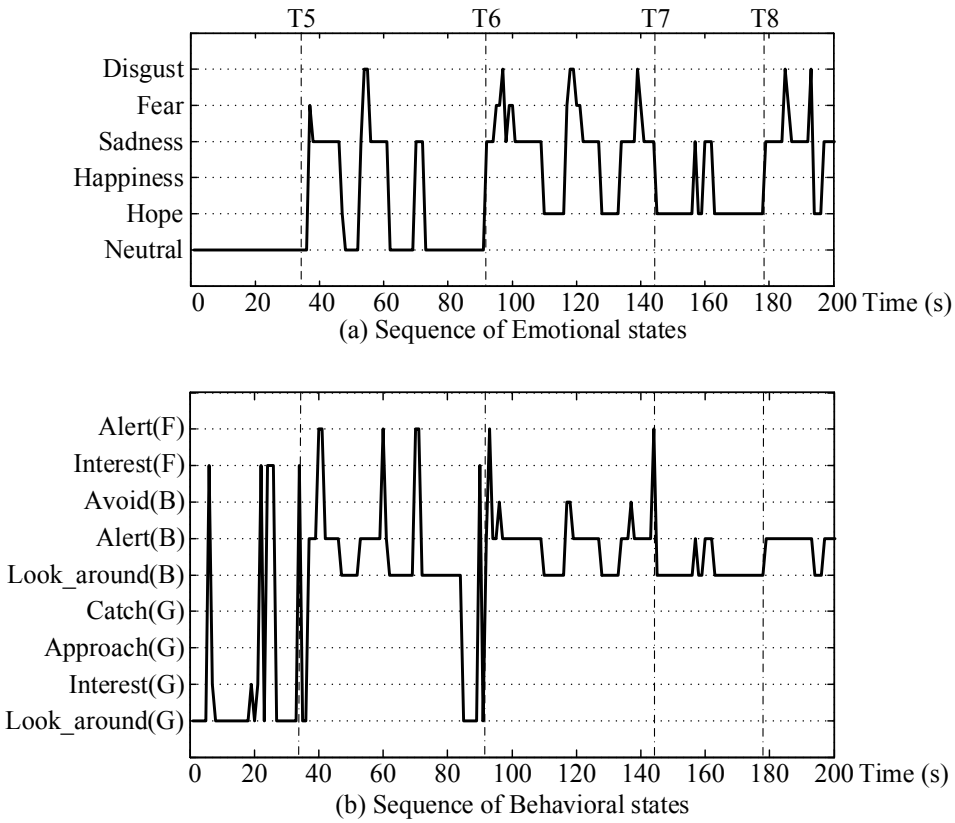


Fig. 4-42 Emotional states and behavioral states (Condition 2)

At T5, the motivation of the robot achieved a negative value, from T5-T6 (Fig. 4-43), the robot recognized the blue object and performed the actions between “Look around (Blue)” and “Alertness”. At the same time, the robot felt unhappiness when it confronted the disliking object, therefore it expressed emotions as “Sadness” and “Fear”. During the period time T6-T7, the blue object was moved closer to the robot’s hand. The Conbe-I robot felt dislike at this situation, and so performed “Alertness” and “Avoidance” behaviors and expressed “Disgust” emotion. Moreover at period time T6-T8, the robot was able to express “Hope” emotion, when it did not capture the disliking object or the robot’s motivation had increased to the positive motivation level.

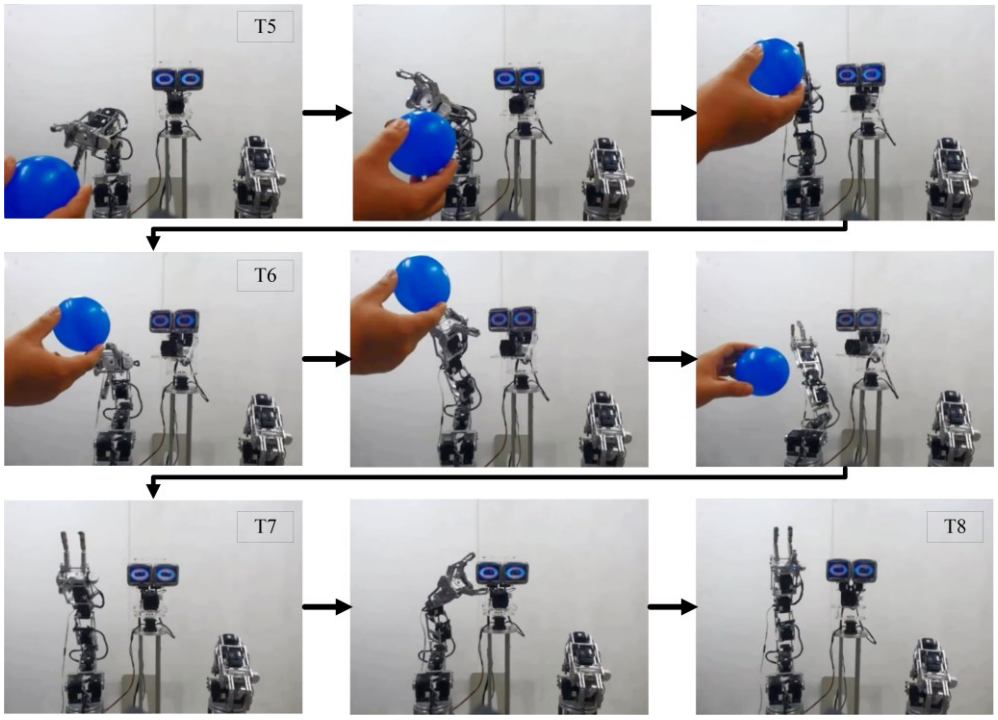


Fig. 4-43 Behavioral and Emotional selection at period time T5-T8

The other research that has the similar results, namely the PIL head robot system [56]. For the PIL robot's emotion model is based on the linear dynamic system, which is difficult to adapt the emotional state immediately. Therefore, the proposed system is possible to improve and fulfill in the dynamics of emotional expression as illustrated by the weight variation of emotional expression in Fig. 4-44 (example in condition 1).

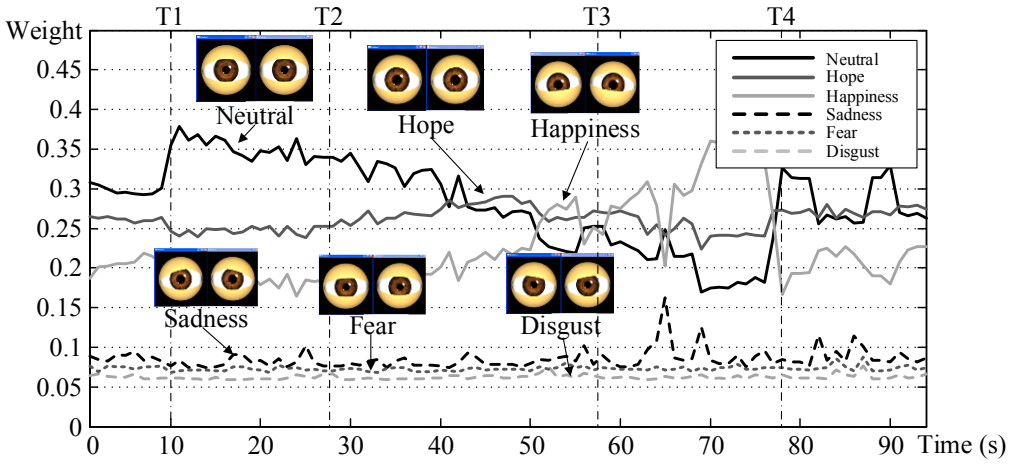


Fig. 4-44 Weight variation of emotional expression (Condition 1)

4.5 Verification

4.5.1 Verification I

There is the question, why emotions need to be twice calculated (they are computed by the Self-Organizing Map and by the Markov theory again). For the answer, the emotions are considered by the Self-Organizing Map to calculate the affective factors. Basically, the value of affective factors can be used to express the emotion, but the Markov model is used for determining in the second time, because an important property is the memoryless property whereby the probability of the next emotional states can be changed by the influence of the affective factors based on the transition rules and the previous emotional states, and one of the reasons is to improve the robot's ability to "think over" for the self-conscious emotional expression. Fig. 4-45 shows the results of emotional states (Condition 2), which is the comparison of differences between before and after using the Markovian model.

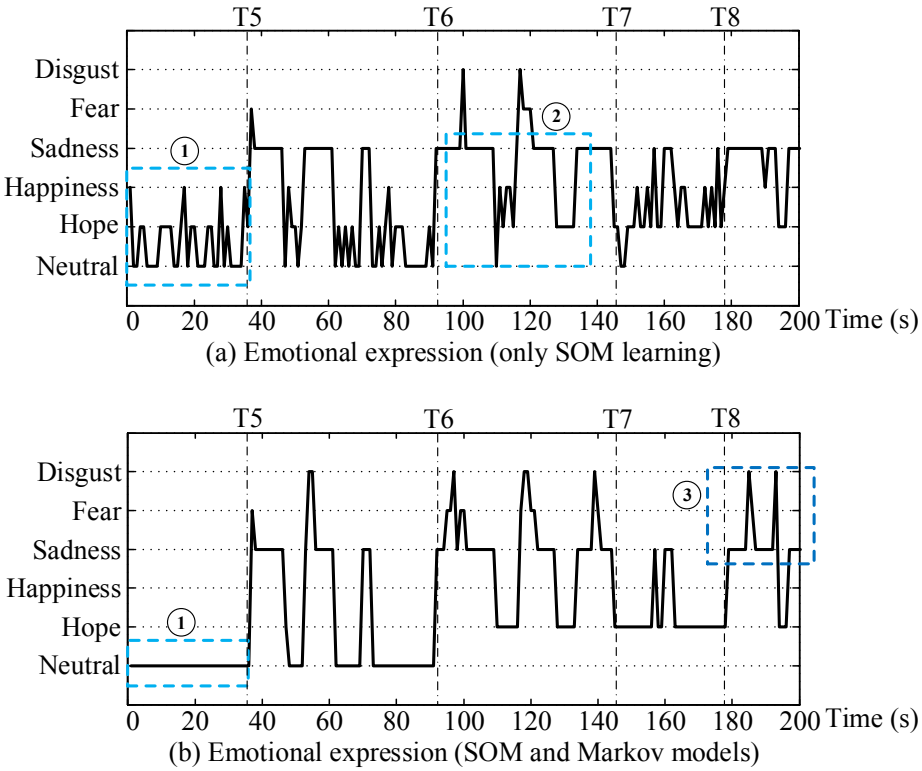
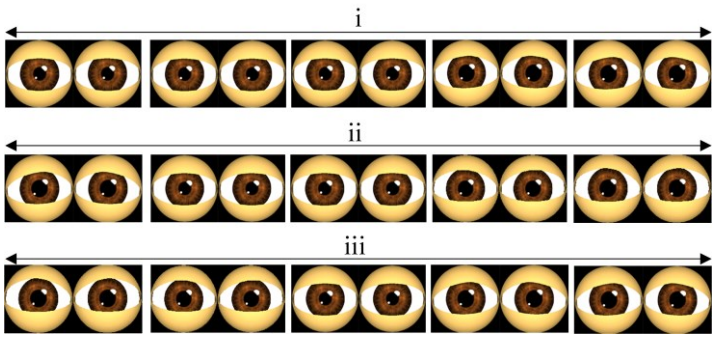


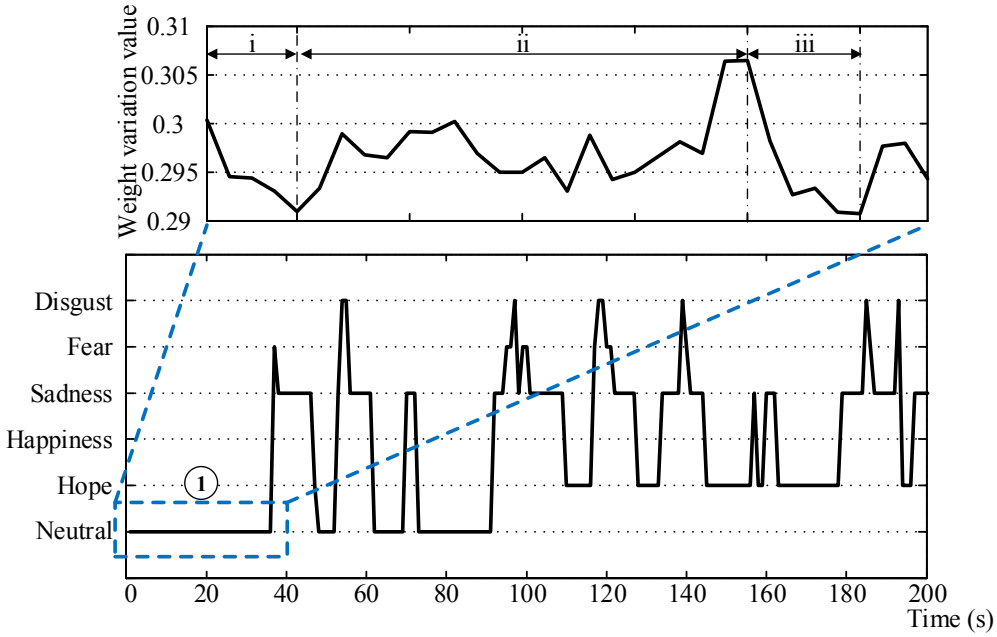
Fig. 4-45 Emotional states in condition 2 (before and after using Markov model)

In Fig. 4-45, that can be considered with 3 points as described below:

At point 1, the emotional expression (without Markov model) in this period that changed emotion states between “Neutral”, “Hope” and “Happiness” suddenly, but for including Markov model with SOM learning, the robot was able to remain “Neutral” emotion and the robot expressed the emotional sequence correspond to the weight variation of “Neutral” emotion as illustrated in Figs. 4-46(a) and 4-46(b). The robot can express emotions evolutionary. At period i, the robot felt a



(a) The eye movements based on the weight variation (at point 1)



(b) Emotional expression (SOM and Markov models)

Fig. 4-46 The weight variation of “Neutral” emotion at point 1

bit sadness because the weight variation decreased and the robot became to have a little bit hope when weight variation of emotion increased as illustrated during period ii. The last period, the robot felt quite a sadness due to the weight variation decreased again.

At point 2, the emotional expression (without Markov model) expressed emotions between "Hope" and "Happiness", the robot should not express these emotions while the blue object was near the robot's hand and the motivation value was continuously decreased. But the system with Markov model was able to suppress "Happiness" emotion in this situation and express the emotions between "Sadness" and "Hope", because it should not be feeling so hopeless although it confronted with a bad situation.

At point 3, the robot's motivation was reduced. During this situation, the emotional expression without a Markov model that express only "Sadness" and "Hope" emotions, but the system with the Markov model can perform and express "Disgust" emotion.

4.5.2 Verification II

In this verification that is the requirement to verify the behavior selection systems (before and after using Markov model). For instance, the behavior selection system is considered in the condition 2 as described above. Fig. 4-47 shows the results (Condition 2) as the comparison of differences between before and after using the Markov model in the behavioral states.

During the task period at T5-T6, the behavior selection system without Markov model selected the actions between "Approach" and "Catch", the robot should not perform these behaviors in this situation, because the robot should be able to just deny the disliking object (Blue object). On the other hand, the behavioral selection system with the Markov model that was able to perform the actions "Alertness" and "Avoidance" without "Approach" and "Catch" behavior based on the emotional state transition between "Sadness" and "Neutral" emotions.

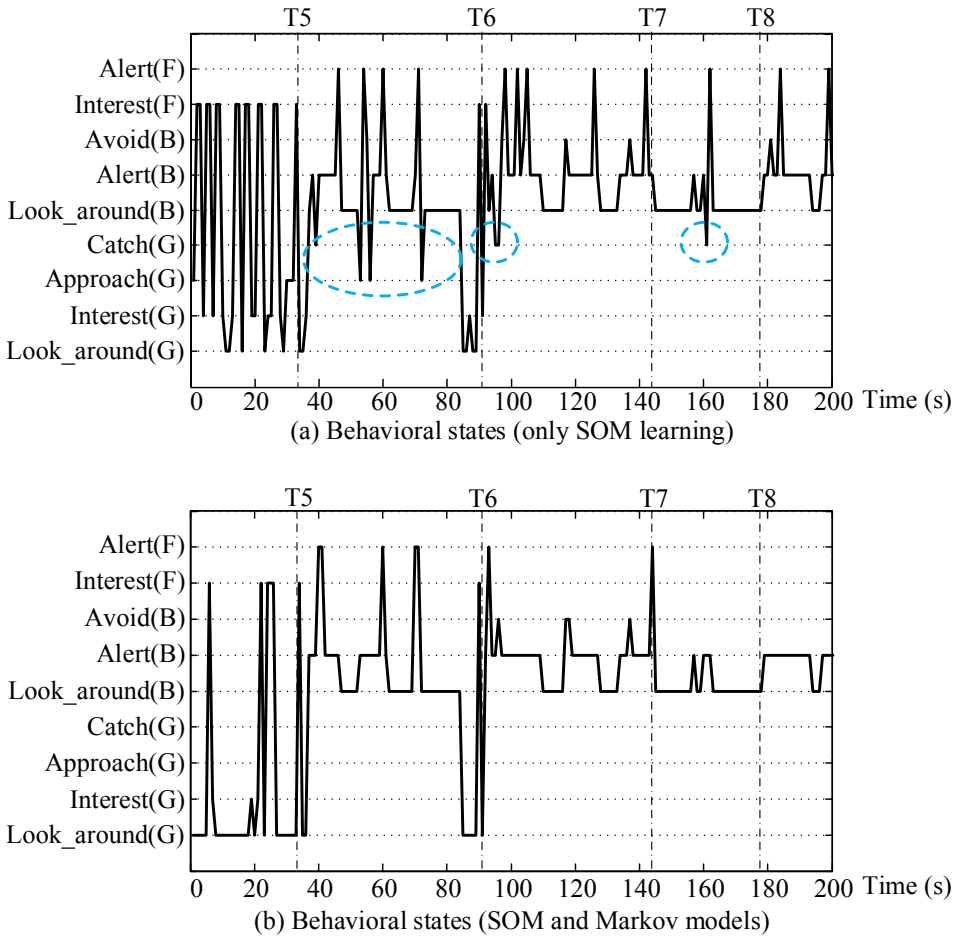


Fig. 4-47 Behavioral states in condition 2 (before and after using Markov model)

4.6 Summary

In this chapter, the overview of the proposed system is thoroughly described that consist of three major processes: the recognition process, cognitive process and behavioral-emotional expression process. The proposed system is executed by the Conbe-I robot in a realistic environment. All experimental results confirmed the effectiveness of the proposed system by dividing as the experiment I (Change emotions based on the emotional state transition model with affective factors), the experiment II (Behavioral selection based on the emotional variations) and the verification I and II showed the comparison of differences of behavioral selection and emotional expression systems between before and after using the Markov model.

Chapter 5

Conclusions

5.1 Conclusions

The implementation of the proposed system (Behavioral-Emotional selection system based on Self-organizing map and Markovian model) is developed from the conventional model (Consciousness-Based Architecture model) and the other researches, which takes its inspiration from the attempt to give the Conbe-I robot need to have the recognition, consciousness and motivation corresponding to the introspective knowledge and some philosophy, in order to select the suitable behavior and emotional expression that look like humans and animals. The main concept and contribution of our proposed method involves in creating behavioral and emotional expressions for the robots based on three neurotransmitters such as dopamine, noradrenaline and serotonin combined with an artificial neural network. However, in this work that used only dopamine to generate a robot's intrinsic motivation. The proposed method has three significant processes. The first process is recognition process that composes of the simple image processing, the calculation of the naturally occurring dopamine and determination of the robot's motivation. In second process, that presents an evolutionary computation of the artificial neuron network method as the unsupervised learning (Self-Organizing Map learning) for clustering the input stimuli (the internal motivation and external situation). The last process is the behavioral-emotional expression process, which develops the model of basic emotions based on Robert Plutchik's research, which consists of six basic emotions: neutral, hope, happiness, sadness, fear and disgust. Moreover, the robot can choose and perform the appropriate behavior based on variations of the emotional state.

From the research purpose that is explained in Chapter 1, That has the attempt to reach all objectives of this research such as trying to classify all behavior and emotions of the robot and generate the behavior and emotion maps based on a Self-Organizing Map (SOM) learning, to create an emotional expression model, which can autonomously express emotional state based on updating the elements of emotional transition state and the affective factors. And the last objective is attempt to define and design the relationship between behavior selection system and emotional expression model. All objectives are achieved by the proposed system and they are verified the effectiveness of system in Chapter 4.

In this thesis, the behavioral-emotional selection system is modeled into the conscious behavior robot (Conbe-I) based on a Self-Organizing Map (SOM) learning and Markovian model. The proposed system can be summarized by dividing to five chapters as described below:

Chapter 1 explains with the background history of the human and robot interaction, the problem statement. The objective of research is to develop a Consciousness-Based Architecture model, which can select the behavior based on changing the emotional state and robot's motivation. Thus, the neuron network learning is presented for improving the conventional model and to solve the problem statements of the research. At the end of chapter 1, the specific objectives of the thesis are illuminated.

Chapter 2 describes briefly in the fundamental theories such as the SOM learning, Markov theory and the kinematics model, which are used to model the behavioral-emotional selection system for the robot. For the background of SOM learning, that is the evolution of relevant biological models such as from two fundamental models (Hebbian learning and Von Malsburg & Willshaw's Self-Organization models) to Kohonen's Self-Organizing Map. A SOM learning is a conceptual mathematic model of topographic mapping from the visual information to the cerebral cortex for modeling and analyzing a high dimensional signal onto a lower dimensional network that implements a characteristic based on a nonlinear projection. In addition, the mathematical model is

also presented as a type of stochastic model, which is called “Markov model”. This model has an important property is “memoryless” property. It means the next state of the system depends only on its current state. In this work, the discrete-state homogenous Markov model is presented and described, because this model is very appropriate for the formation of the human emotions. At the last of chapter 2, the motion control system is described that is called “the kinematics model”. There are two types of kinematic models namely forward kinematics and inverse kinematics models.

Chapter 3 shows the system structure of Conbe-I robot that explains the system configuration and overview of software. The system configuration is explained such as a CCD camera and a web camera are installed for obtaining the images (target object and the external situation). Dynamixel servo motors are used at each joint in the arm part and head part. For the overview of software, Main GUI control is the main program, which can operate the sub-programs such as the image processing module, motivation module, SOM and Markov models and for controlling each servo motor in the robot. Finally, the computation of robotic arm posture is explained in order to perform the sequential movement that can continuously operate the movement, and toward to the target object without the inverse kinematic solution.

Chapter 4 the overview system is described, which consists of three processes (namely, the recognition process, the cognitive process and the behavioral-emotional expression process). At the end of chapter 4 shows the experimental results, all behaviors and emotions were classified on behavior and emotion maps based on SOM learning. The robot’s emotional expression was evaluated by the affective factors, which were used to update the probabilities of emotional state transition based on Markov modeling theory. All experimental results confirmed the effectiveness of the proposed model by dividing as the experiment I (Change emotions based on the emotional state transition model with affective factors), the experiment II (Behavioral selection based on the emotional variations) and the verification I and II showed the comparison of differences of

behavioral selection and emotional expression systems between before and after using the Markov model.

In this chapter is the conclusion that summarizes an autonomous behavioral-emotional system. The robot can perform both select autonomous actions in response to a motivation module and to express intelligent emotions. Most importantly, the advantages of the proposed method, not only Conbe-I robot can use this method, but also the other robot systems can perform by the proposed method with improving or adding the new recognition objects in the SOM learning. In addition, the proposed method is able to predict the next emotion based on the previous emotion. To achieve this proposed method, the emotional transition of Conbe-I robot can be changed easily by modifying the emotional state transition matrix, thus the behavioral-emotional selection system is very flexible and can apply to multiple applications of the robotics system.

5.2 Recommendations for future research

The present approach is able to be further extended to improve the overall performance of the proposed system, some recommendations for future research are suggested as follows:

1. To increase the ability of the robotic system by applying artificial intelligence for memorizing the situation and developing a robot capable to think, learn and take on tasks it hasn't tried before.

2. To study and investigate other neurotransmitters, such as noradrenaline and serotonin, to combine with the dopamine system for generating a dynamic emotional expression model that is similar to Lövheim cube of emotional model as shown in Fig. 5-1.

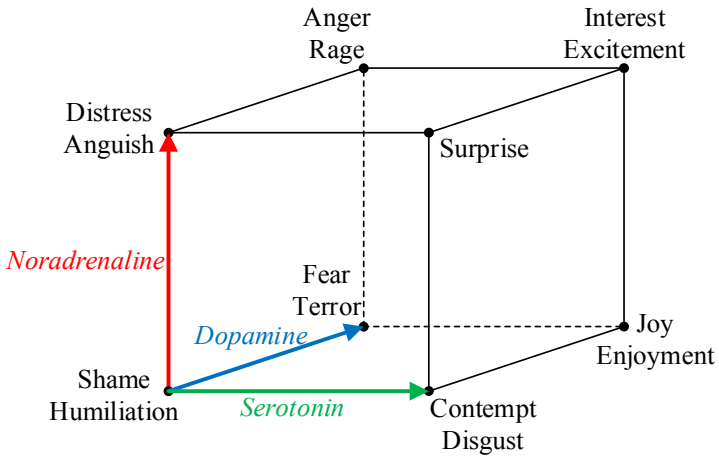


Fig. 5-1 Lövheim cube of emotional model

References

[1] S. JOSE, (2010, July 12), 'A Global Market Report', Service Robotics [Online].Available: http://www.prweb.com/releases/service_robotics/professional_robots/prweb4240924.htm.

[2] S. Kiesler and P. Hinds, "Introduction to this special issue on human-robot interaction," J. of Human-Computer Interaction Vol.19, No.1, pp. 1-8, 2004.

[3] C. Breazeal, "Function Meets Style: Insights From Emotion Theory Applied to HRI," IEEE Trans. on Systems, Man, and Cybernetics, Vol. 34, No. 2, pp. 187-194, 2004.

[4] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A Survey of Socially Interactive Robots," J. of Robotics and Autonomous Systems, Vol.42, pp. 143-166, 2003.

[5] Y. Mori, K. Ota, and T. Nakamura, "Robot Motion Algorithm Based on Interaction with Human," J. of Robotics and Mechatronics, Vol.14, No.5, pp. 462-470, 2002.

[6] J. McCarthy, "Making robots conscious of their mental states," In Muggleton S (Eds.), Machine intelligence. Oxford: Oxford University Press, pp. 3–17, 1996.

[7] N. Goto, and E. Hayashi, "Designed of Robotic Behavior that imitates animal consciousness," J. of Artificial Life and Robotics, vol. 12, pp. 97-101, 2008.

[8] E. Hayashi, T. Yamasaki, and K. Kuroki, "Autonomous Behavior System Combing Motivation with Consciousness Using Dopamine," IEEE Int. Symp. on Computational Intelligence in Robotics and Automation, pp. 126-131, 2009.

[9] M. Yoshida, and E. Hayashi, "Designed of robotic behavior that imitates animal consciousness: Construction of the user-recognition systems," in Proc. AROB 17th, 6 pages, 2012.

[10] T. D. Thao, D. J. Herman, and D. V. Morano, "Phenomenology and Dialectical Materialism," (Boston Studies in the Philosophy of Science), D. Reidal, Pub. Co., 1986.

[11] C. Woo Yu, and J. Young Choi, "Behavior Decision Model Based on Emotion and Dynamic Personality," Int. Conf. on Computer Applications in Shipbuilding (ICCAS), 6 pages, 2005.

[12] R. W. Picard, "Affective computing," paperback edn., First MIT Press, Cambridge, 2000.

[13] J. Moren and C. Balkenius, "A computational Model of Emotional Learning in the Amygdala," J. of Cybernetics and Systems Vol.32, No. 6, pp.611-636, 2001.

[14] M. Obayashi, T. Takuno, T. Kuramoto, and K. Kobayashi, "An Emotional Model Embbed Reinforcement learning System," IEEE Int. Conf. on System, Man, and Cybernetics (SMC), pp.1058-1063, 2012.

[15] A. Ortony, G. L. Clore, and A. Collins, "The Cognitive Structure of Emotions," [M]. London: Cambridge University Press, 1988.

[16] Q. Xu, L. Zhou, and F. Jiao, "Design for User Experience: an Affective-Cognitive Modeling Perspective," IEEE Int. Conf. on Management of Innovation and Technology (ICMIT), pp. 1019-1024, 2010.

[17] N. Kubota, and S. Wakisaka, "An Emotional Model Based on Location-Dependent Memory for Partner Robots," J. of Robotics and Mechatronics, Vol.21, No.3, pp. 317-323, 2009.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1, Foundations, D. E. Rumelhart, J. L. McClelland and the PDP research group, Eds. Cambridge, Mass, MIT Press, 1986, pp. 318-362.

[19] R. S. Sutton, and A. G. Barto, "Reinforcement Learning: An Introduction," MIT Press, Cambridge, MA, 1998.

[20] Y. Liu, and R. H. Weisberg, "A Review of Self-Organizing Map Applications in Meteorology and Oceanography," Journal of Self-Organizing Map – Applications and Novel Algorithm Design, pp. 253-272, 2011.

[21] T. Kohonen. "Self-organized formation of topologically correct feature map," Biological Cybernetics, vol. 43, pp. 56-69, 1982.

[22] D.O. Hebb. "The organization of behavior," New York: Wiley, 1949.

[23] E. Oja. "A simplified neuron model as a principal component analyser," Journal of Mathematical Biology, vol. 16, pp. 267-273, 1982.

[24] E. Oja. "Neural networks, principal components and subspaces," International Journal of Neural Systems," vol. 1, pp. 61-68, 1989.

[25] E. Oja. "PCA, ICA and nonlinear Hebbian learning," In: Fogelman-Soulié F, Gallinari P (eds.) Proc. Intl. Conf. Artificial Neural Networks, 9-13 October, Paris, France, pp. 89-94, 1995.

[26] D. Marr. "A theory of cerebellar cortex," Journal of Physiology, vol. 202, pp. 437-470, 1969.

[27] D. J. Willshaw, O. P. Buneman, and H. C. Longnet-Higgins. "Non-holographic associative memory," Nature, vol. 222, pp. 960-962, 1969.

[28] C. von der Malsburg, and D. J. Willshaw. "Self-organization of orientation sensitive cells in the striate cortex," Kybernetik, vol. 4, pp. 85-100, 1973.

[29] D. J. Willshaw, and C. von der Malsburg. "How to patterned neural connections can be set up by self-organization," Proc. Royal Society of London – Series B, vol. 194, pp. 431-445, 1976.

[30] S. I. Amari. “Topographic organization of nerve fields,” Bulletin Mathematical Biology, vol. 42, pp. 339-364, 1980.

[31] H. Ritter, T. Martinetz, and K. Schulten. “Neural Computation and Self-organizing Maps: An Introduction,” Addison-Wesley, Reading, MA, 1992.

[32] J. H. Kaas, M. M. Merzenich, and H. P. Killackey. “The reorganization of somato-sensory cortex following peripheral nerve damage in adult and developing mammals,” Annu. Rev. Neurosci., vol. 6, pp. 325-356, 1983

[33] D. H. Hubel, and T. N. Wiesel. “Receptive fields. binocular interaction and functional architecture in the cat’s visual cortex, ” J. Physiol., vol. 160, pp. 106-154, 1962.

[34] D. H. Hubel, and T. N. Wiesel. “Ferrier Lecture. Functional architecture of macaque monkey visual cortex,” Proc. R. Soc. Lond. B., vol. 198, pp 1-59, 1977.

[35] N. Suga. “The extent to which biosonar information is represented in the bat auditory cortex,” In Neuro computing 2: Directions for Research, eds J. A. Anderson, A. Pellionisz, E. Rosenfeld, editors. (Cambridge, MA: MIT Press ;), pp. 259-294, 1985.

[36] T. Kohonen, “Self-Organizing Maps,” Springer Series in Information Science, 3rd ed., Springer-Verlag, Berlin Heidelberg, vol. 30, 2001.

[37] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, “Engineering Applications of the Self-Organizing Map,” Proceedings of the IEEE, vol. 84, no. 10, pp. 1358-1384, 1996.

[38] K. Obermayer, K. Schulten, and G. G. Blasdel. “A neural network model for the formation and for the spatial structure of retinotopic maps, orientation and ocular dominance columns,” Theoretical Biophysics Technical Report TB-91-09, The Beckman Institute, University of Illinois Urbana-Champaign, 1991.

[39] M. Herrmann, “Self-Organizing Feature Map with Self-Organizing Neighborhood Widths,” Proc. on IEEE Int. Conf. on Neural Networks, vol. 6, 1995.

[40] A. Flexer, “Limitations of self-organizing maps for vector quantization and multi-dimensional scaling,” Technical Report oefai-tr-96-23, The Australian Research Institute for Artificial Intelligence, 1997.

[41] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, “SOM Toolbox for Matlab 5,” Technical report, pp. 59, 2000.

[42] E. A. Uriaite, and F. D. Martin, “Topology Preservation in SOM,” International Journal of Mathematical and Computer Sciences, vol. 1, pp. 19-22, 2005.

[43] J. Bates, “The Role of Emotion in Believable Agents,” Communication of the ACM, vol. 37(7), pp. 122-125, 1994.

[44] A. Margulies, “Empathy, Virtually, and the Birth of Complex Emotional States: Do We Find or Do We Create Feelings in the Other,” S. L. Albon, D. Brown, E. J. Khantzian, and J. E. Mack (Eds), Hum. Feel. Explor. Affect Dev. Meaning, Anal. Press., 1993.

[45] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, “Introduction in Robot Modeling and Control,” John Wiley and Sons, Inc., New Jersey, NJ, 2006.

[46] J. Denavit, and R. S. Hartenberg, “A kinematic notation for lower-pair mechanisms based on matrices,” Journal of Applied Mechanics, vol. 1, pp. 215-221, 1955.

[47] J. Funda, R. H. Taylor, and R. P. Paul, “On homogeneous transform, quaternions, and computational efficiency,” IEEE Trans. Robot. Automat., vol. 6, pp. 382-388, 1990.

[48] J. J. Craig, “Introduction to Robotics Mechanics and Control,” USA: Addison Wesley Publishing Company, 1989.

[49] R. Plutchik, “Emotion: Theory, research, and experience,” vol. 1, Theories of emotion 1, New York: Academic, 1980.

[50] H. Kimura, “A trial to analyze the effect of an atypical anti-psychotic medicine, risperidone on the release of dopamine in the central nervous system,” J of Aichi Med. Univ. Assoc., vol. 33(1), pp. 21-27, 2005.

[51] J. A. Russel, “A circumplex model of affect,” J. of Personality and Social Psychology, vol. 39, pp. 1167-1178, 1980.

[52] K. Koiji, and B. Martin, “Towards an emotion core based on a Hidden Markov Model,” Proc. of 13th IEEE International Workshop on Robot and Human Interactive Communication, Sep. 20-22, Kurashiki, Okayama, Japan, pp. 119-124, 2004.

[53] K. Inoue, T. Arai, and J. Ota, “Behavior Acquisition in Partially Observable Environments by Autonomous Segmentation of the Observation Space,” J. of Robotics and Mechatronics, vol. 27(3), pp. 317-323, 2015.

[54] A. Mehrabian, “Framework for a comprehensive description and measurement of emotional states,” J. of Genetic, Social, and General Psychology Monographs, vol. 121(3), pp. 339-361, 1995.

[55] M. S. EI-Nasr, T. loerger, and J. Yen “FLAME: fuzzy logic adaptive model of emotions,” J. of Autonomous Agents and Multi-Agent Systems, vol. 3(3), pp. 219-257, 2000.

[56] H. Ahn, P. Kim, and J. Choi et al, “Emotion head robot with behavior decision model and face recognition,” Proc. of International Conference on Control, Automation and Systems, Oct. 17th-20th, COEX, Seoul, Korea, pp. 2719-2724, 2007.

[57] H. Lövheim, “A new three-dimensional model for emotions and monoamine neurotransmitters,” J. of Medical Hypotheses, Vol.78, pp. 341-348, 2012.

Appendix

A. The input patterns (Behavior and Emotion patterns)

Table A.1 Behavior pattern for SOM learning (behavior map)

	Moti_flag	Moti_value	Moti_slope_flag	Moti_slope_value	Green	Blue	Flesh
Look around (green object)	1	0.01	1	0.001	0.05	0.001	0.01
Interest (green object)	1	0.2	1	0.01	0.1	0.001	0.01
Approach (green object)	1	0.4	1	0.02	0.4	0.001	0.01
Catch (green object)	1	0.6	1	0.001	0.8	0.001	0.01
Look around (blue object)	0	0.01	1	0.001	0.001	0.05	0.01
Be alert (blue object)	0	0.4	0	0.01	0.001	0.4	0.01
Avoidance (blue object)	0	0.6	0	0.001	0.001	0.8	0.01
Interest (flesh-color)	1	0.15	1	0.001	0.001	0.001	0.2
Be alert (flesh-color)	0	0.15	0	0.001	0.001	0.001	0.2

Table A.2 Emotion pattern for SOM learning (emotion map)

	Moti_flag	Moti_value	Moti_slope_flag	Moti_slope_value	Green	Blue	Flesh
Neutral	1	0.01	1	0.001	0.05	0.001	0.01
Hope	1	0.4	1	0.002	0.3	0.001	0.01
Happiness	1	0.6	1	0.005	0.8	0.001	0.01
Sadness	0	0.1	0	0.01	0.001	0.1	0.01
Fear	0	0.25	0	0.02	0.001	0.3	0.01
Disgust	0	0.5	0	0.001	0.001	0.6	0.01

B. Publications and Presentations from the Present Research Work

1. Wisanu Jitviriya, Masato Koike, and Eiji Hayashi, "Emotional Model for Robotics System Using a Self-Organizing Map Combined with Markovian Model," Journal of Robotics and Mechatronics (JRM), vol. 27, no. 5, pp. 563-570, 2015.
2. Wisanu Jitviriya, Masato Koike, and Eiji Hayashi, "Behavior Selection System Based on Emotional Variations," in Proc. the 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2015), pp. 462-467, August 31 – September 4, 2015.
3. Wisanu Jitviriya, Masato Koike, and Eiji Hayashi, "Development of Behavior and Emotion models for Conbe-I using a Self-Organizing Map learning," in Proc. the 23th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2014), pp. 411-416, August 25 – 29, 2014.
4. Wisanu Jitviriya, and Eiji Hayashi, "Analysis of Robotic Arm's Behavior using Self Organizing Map combined with Consciousness-Based Architecture Module," in Proc. 2013 IEEE/SICE International Symposium on System Integration (SII2013), pp. 533-538, December 15-17, 2013.
5. Wisanu Jitviriya, Motoki Yoshida, Kyoko Tanaka, and Eiji Hayashi, "Development of Memorizing System with Action Selection based on Consciousness and Behavior of Animal on Robot Systems," in Proc. the 21th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2012), pp. 521-526, September 9 – 13, 2012.

List of Figures

Page

Page

Page

Page

List of Tables

Page

Acknowledgements

I would like to express my special appreciation and thanks to my advisor, Professor Dr. Eiji HAYASHI, you have been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. Your advice on both research as well as on my career has been priceless. And I would also like to thank my committee members, Professor Dr. Hiroyuki NARAHARA, Professor Dr. Takahiro ITO, Professor Dr. Masato TSURU, Associate Professor Dr. Keisuke SUZUKI, and Associate Professor Dr. Hiroshi OHTAKE for serving as my committee members. I also want to thank you for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions.

I also thank the Rotary Yoneyama Memorial Foundation scholarship (Professor Dr. Masuda YUICHI), Satooya scholarship, and the 100th Anniversary Memorial scholarship. Your generosity allows me to take my goals and dreams a reality and this scholarship has afforded me the opportunity to continue my educational pursuits.

Finally, a special thanks to my family. Words cannot express how grateful I am to my good father (Mr. Wisit Jitviriya), mother (Mrs. Sunee Jitviriya) and my family for all of the sacrifices that you have made on my behalf. Because of your prayer and support for me are what sustained me thus far.

WISANU JITVIRIYA