International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:3, 2007

Learning Process Enhancement for Robot Behaviors

Saeed Mohammed Baneamoon, Rosalina Abdul Salam, and Abdullah Zawawi Hj. Talib

Abstract**—**Designing a simulated system and training it to optimize its tasks in simulated environment helps the designers to avoid problems that may appear when designing the system directly in real world. These problems are: time consuming, high cost, high errors percentage and low efficiency and accuracy of the system. The proposed system will investigate and improve the efficiency and accuracy of a simulated robot to choose correct behavior to perform its task. In this paper, machine learning, which uses genetic algorithm, is adopted. This type of machine learning is called genetic-based machine learning in which a distributed classifier system is used to improve the efficiency and accuracy of the robot. Consequently, it helps the robot to achieve optimal action.

Keywords—Machine Learning, Genetic-Based Machine Learning, Learning Classifier System, Behaviors.

I. Introduction

ROBOT plays an important role in different areas in real world; this refers to their ability to perform difficult tasks in complex an unstructured environments with higher efficiency and accuracy.

However, robotics systems do have undesirable limitations in their efficiency and accuracy; therefore machine learning techniques prodvide solutions to overcome these limitations. In principle, machine learning techniques are used to allow robot to successfully adapt with their environment [3].

Classifier system is a class of machine learning systems which use bucket brigade (BB) algorithm and genetic algorithm (GA) as learning mechanisms to produce adaptive system. The role of BB algorithm in classifier system is to assign better action while genetic algorithm is used to generate new rules [1], [3].

This paper is organized as follows: Section 2, presents the background of this study. In section 3, related studies are reviewed. The research methodology is described in section 4. Section 5 introduces the properties of the proposed system. Finally, conclusion of present research topic is presented.

Manuscript received April 02, 2007.
Saeed Mohammed Baneamoon is with School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia (e-mail: saeed@cs.usm.my).
Rosalina Abdul Salam is with School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia (phone: + 604-6532486; fax: +604-6573335; e-mail: rosalina@cs.usm.my).
Abdullah Zawawi Hj. Talib is with School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia (phone:+ 604-6534380; fax: +604-6573335; e-mail: azht@cs.usm..my).

II. Background

A. Machine Learning

Machine learning (ML) is one of the important areas in artificial intelligence that concentrates on useful development of techniques which focused on improving the systems through designing programs that have ability to increase their performance with their experiences[2],[7].

From the above, one can say that ML techniques are able to help robotics systems to achieve their behaviors with efficiency and accuracy [1]-[3], [9].

B. Genetic-Based Machine Learning

As they use GA to achieve role discovery these systems are called genetic-based machine learning (GBML). The principle theory of GBML systems was introduced by Holland (1962) [1], [4]. GBML systems are defined as a type of ML that have been learned to achieve task through interaction with an environment [1], [4].

C. Learning Classifier System

Learning classifier system (LCS) is a class of GBML which is first introduced by Holland (1976).

LCSs are used as main components in designing the simulated robot control system to determine which behavior must be used [3] – [5], [8].

LCS uses two learning process; (BB) algorithm, which works in the apportionment of credit, and GA which works in the rule discovery system [1], [3], [10].

The three main components of LCS are: the performance system (rule and message system), the apportionment of credit system and the rule discovery system.

1) The Performance System

The performance system consists of:
- Input and output interface (Detectors and effectors).
- A classifier store.
- A message list

The performance system is responsible for the interaction between the robot and the external environment through the detectors and effectors. On the other hand, classifier store has a fixed size with a set of classifier, each classifier in the classifier store consists of condition/action part and its strength that determines the winner classifier, the condition/action part has the following form:

IF <condition1> & <condition2> &..& <condition N>
THEN<action>

The condition part in the classifier store is encoded as string with fixed length from the ternary alphabet {0,1,#}. On the other hand, action part is encoded with fixed length string defined by binary alphabet {0,1} [1], [3], [4], [10].

In the condition part the "#" symbol which is called "don't care" makes the classifier more general because the match is either 0 or 1.

The strength of the classifier is based on a fitness value that is used as a measure to determine the degree of performance of classifier. The higher the strength the better the performance of the classifier. In the beginning all the classifiers have the same initial strength values [1], [3], [10].

The environmental message received by the detectors is matched against the condition part of classifiers in the classifier store, all classifiers that match the environmental message is collected in matching pool then it is sent to the apportionment of credit system [1], [3], [10].

The basic execution cycle of the performance system is as follows:

Step 0: Read messages by detectors and place them on the current message list.
Step 1: Determine the matching classifiers by comparing all messages to all conditions.
Step 2: For each match generate a message for the new message list.
Step 3: Replace the current message list by the new message list.
Step 4: Process the new message list through the effectors to produce system output.
Step 5: Return to step 0.

2) The Apportionment of Credit System

The function of Apportionment of Credit System (AOC) is to classify the matching classifiers depending on their usefulness. In AOC the strength of the classifiers in matching pool is modified. The strength modification occurs by redistributing rewards to useful classifiers and others will be punished [1] – [3].

In this part the conventional BB algorithm is used. The function of BB algorithm is to determine the best classifier by changing the strength of classifiers. The strength of classifiers is changed by distributing rewards and penalties. The rewards are given to those classifiers that attain the goals while the penalties are given to those that have low probability. This is carried out by the following processes: auction, clearinghouse (reinforcement & punishment) and taxation [1] – [3].

An auction competitive occurs between all classifiers in matching pool that match environmental message to determine the winner classifier by calculating the bid of all classifiers in matching pool, as shown in "(1)" [1] – [3].

$$B_i(t) = C_{bid}\, S_i(t)\, \sigma \qquad (1)$$

Where

C_{bid} : Classifier bid coefficient: positive constant $0 < C_{bid} < 1$

$S_i(t)$: Strength of classifier I at beginning of iteration t.

i : Classifier index.

σ : Specificity, which represent the number of non-'#' symbol 'don't care' in the condition part relative to its length.

The classifier with the highest bid is selected as the winner classifier.

Then the strength of the winner classifier is decreased by the amount of its bid as shown in "(2)". On the other hand the strength of the other classifiers is not changed [1] – [3].

$$S_i(t+1) = S_i(t) - B_i(t) + R_i(t) \qquad (2)$$

Where,

$R_i(t)$: Reward from the environment during iteration t.

$B_i(t)$: Classifier's bid during iteration t . Only the bid of the winner classifier is paid.

Taxation was used to avoid the classifier population from being confused with artificially high strength classifiers of little or without utility [1] – [3].

There are two types of taxes: Tax_{life} and $Tax_{bid.}$

Tax_{life} is a fixed rate applied to all classifiers that don't match environmental message and their strength is modified as shown in "(3)" [1] – [3].

$$S_i(t+1) = S_i(t) * (1 - Tax_{life}) \qquad (3)$$

On the other hand a fixed rate Tax_{bid} is applied to all classifiers that match environmental message except the winner classifier and their strength is modified as shown in "(4)" [1] – [3].

$$S_i(t+1) = S_i(t) - Tax_{bid} * B_i(t) \qquad (4)$$

From "(2)", "(3)" and "(4)" we conclude the general equation for updating the strength is shown in "(5)".

$$S_i(t+1) = (1 - Tax_{life})\, S_i(t) + R_i(t) - Tax_{bid} * B_i(t) \qquad (5)$$

Where:

$Tax_{life} = 0$ and $Tax_{bid} = 1$ for the winning classifier

$Tax_{life} = 0$ and $Tax_{bid} < 1$ for all classifiers that match environmental message except the winner classifier,

$Tax_{life} < 1$ and $Tax_{bid} = 0$ for all classifiers that don't match environmental message [3].

3) The Rule Discovery System

A GA is a search procedure. GAs are based on the idea of natural evolution. GAs simulates biological genetics as follows [1]:

- Structure

The information in GA is encoded into strings. These strings are like a chromosome in biology. Each string consists of features, and each feature has a feature value. These features and feature values are analogous to gene and allele respectively [1], [6], [7].

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:3, 2007

- Function

As with biological genetics, in GA two strings may combine and the result is a new individual. This new individual becomes one of the strings in the new population. This operation in GA is called crossover. On the other hand, the mutation in GA occurs when one feature value changes its value but this operation is rarely occurred [1]1, [6], [7].

The simple GA operators, involved in reproduction, are:
- Selection.
- Crossover.
- Mutation.

GAs can find new solution through search method in space of individuals. Finding good individual depends on some fitness function [1], [6], [7].

LCS requires an approach that can add a better rule into the system. Therefore, GA is used in the rule discovery system of LCS [1].

The rule discovery system uses GA to create new classifier by using GA operators; reproduction, selection, crossover and mutation. These new classifiers are then placed in the population and processed by the auction, clearinghouse, and taxation to properly evaluate their role in the system [1], [3], [4], [10].

In LCS theory there are two approaches that have been developed with the use of GA. There are:

- Michigan Approach

In this approach each classifier in the population represents a single individual.

- Pittsburgh Approach

In this approach a set of classifiers in the population represents a single individual.

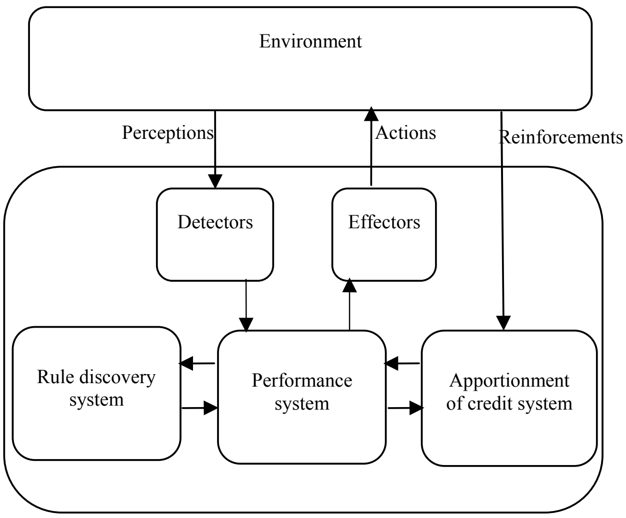

Fig. 1 Learning classifier system

The main cycle of LCS is as follows:
Step 0: Initialization of LCS:
Step 1: Activate the detector interface and post the environmental the received message to the message list.
Step 3: Perform the matching of all conditions of all classifiers in classifier store against the message list.
Step 4: All classifiers that matched compete in an auction and one shall be permitted to execute action of LCS. In the auction, the biding for all classifiers is performed.
Step 5: Collect taxes (collect life tax from all classifiers that do not match and bidding tax from matched classifiers).
Step 6: Pass the action of the classifier that won the clearinghouse which decrease the strength of the current winner by the amount of its bid value and increases its strength by its reward value.
Step 7: Record the winner classifier for this iteration.
Step 8: If steady state is reached: apply genetic algorithm.
Step 9: If not the last iteration Then go to step 4.
Step 10: Activate the effectors interface of LCS; i.e. perform the action it describes.
Step 11: Go to step 0.

In BB algorithm and GA the classifier's strength plays an important role in fitness function to direct the searching process [1], [3], [4], [10].

The interaction between robot and its environment can be illustrated through condition-action rules in classifier store of the classifier system. The condition represents the current state of the environment while the action represents what the robot must do [3], [8] ,[10].

D. Behaviours

We can define behavior as reaction between robot and its environment. In this reaction stage the robot senses that environment and acts on it by its sensors. There are two main behaviors [3]:

1) Basic Behaviours

Behaviors that do not partition into simpler behaviors [3].

2) Complex Behaviours

Behaviors that can be divided into simpler behaviors [3].

Complex behaviors can be built from the following simple behaviors in different ways [3]:

- *Approaching behaviour;* that is, a behavior of feeding that occurs when the robot is closer to still or moving object [3].
- *Chasing behavior;* that is, the robot follows still or moving object and tries to catch it [3].
- *Avoidance behavior;* that is, the robot avoids physical collision with an object of a given feature such as obstacles [3].

- *Escaping behavior;* that is, the robot moves far from an object with a given feature [3].

There are two types of behavior:

1) *Stimulus-response (S-R) behavior*, that is, the detectors is connected in a direct way with the effectors [3].

2) *Dynamic behavior*, where in this type, internal state is built between detectors and effectors [3].

III. Related Works

Applying Holland's LCS began by Goldberg. He used classifier system to study learning control of simulated gas pipeline control system. He set appropriate control of flow rates and pressures, and then determine if a leak occurs [1].

Other related work is, the Wilson's ANIMAT system. In this system Wilson use simple classifier system and learns it in a simulated environment: searching woods, seeking food and avoiding trees [1], [9].

Wilson developed a Boolean function learning that learns functions of multiplexer. This system is called BOOLE. [1].

Dorigo and Uwe Schnepf use a classifier system to train robot to follow a moving light source and to learn to avoid hot dangerous objects. They designed a two level hierarchy LCS [4].

John S. Bay proposed a new architecture, the distributed learning classifier system (DLCS), which generalizes the message passing behavior of the LCS from internal messages within a single agent to broadcast messages among multiple agents. [5].

Dorigo & Colombetti compared the performance of different architectures solutions for chase and escape behaviors using a monolithic architecture and a hierarchical architecture. They also compared the performance of different architectures solutions for chasing, escaping and feeding behaviors using a monolithic architecture and a hierarchical architecture [3].

Petr Musilek, Sa Li, and Loren Wyard-Scott tried to enhance Learning classifier system to develop efficiency of mobile robot navigation avoidance obstacles. Their system used genetic algorithm after a certain number of iterations to avoid local minima. In their system the rules will not be replaced in later generations and will not become a parent in the next generation. In their system the genetic operators are only applied to the action part [10].

IV. Proposed Research Methodology

The importance of robot comes from its ability to operate tasks with high efficiency and accuracy. Training a system in an unchanged environment is simple. On the other hand, in a changed environment the training will be more complicated.

This research suggests a simulated system. Generally, the following research methods are used.

First, an environment, an autonomous robot and the objects that it perceives are defined as shown in Fig. 2. These objects are:

A moving object, which moves toward the goal and its initial position, will be random.

The goal, which will have a fixed position.

Obstacles, which will have fixed positions.

The lair, which will have a fixed position.

In addition there is an emergency which could only be heard when the distance between a moving object and the goal becomes less than or equal to the known fixed distance.

Next the proposed behaviors that the simulated robot must execute in the environment are determined. For this stage we suggest complex behavior consisting of the following four basic behaviors:

Escaping Behavior:

Occurs when the distance between moving object and the goal is less than or equal to the known fixed distance. Then the simulated robot escapes toward the lair.

Avoidance Behavior:

When the simulated robot perceives obstacles it avoids physical contact with them.

Chasing Behavior:

The simulated robot follows the moving object and each cycle moves one step toward the moving object.

Approaching Behavior:

When an emergency is heard the simulated robot changes its behavior to approaching behavior, that is, it will move two steps in each cycle toward the moving object to catch it before reaching the goal.

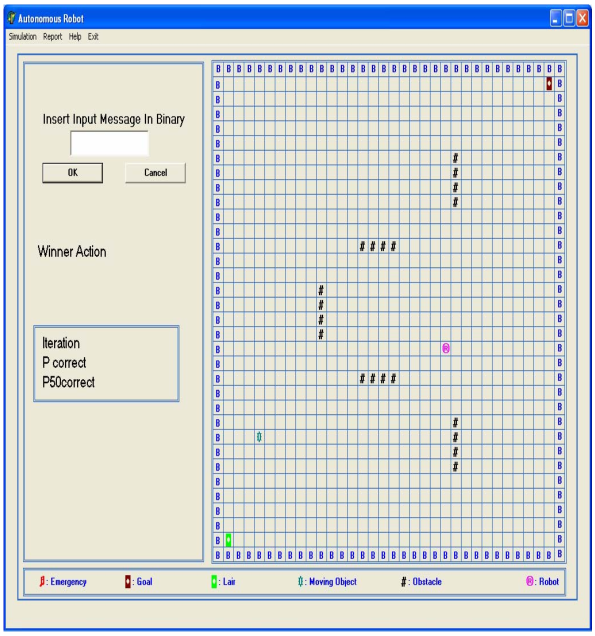

Fig. 2 Snapshot of the simulated environment

We also design the control system architecture for the simulated robot, by using distributed LCS system with hierarchical architecture; as shown in Fig. 3. This system consists, of a set of five classifier systems that is organized in three-levels, where the lower level consists of three LCSs that is responsible for interacting the system with environment and competitiib between behaviors, and levels two and three consist of one LCS that achieve coordinate between behaviors and determine final action of the proposed system.

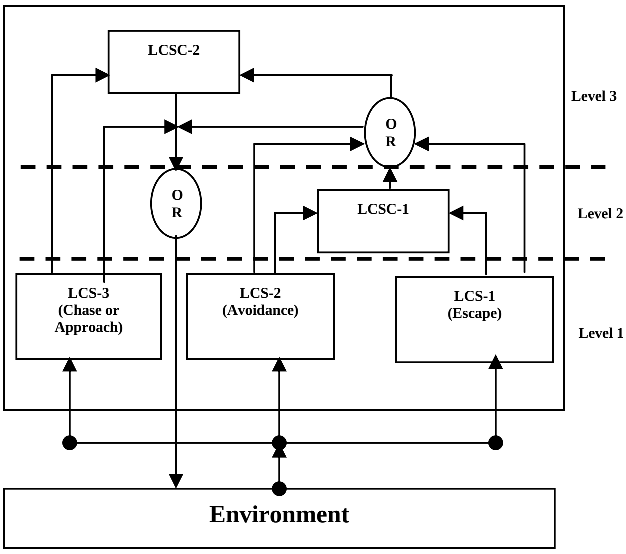

Fig. 3 Structure of the proposed system

In representing the classifiers in population all LCSs used the standard ternary alphabet strings {0,1,#} with fixed length. On the other hand, the representation of the environmental message, input message and output message of each LCS the binary alphabet strings {0,1} are used with fixed length.

In designing stage a strategy is used to train the autonomous robot to determine the correct behavior. The first step is to train the simulated robot to achieve the four basic behaviors through training each LCS independently. The next step is to train the switch (coordinator) to determine the target effectors.

Finally a number of experiments are executed and analyzed the effect of the simulated robot's performance is analyzed. In the proposed system, performance of the simulated robot is measured as the ratio of the number of correct moves to the total number of moves performed from the beginning of the simulation.

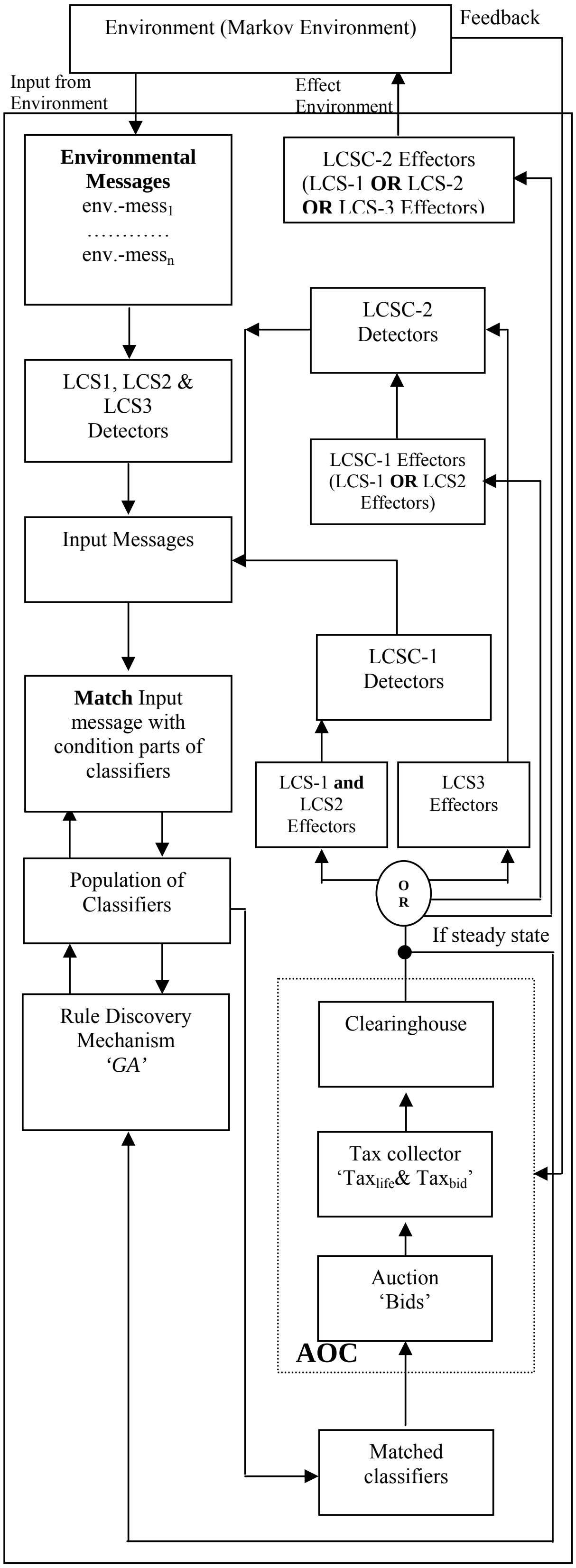

Fig. 4 Interaction of the proposed system with the environment: Learning Mode

V. Properties of the Proposed System

The main properties of the proposed system are as follow:

A. Markov Environment

In the proposed system the simulated robot immediate senses are all information that are necessary to select the best action.

B. Michigan Approach

In this study the proposed system uses Michigan approach, that is, each classifier in the population of each LCSs represents a single individual.

C. Modular Shaping

In the proposed system modular shaping is used as shaping policy, to that is the basic LCSs will be trained, and then, after they have reached a good performance level, they will be fixed.

D. Complex Behaviour

In the proposed system a single action is made to perform four tasks simultaneously. That is, the simulated robot might distinguish between these behaviour and succeed in choosing the correct behaviour.

E. Dynamic Behaviour

In the proposed system the reactive responses are not connecting detectors to effectors in a direct way, but there are some kinds of internal state to mediate between input and output of the simulated robot.

VI. Conclusion

This paper is concerned with the developments of the simulated robot behaviors by using distributed LCS as control system. Our aim is to enhance the speed of the learning process of the robot and make it able to choose the correct behavior in a complex environment. Finally, this will increase the efficiency and the accuracy of the robot behaviors.

References

[1] Goldberg, D. E. "Genetic Algorithms in Search, Optimization, and Machine Learning". Addison-Wesley, Reading, Mass. .1989.

[2] Luger & Stubblefield "Artificial Intelligence (Structures and strategies for complex problem solving) " Addison Wesley Longman, Inc., 1998.

[3] Marco Dorigo & Marco Colombetti , "Robot shaping (An experiment in behavior engineering) " The MIT Press , Cambridge, Massachusetts, London, England, 1998.

[4] Marco Dorigo, "Genetic-Based Machine Learning and Behavior-Based Robotics: A New Synthesis", 0018-9472/93, *IEEE Transactions on System, Man, and Cybernetics*. Vol. 23, No. 1, January/ February 1993.

[5] John S. Bay, "Learning Classifier Systems for Single and Multiple Mobile Robots in Unstructured Environments" *Mobile Robots X, Philadelphia, PA*, Nov. 1995, pp. 88-99.

[6] D. E. Goldberg. "Sizing populations for serial and parallel genetic algorithms" – Third international conference on genetic algorithms, 1989.

[7] Alan C. Schultz. "Learning robot behaviors using genetic algorithms "- The international symposium on robotics and manufacturing , 1994.

[8] Gabriel Robert, Pierre Portier, Agens Guillot, "Classifier Systems as 'ANIMAT' Architectures for action selection in MMORPG" – AnimatLab, Laboratorie d'Informatique de Paris, France, 2002.

[9] Larry Bull, "Learning Classifier Systems: A Brief Introduction" – Faculty of Computing Engineering & Mathematical Sciences, University of the West of England, U.K, 2003.

[10] Petr Musilek, Sa Li, and Loren Wyard-Scott, "Enhanced Learning Classifier System for Robot Navigation" - Department of Electrical and Computer Engineering, University of Alberta, Canada, 200.

Saeed Mohammed Baneamoon received the M.Sc. degree in computer engineering from Technical University, Sofia, Bulgaria in 1996, the M.Sc. degree in computer science from University of Technology, Baghdad, Iraq in 2003 and is currently a Ph.D. candidate in School of Computer Sciences, Universiti Sains Malaysia, USM, Penang, Malaysia. His research interest focuses on the field of Artificial Intelligence and Robotics.

Rosalina Abdul Salam is a senior lecturer at the School of Computer Sciences , Universiti Sains Malaysia and a member of Artificial Intelligence Research Group. She received her Bachelors degree in Computer Science in 1992 from Leeds Metropolitan University, United Kingdom. She was a system analyst in Intel Penang, from 1992 to 1995. She returned to United Kingdom to further her studies. She received her Masters degree in Software Engineering from Sheffield University, United Kingdom in 1997. She completed her PhD in 2001 from Hull University in the area of computer vision.

She has published more than 60 papers in journals and conferences. She is a member of International Computational Intelligence Society and World Enformatika Society. Recently she joined the editorial board of the International Journal of Computational Intelligence and the International Journal of Signal Processing.

Presently, she is continuing her teaching, graduate supervisions and her research. Her current research area is in the area of artificial intelligence, image processing and bioinformatics applications. The most recent project that she is working is on underwater images and cellular images.

Abdullah Zawawi Talib obtained B. Sc. (Hons.) in Mathematical Sciences from the University of Bradford, Britain in 1983, M. Sc. in Computing Science from the University of Newcastle upon Tyne, Britain in 1985 and PhD in Computer Science from the University of Wales in Swansea, Britain in 1995. He started his career as a university lecturer at the University of Science Malaysia in 1986 and promoted to the current position as an Associate Professor in 2003. He has also served as chairperson for computer science, information systems and computing science programmes at the school of computer sciences. Currently he is the deputy dean for academic and student development at the same school. His research interests include graphics and visualization, geometric computing, computational modelling and intelligent systems. He has published over 50 papers in conferences and journals, and served as conference organising chair, member of programme committees for a number international/national conferences. He has also reviewed several conference and journal papers.