ENHANCED DISTRIBUTED LEARNING CLASSIFIER SYSTEM FOR SIMULATED MOBILE ROBOT BEHAVIOURS

SAEED MOHAMMED SAEED BANEAMOON

UNIVERSITI SAINS MALAYSIA

2010

ENHANCED DISTRIBUTED LEARNING CLASSIFIER SYSTEM FOR SIMULATED MOBILE ROBOT BEHAVIOURS

by

SAEED MOHAMMED SAEED BANEAMOON

Thesis submitted in fulfillment of the requirements
for the degree of
Doctor of Philosophy

November 2010

ACKNOWLEDGEMENTS

I am deeply indebted to my supervisor, Assoc. Prof. Dr Abdullah Zawawi Talib for his unfailing interest, guidance and wisdom to complete this thesis. I am also extremely grateful to my co-supervisor, Professor Dr. Rosalina Abdul Salam for her supervision and guidance during all of the work toward this thesis. Her support, confidence, valuable comments and suggestions often guided me in my research.

I would like to thank Staff of the School of Computer Sciences at Universiti Sains Malaysia (USM) for this opportunity to gain experience and to broaden my horizons.

I am especially indebted of the Hadramout University of Science and Technology (HUST) for granting me of scholarship throughout the study at the Universiti Sains Malaysia (USM).

I would like to thank Puan Faten for proof reading and her comments. Thanks also to my friends Ali Binsama, Dr. Yaser Aljahwary, Dr. Ryad Albatani, Dr. Hassan Alfadly, Dr. Khaled Binmukhashin, Dr. Atef Altamimy, Dr.Hassan Alkafaf and many more that I have acquired throughout my years.

Finally, and most important, I would like to thank my parents, my sisters and brothers and their families, and my own little family – my wife Dr. Mirfat and my sons Manaf and Munef, for their unconditional love and constant support over the years is something that I cannot thank them enough for. I also thank them for having faith in me, and my capabilities, and the advice they offered to me over all these years.

TABLE OF CONTENTS

CHAPTER 2 – BACKGROUND

CHAPTER 3 – LITERATURE REVIEW

CHAPTER 4 – SIMULATED ROBOT SYSTEM

CHAPTER 5 – ENHANCEMENT OF BUCKET BRIGADE ALGORITHM

CHAPTER 6 – APPLYING ENHANCED STEADY STATE FOR CALLING GENETIC ALGORITHM

CHAPTER 7 – ENHANCEMENT APPROACHES OF PERFORMANCE SYSTEM

CHAPTER 8 – CONCLUSION AND FUTURE WORK

APPENDICES

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACS	Anticipatory Classifier System
AI	Artificial Intelligence
AOC	Apportionment of Credit System
BBA	Bucket Brigade Algorithm
BBR	Behaviour-Based Robotics
DH	Default Hierarchy
DLCS	Distributed Learning Classifier System
DP	Dynamic Programming
EB	Effective Bid
ER	Evolutionary Robotics
FCS	Fuzzy Classifier System
FL	Fuzzy Logic
FSM	Finite State Machine
GA	Genetic Algorithm
GBML	Genetic-Based Machine Learning
ICS	Interactive Classifier System
LCS	Learning Classifier System
ML	Machine Learning
NCS	Neural Classifier System
NN	Neural Network
RL	Reinforcement Learning
S-R	Stimulus – Response
TCS	Temporal Classifier System
TD	Temporal Difference
XCS	eXtended Classifier System
ZCS	Zeroth-Level Classifier System

SISTEM PENGELAS PEMBELAJARAN TERAGIH YANG DIPERTINGKAT UNTUK KELAKUAN ROBOT BERGERAK TERSIMULASI

ABSTRAK

Empat kelakuan asas robot bergerak adalah mengejar, mendekati, mengelak dan melepaskan diri. Masalah utama dalam sistem robot adalah dalam pemilihan kelakuan yang betul. Tujuan penyelidikan ini adalah untuk mengatasi masalah pemilihan kelakuan. Tesis ini mencadangkan kaedah-kaedah yang dapat mengatasi masalah pemilihan kelakuan yang baik dan masalah penghapusan terhadap kelakuan yang baik. Tesis ini juga menumpukan perhatian kepada masalah maklumat yang hilang, menyelesaikan masalah ayunan antara kelakuan yang betul dan salah, dan menyentuh kecekapan yang rendah dalam pemetaan input kepada kelakuan yang betul. Sistem Pengelas Pembelajaran Teragih (DLCS) yang mengandungi lima Sistem Pengelas Pembelajaran (LCS) dengan seni bina berhierarki tiga peringkat digunakan. Algoritma Briged Baldi (BBA) yang dipertingkat dibangunkan untuk mengelak masalah pemilihan pengelas dengan nilai kekuatan yang tinggi tetapi dengan kelakuan yang salah. Satu pendekatan yang mengesan nilai keadaan mantap untuk memanggil algoritma genetik (GA) dicadangkan untuk mengatasi masalah penghapusan pengelas yang baik dan perangkap minimum tempatan. Akhirnya, penyelesaian yang cekap untuk melindungi pengesan, menyokong pembentukan hierarki lalai dan ayunan antara tindakan yang betul dan salah diperkenalkan untuk mengelak kegagalan prestasi, pengitlakan pengelas yang mempunyai kebolehan untuk meliputi syarat spesifik dan umum, dan kehilangan pengelas yang diingini. Secara keseluruhan, pendekatan-pendekatan yang dipertingkat berprestasi baik dan proses pembelajaran yang dipertingkat yang dicadangkan dalam kajian ini menjadikan pembelajaran robot lebih berkesan. Robot tersimulasi ini diuji dan keputusannya menunjukkan bahawa robot berkenaan berprestasi lebih baik dengan empat kelakuan asas. Robot tersimulasi ini juga diuji pada banyak kelakuan kompleks yang merupakan mana-mana gabungan empat kelakuan asas dan keputusannya menunjukkan bahawa robot berkenaan juga berprestasi lebih baik dengan jenis kelakuan sebegini.

ENHANCED DISTRIBUTED LEARNING CLASSIFIER SYSTEM FOR SIMULATED MOBILE ROBOT BEHAVIOURS

ABSTRACT

The four basic behaviours of mobile robot are chasing, approaching, avoiding and escaping. The main problem in robotic system is in selecting the correct behaviour. The aim of this research is to overcome the behaviour selection problem. This thesis proposes methods that can overcome the problems of good behaviour selection and good behaviour deletion. It also addresses the problem of missing information, solves the problem of oscillating between correct and incorrect behaviours, and addresses the low efficiency in mapping the input to the correct behaviour. A Distributed Learning Classifier System (DLCS) consisting of five Learning Classifier Systems (LCS) with hierarchical architecture of three levels is used. An enhanced Bucket Brigade Algorithm (BBA) is developed to avoid the problem of choosing classifiers with high strength value but with incorrect behaviour. An approach that detects steady state value for calling genetic algorithm (GA) is proposed to overcome the problems of good classifiers deletion and the local minima trap. Finally, efficient solutions for covering detectors, supporting default hierarchies formation and the oscillation between correct and incorrect action are introduced to avoid performance failure, generalisation of classifiers that have the ability to cover the specific and general conditions, and loss of desirable classifiers respectively. Overall, the enhanced approaches performed well and the enhanced learning processes proposed in the current study makes robot learning more effective. The simulated robot is tested and results have shown that it performs better with the four basic behaviours. The simulated robot is also tested on many examples of a complex behaviour which is any combination of the four basic behaviours and the results have shown that it performs better with this type of behaviours as well.

CHAPTER 1

INTRODUCTION

1.1 Introduction

The development of mobile robot behaviours that allows the robot to adapt itself to the real world is an important area of robotic research. Robotics system has considered different types of basic behaviours such as wandering, chasing, approaching, avoiding and escaping. The combination of two or more of these basic behaviours makes a complex behaviour.

However, robotics system does have undesirable limitations in its efficiency and accuracy in performing complex behaviour such as difficulty in seeking a goal location and slow in avoiding obstacles. Machine Learning (ML) is an area of artificial intelligence that provides solutions to overcome these limitations and allows robots to successfully adapt to their environment (Dorigo and Colombetti, 1998).

ML focuses on improving the applications such as speech recognition, medical diagnosis, computer vision and robot control through designing programs that have an ability to increase performance and experience of these applications (Mitchell, 2006; Luger and Stubblefield, 2008). The robotics system that uses ML must have systemic procedures for designing, developing and testing in order to make it viable for use in a practical area (Dorigo and Colombetti, 1998).

Genetic algorithm (GA) has been used for many ML applications including in designing a robot control system. GA is an adaptive search procedure and its intensity refers to parallel searching of the best solution from a population of candidate solutions used in the simulation. In robot simulation, each solution in the population of solutions can represent behaviour of the robot. Applying GA operators will increase the probability of improving the performance of the population of behaviours through finding a good behaviour. As a result, this good behaviour is then applied to the real world (Goldberg, 2007).

Genetic-based machine learning (GBML) is a class of machine learning algorithm that uses GA. GBML techniques are able to help robotic systems to perform their actions with efficiency and accuracy through its flexibility and its mechanisms of structural adaptation that are better for a robot controller. It can effectively interact with the environment and allows the robot with learning ability to carry out its task (Goldberg, 2007).

Learning Classifier System (LCS) is a class of GBML systems which uses Bucket Brigade Algorithm (BBA) and GA as learning mechanisms to produce an adaptive system. The role of BBA in a classifier system is to assign a more correct behaviour of the system. This occurs by changing the strength value of all classifiers in classifier store so that the matching classifiers can be classified based on their usefulness. On the other hand, GA is used to inject new classifiers to the classifier store by using its operators (Dorigo and Colombetti, 1998; Kovacs, 2002a; Lanzi *et al.,* 2000a; Goldberg, 2007; Lanzi, 2008). The architecture of LCS makes it possible to be used as main components in designing the simulated control system for robot in an effort to determine which behaviour must be used (Dorigo and Colombetti, 1998; Lanzi *et al.*, 2000a; Holmes *et al.*, 2002; Lanzi, 2008).

Hence, this thesis will focus on the challenge of learning simulated robot control system based on Distributed Learning Classifier System (DLCS) with hierarchical architecture to perform a complex behaviour which consists of any combinations of the four basic behaviours.

1.2 Motivation

Nowadays mobile robot is extensively used in many different industries. New applications that use a robot as a tool need to be developed using the available technology. Therefore, improvement of robot behaviours in performing complex behaviours in real world is an important area of robotics research.

In robotics research, it is imperative to design a simulated system for the robot and train it to perform complex tasks in a simulated environment with efficiency and accuracy, so that designers can avoid problems that may appear when designing the system directly in a real world environment.

On the other hand, adaptability or learning complex tasks in a simulated environment is useful in making the robot adaptable in a variety of domains. As a result, robotics technology can be exploited in many new applications in different areas in the real world.

Moreover, it becomes clearer that in current robotics research, the learning process techniques play an important role in increasing the behavioural quality of the control system for simulated and real robots.

1.3 Problem Statement

In all robotics applications, a robot is required to accomplish specific tasks. Therefore, the robot must be able to adapt with its environment by having the capability to explore and learn its environment (Dorigo and Colombetti, 1998).

The main problem in a robotic system is in selecting the correct behaviour in performing a complex behaviour. Therefore, an important challenge for researcher is to produce robots that can continually adapt to their environment and tasks, and can constantly improve their performance (Dorigo and Colombetti, 1998).

The Distributed Learning Classifier System (DLCS) is a technique that improves the efficiency of the robot in performing a complex behaviour (Dorigo and Colombetti, 1998). However, the existing methods are mainly limited to decreasing the number of reinforcement cycles necessary for learning a given task. On the other hand, the problem of increasing convergence times in the case of large search problems has not completely been solved. Furthermore, local minima trap, loss of desirable classifiers, classifiers that are too general, loss of performance or the need for more training in dynamic environments are some of the major areas for improvement. As a result, the DLCS does not learn to solve complex behaviour problems very well and is still the subject of considerable research (Katagami *et al.,* 2003; Cazangi *et al.*, 2003; Musilek *et al.*, 2005).

1.4 Objectives

In this research, LCS is used as a learning paradigm for the development of robot behaviour. A simulated robot in a simulated environment is suggested as it is proven to be useful in testing simulated design options that are powerful to turn to in the real world.

This research addresses some issues in mobile robot based on LCS. The first issue is that in LCS, each classifier consists of condition/action part and strength value of type real that determines the winner classifier. Competition occurs between all classifiers in the population of matched classifiers in determining the winning classifier that will affect the environment. This competition is based on the strength value. The classifier with high strength value is selected to post its action. Sometimes, classifier with high strength value is selected but its action is incorrect. The drawback of the existing BBA is the difficulty in avoiding such problem. The second issue is a system in a steady state has strength and bid values that are unchanging in time. The third issue is that in LCS, every time the detectors receive the environmental message or the input message and the performance system part matches the environmental message or the input message with the condition parts of all classifiers in the classifier store. If there is no matching, it means that the environmental message or the input message is not covered by any classifier. Therefore, a covering detectors problem occurs. The fourth issue is the problem of oscillation occurs when “don’t care” (#) symbol is used in representing the condition parts of the classifiers, the “don’t care” (#) symbol matches either 0 or 1. Oscillating classifiers are those classifiers that can receive the rewards when matched by some messages and the punishments when matched by some other messages. The problem of oscillation causes the strength values of those classifiers to oscillate. The last issue is Default Hierarchy (DH) which is defined as rule sets that represent knowledge of learning classifier system regarding its environmental states. DH covers most of the possible environmental states because the use of “don’t care” (#) symbol that matches either 0 or 1 makes the classifier to be more general. Rules with many “don’t care” (#) symbol have the ability to cover the specific and general conditions.

Therefore, the objectives of the research relate to improving the efficiency and the accuracy of the robot behaviour in choosing its correct task. In more detail, the objectives are:

1. To investigate and propose an enhancement of the Bucket Brigade Algorithm (BBA) in order to increase the correct selection of robot action

2. To improve the approach of determining steady state values for calling GA in order to address the problem of the deletion of good behaviour.

3. To introduce an efficient solution of covering detectors problem in order to address the problem of missing information.

4. To suggest a method for solving the problem of oscillation between correct and incorrect actions.

5. To propose an efficient approach for supporting default hierarchies formation in order to address the low efficiency in mapping the input to the correct output behaviour.

1.5 Scope and Limitations

Robot deals with different areas in the real world that requires high effectiveness and accuracy in interacting to its environment. This research concentrates on enhancing the performance of the interaction between mobile robot and its environment by focusing on learning process enhancement for simulated mobile robot in order to accomplish complex behaviours. The simulated environment is two dimensional (2-D), and the simulated robot that perceives a moving object moves towards the goal and the initial position of the moving object is random. The goal, obstacles also and the lair (a place in which a robot seeks concealment) have fixed positions.

1.6 Research Approach

In order to accomplish the research objectives, the steps involved in this research are as shown in Figure 1.1.

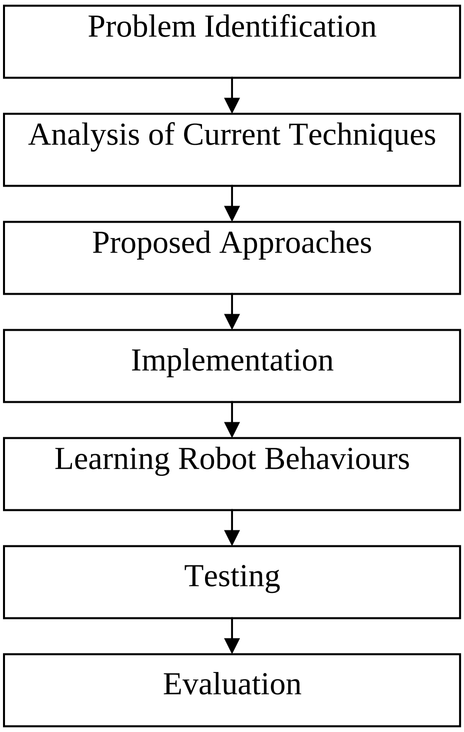

Figure 1.1: Research approach

1.6.1 Problem Identification

In this step, the problem of complex behaviours is investigated in order to improve the performance of the simulated robot in choosing the correct action through addressing the problem of selecting the incorrect action, cancellation of good behaviour, missing information, oscillating between correct and incorrect behaviour and the low efficiency in mapping the input to the correct output behaviour.

1.6.2 Analysis of Current Techniques

This step focuses on current methods and techniques, and is concerned with the complex behaviour of the robot. In particular, this research focuses on the LCS-based robot controllers. Based on the literature review, there are limitations in the existing approaches. Therefore, the current research will address these limitations.

1.6.3 Proposed Approaches

This step is concerned with the proposed methods and approaches to enhance the solution of the problems that have been identified from the existing techniques in order to achieve the objectives of the research. Therefore, in this research, the proposed methods are based on the improvement of LCS's processes, and this will be done by:

- Enhancing the BBA.
- Proposing an approach to detect steady state value for calling GA. In LCS, normally GA is used because the power of GA derives from its ability to

achieve parallel search for good solutions from a vast amount of candidate solutions (Lanzi, 2008).

- Suggesting solutions for the problems of covering detectors, oscillation, and default hierarchies formation.

1.6.4 Implementation

In this step, the proposed approach of the previous step will be implemented using an object-oriented programming language. Several algorithms that enhance the correct behaviour selection of the robot will be implemented in order to achieve the objectives of the research.

1.6.5 Learning Robot Behaviours

In this step, the enhanced efficiency and accuracy of a robot in performing correct behaviour selection will be obtained through firstly, training the simulated robot to achieve the four basic behaviours by training each LCS in the lower level independently, secondly training each LCS in the upper level to switch behaviours and to determine the final behaviour, and finally fixing the system after all the LCSs in the whole simulated system have reached a good performance level.

1.6.6 Testing

The role of testing is to determine whether further improvement is needed for the system performance if the enhanced simulated robot does not produce satisfactory results. Therefore, the system performance will be assessed based on the accuracy of the correct behaviour selection of the robot so that correct behaviours are achieved.

1.6.7 Evaluation

This step is concerned with examining the performance efficiency of the proposed system through evaluation of the results of the proposed approach against the existing approaches. In this regard, performance measure and statistical analysis will be used to evaluate the result to ensure that the proposed approach is thoroughly evaluated. These methods that will be used are as follows:

1. *The Performance Measure:* the 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. A correct move of the robot is the shortest move with respect to the current goal. This will be used to examine the efficiency of the simulated robot in choosing correct behaviour before and after the enhancement. For example, if after 14 iterations the chase behaviour has been active for 6 iterations with 5 correct moves, and the escape behaviour has been active for 8 iterations with 4 correct moves, then the chase behaviour performance is 5/6, the escape behaviour performance is 4/8, and the global performance (5+4)/(6+8)=11/14 (Dorigo *et al.*1994a; Dorigo and Colombetti, 1998).

2. *Statistical Analysis:* regression is used to evaluate the efficiency of the simulated robot behaviours before and after the enhancement. Regression measures the degree of relationship between a single dependent variable (Performance) which is considered to be a function of independent variable (Iterations) (Pallant, 2007).

1.7 Research Contributions

This thesis concentrates on the problem of learning mobile robot to perform its correct tasks in a simulated environment and enhancing approaches for learning approximate optimal solutions to accomplish complex behaviour which is any combinations of the four basic behaviours.

Therefore, the study focuses on the use of learning classifier system as a main component in designing a simulated control system for robot to achieve complex behaviours. The two major contributions of the research are:

- An enhanced algorithm of the Bucket Brigade Algorithm (BBA) that works in LCS that avoids the problem of classifier with high strength value but has chosen the incorrect behaviour.

- An approach that detects steady state value for calling genetic algorithm (GA) that overcomes the problems of deletion of good classifiers and the local minima trap.

Other contributions are:

- An efficient solution for covering detectors problem that avoids the performance failure of the simulated system.

- A method for oscillation between reward and punishment problem that avoids loss of desirable classifiers which leads to a lower performance of the simulated robot relating to the complex behaviours.

- An approach that supports default hierarchies formation that can overcome generalization of classifiers and thus, the simulated robot is more effective in mapping the input to the correct output behaviour.

1.8 Structure of the Thesis

The thesis is organized into eight chapters. Chapter 2 introduces the background of this research. This chapter describes behaviour-based robotics, machine learning, reinforcement learning, genetic algorithm and genetic-based machine learning. The chapter also presents learning classifier systems in sufficient detail for better understanding of the subsequent chapters. Next, the idea and the architecture of LCS for robot control system is introduced. Finally, learning strategy of DLCS for robot control system is described.

Chapter 3 presents several approaches related to robot controller based on simple and distributed LCS. It illustrates the enhancement approaches of the BBA that have been suggested by different researchers. It also describes the approaches for calling GA. Then, it describes the approaches relevant to problems of covering detectors, oscillation and default hierarchies formation. Finally, this chapter discusses the limitations of the existing approaches that motivate the proposed research.

Chapter 4 presents the proposed method. The simulated robot system, learning mode of the simulated system, the algorithm of the simulated system, and properties of the proposed system are reviewed.

Chapter 5 details a complete enhanced bucket brigade algorithm proposed in this research. The chapter sets the parameters used in the experiment. The experimental result is then discussed and the proposed enhancement is compared with the closest research works on the robot controller based on LCS. Finally, a chapter summary is given.

Chapter 6 presents enhanced approach to determine the steady state values for the strength and the bid of classifiers for calling a genetic algorithm (GA). The experimental results are shown and the proposed enhancement is also compared with the closest research work on the robot controller based on LCS. Finally, a chapter summary is presented.

Chapter 7 proposes a set of enhancement of the performance system part in LCS by suggesting a set of efficient solutions for different problems and comparing them with each other. These problems are: covering detectors problem, oscillation problem and default hierarchies formation problem. The experimental result is then discussed and the proposed enhancement is also compared with the closest research work on the robot controller based on LCS. Finally, a chapter summary is given.

Chapter 8 concludes the thesis and the work presented in this thesis. This chapter also discusses the contributions of the research work and presents limitations and future directions that can be further taken from this work.

CHAPTER 2

BACKGROUND

2.1 Introduction

Robots have been used to perform difficult tasks in a complex and unstructured environment such as environmental clean-up, mine removal, fire-fighting and rescue operations. A robot can be controlled by a human operator but nowadays most robots are controlled by computer programs. There are many variations for a robot control program, but they are often too difficult to be understood or too complex to be adequately managed with static behaviours hand-coded by a programmer. In order to develop an intelligent robot, its control system must be able to perform complex tasks in real time (Dorigo and Colombetti, 1998).

The standard Artificial Intelligence (AI) approach in building a robot control system is to vertically decompose its control into various units according to their functionality. This vertical decomposition includes the following units: perception, modeling, planning, task execution and motor control as shown in Figure 2.1 (Brooks, 1986a).

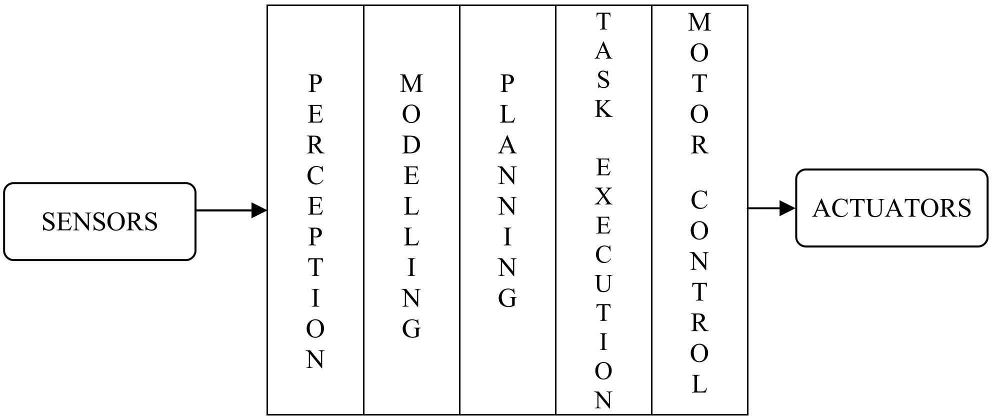

Figure 2.1: Vertical decomposition into functional modules

Brooks of MIT (1986a) founded the work which was known as “Behaviour-Based Robotics”. Behaviour is the interaction between the robot and its external environment. In this reaction stage, the robot senses the environment and acts on it by its sensors. There are two main structures of behaviours depending on the possibility of dividing the behaviour into simpler behaviour. These two behaviours are basic and complex behaviours. Basic behaviours are not structured into simpler ones such as wandering, feeding and avoiding. But complex behaviours can be divided into simpler behaviours. Dorigo and Colombetti, (1998) have considered four kinds of basic behaviours in which complex behaviour can be built from. These basic behaviours are:

1) ***Approaching Behaviour*** **-** a behaviour of feeding that occurs when the robot is closer to still or moving object.
2) ***Chasing Behaviour*** **-** the robot follows still or moving object and tries to catch it.
3) ***Avoiding Behaviour*** **-** the robot avoids physical collision with an object of a given feature such as obstacles.
4) ***Escaping Behaviour*** **-** the robot moves far from an object with a given feature.

Brooks (1986a, 1986b) proposed a horizontal decomposition based on task achieving behaviours as the organizational principle. The method decomposes the desired intelligent behaviour into a set of simpler behaviours. Each behaviour has its own computational system for sensing, reasoning and acting capabilities as shown in Figure 2.2. After each simple behaviour has been implemented and tested, they are then composed into a more complex behaviour in the system.

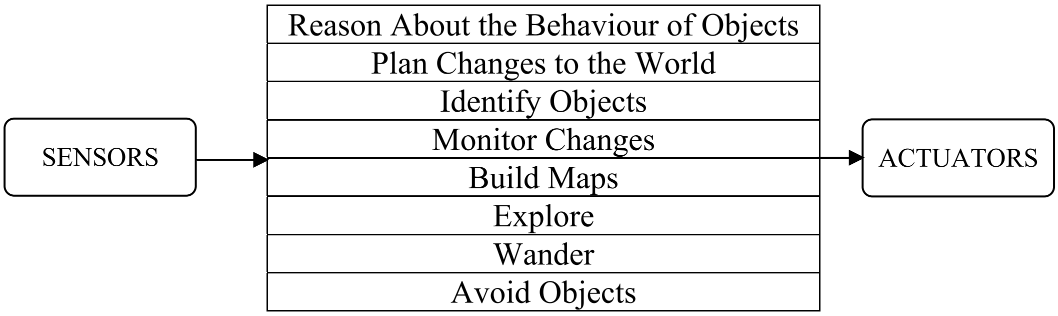

Figure 2.2: Horizontal decomposition into behavioural modules

This horizontal decomposition creates a different architecture of robot controller layer by layer which is called subsumption architecture as shown in Figure 2.3. Each layer is decomposed into a set of processing modules. Each processing module is a Finite State Machine (FSM) that has the ability of holding some data structures. Each FSM had input lines and output lines that could be connected to one or more FSM, or to sensors or actuators. These processing modules run synchronously and with the ability to monitor and to influence the behaviours of the layer below it. These behaviours can interact with each other. In this architecture, inputs can be suppressed and outputs can be inhibited. This is the mechanism where higher-level layers subsume the role of lower-level layers (Brooks, 1986a).

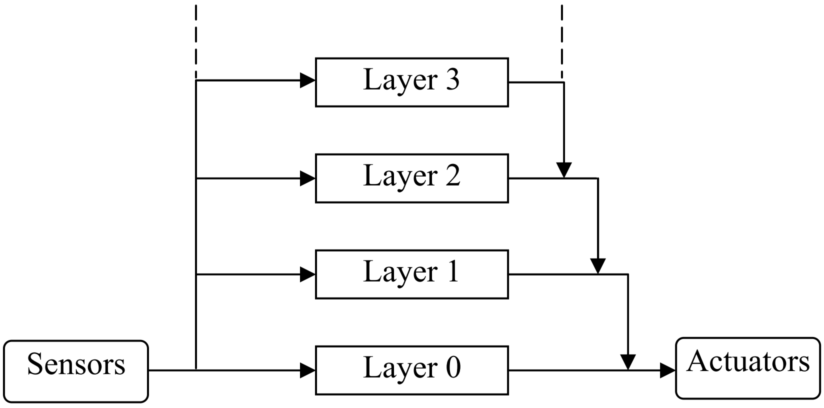

Figure 2.3: The subsumption architecture

Tinbergen has developed the ethological model of animal behaviour. The Tinbergen model is described as a hierarchy of behavioural modules called ***instinct centres***. Each instinct centre is decomposed into more granularity behavioural sequences represented by instinct centres at the next lower level. The instinct centres at the same level of the hierarchy compete against each other in order to become active by sending inhibitory signals corresponding to its level of excitation. Each instinct centre is influenced by the excitation coming from inner and outer sensors which are strictly related to a so-called ***innate releasing mechanism***. The excitation is released if the threshold value has been achieved. In this way, the releasing mechanism serves to prevent the random behaviour in the behavioural organization (Dorigo and Colombetti, 1998).

Dorigo *et al.* (1994a), and Dorigo and Colombetti (1998), have integrated ideas developed in the disciplines of ethology and behaviour-based robotics represented by the works of Tinbergen and of Brooks to build a robot controller. The simulated robot control system described in this thesis is the closest to the work done by Dorigo *et al.* (1994a), and Dorigo and Colombetti (1998), which consists of three layers as shown in Figure 2.4. Layer 0 is a behaviour layer which consists of a set of independent processing modules; each module is responsible for a simple task in the subsumption architecture. Behaviour layer includes four kinds of basic behaviours: moving to the goal, avoiding obstacles, catching the object and escaping from the danger. Layers 1 and 2 consist of one processing module each; these layers are behaviour-managing layers that adjust the behaviour of the robot according to the current task, and decide which process will control the robot and the final output to the actuator.

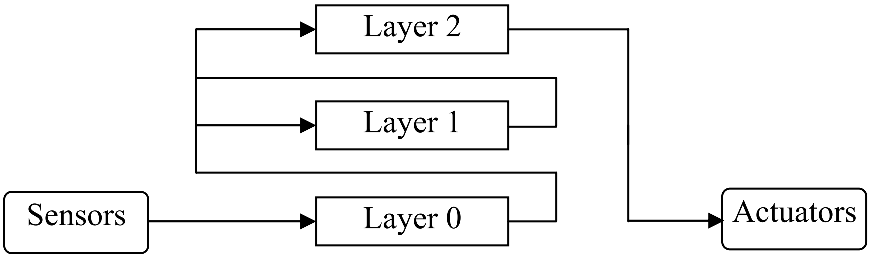

Figure 2.4: The proposed layered architecture

2.2 Behaviour-Based Robotics

Behaviour-Based Robotics (BBR) links the areas of Artificial Intelligence, Engineering and Cognitive Science. BBR aims to develop approaches for designing intelligent physical and also simulated robot controller, and to use robotics to model and better understand biological systems (usually animals ranging from insects to humans). BBR controllers consist of a set of behaviours that achieve goals such as avoiding collisions, moving to the goal and escaping from the danger (Mataric, 1999).

Behaviour-Based Robotics (BBR) emerged from dissatisfaction with standard approach in the design of intelligent robot controller which breaks up robot controllers into functional units with each unit representing a different activity (Brooks, 1986a). Therefore, instead of designing based on the various functions, robot controller is designed based on the behaviours to be achieved. Brooks (1986a, 1986b) suggested that each behavioural unit should be independent of other behaviours. A set of these independent behaviours can be composed into a more complex behaviour in the system.

BBR has proven a variety of standard robotic capabilities including obstacle avoidance, navigation, terrain mapping, following, chasing/pursuit, object manipulation, task division and cooperation, and learning maps, navigation and to walk (Mataric, 1999). Applications of BBR have continued to grow in different areas, in particular the critical areas such as demining and rescue operations.

Machine Learning (ML) is one of the AI methods used broadly in the field of robotics. Variations and adaptations of ML, and in particular reinforcement learning, have been effectively applied to BBR in the design of the robotic model in simulated and real environment. BBR have demonstrated learning to walk, navigate, divide tasks, score goals in robot soccer and others. Others methods continue to be actively explored and applied to BBR as their role in animal modelling and practical applications such as Artificial Life, Evolutionary Computation / Genetic Algorithms, Fuzzy Logic, Vision And Learning and Multiagent Systems (Mataric, 1999).

2.3 Machine Learning

Learning process is applied to acquire the system knowledge resulting from experience in that environment. The “system” mentioned here can be biological or artificial (computer). This enables the system to perform the same task more effectively and more efficiently than the previous time (Wootinun, 2003).

Machine learning (ML) is a subfield of artificial intelligence (AI) that aims at designing a computer program which can develop algorithm and techniques based on example data or the exploitation of the past experience. As a result, intelligent systems that are able to learn and adapt with the changing environment can be established. These systems are able to find an appropriate future solution of a given problem such as in recognition, diagnosis, robot control, planning and prediction (Mitchell, 2006, Luger and Stubblefield, 2008).

ML paradigms are classified into different categories based on their basic assumptions on representation, performance methods, and learning algorithms. ML paradigms are classified into five categories depending on how they learn (Langley, 1998; Roberts, 2003):

1) ***Inductive learning:*** Given a set of examples (inputs and classification pairs), the system tries to approximate the evaluation function in order to make accurate predictions about future examples. This paradigm uses different algorithms such as condition-action rules, decision trees or similar logical knowledge structures. The information on classes or predictions are stored in the action parts of the rules or the leaves of the tree. In the rule-induction framework, learning algorithms usually carry out a greedy search through the space of rule sets or decision trees using statistical evaluation functions to select attributes in order to incorporate them into the knowledge structure.

2) ***Instance-based or case-based learning:*** In this paradigm, knowledge is represented in terms of specific cases or experiences and relies on flexible matching methods to retrieve these cases and apply them to new situations. One common approach is based on some distance metric that finds the stored

case nearest to the current situation. The approach uses this metric as classification or prediction.

3) ***Analytic learning:*** In this paradigm, the knowledge is represented as rules in logical form but typically employs a performance system that uses search to solve multi-step problems. In the common technique, knowledge is represented as inference rules, and later problems are phrased as theorems and proofs are searched. Learning mechanisms in analytical learning framework use background knowledge to construct proofs or explanations of experience. They are then compiled into more complex rules that can solve similar problems.

4) ***Connectionist learning:*** This paradigm is inspired by the model of human brain as an enormous parallel computer. In this approach which is also called neural networks, knowledge is represented as a multilayer network of threshold units that spreads activation from input nodes through internal units to output nodes. Weights on the links determine how much activation is passed on in each case. The activations of output nodes can be translated into numeric predictions or discrete decision about the class of the input.

5) ***Evolutionary learning:*** This paradigm is a learning task which employs in its search engines a technique belonging to the evolutionary computation field. Evolutionary computation techniques are optimization tools based on evolution of biological life in the natural world like Darwinian natural selection or the genetic codification of life forms. An optimal solution is

discovered by successively breeding generations assessed by an objective "fitness" function.

The behaviours of robot are determined by the structure and dynamics of both the robot and its environment, in particular by the interface between them. ML distinguishes between two types of behaviours namely Stimulus-Response (S-R) behaviour and dynamic behaviour. S-R behaviour which does not require the use of internal state that is the detectors are connected in a direct way with the effectors in which a stimulus to the system by its sensors causes a direct response by its effectors. On the other hand, dynamic behaviour contains several kinds of internal state that mediate between input and output that is the internal state that performs multiple layers of computation before a response is generated (Dorigo and Colombetti, 1998).

In several cases, Reinforcement Learning (RL) using high-level task performance feedback is applied to the evolution of robot controllers. Such feedback is required for the evolution of complex behaviours. Evolutionary computation (GA) and RL are powerful approaches that are capable of generating autonomous learning in robot behaviour problems. A well-established algorithm that combines these techniques is the Learning Classifier System (LCS). LCS is a dynamic, rule-based system that utilizes RL to learn by example and induction. LCS is flexible and structurally adaptive for a robot controller. It can effectively interact with the environment and eventually endows the robot with learning ability to carry out its task. It is adopted as the basic architecture of a robot controller because it learns syntactic rules consisting of strings of binary alphabet structure which are easily implemented by a computer. Since the RL and GA are key components of a LCS, a detailed introduction is given in the following sections.

2.4 Reinforcement Learning

Reinforcement Learning (RL) can be defined as the problem of an agent that learns how to interact in a given environment by performing a task through trial and error that optimizes a function of scalar called "reinforcement". An agent is defined as a system situated within and a part of environment that senses the environment and acting upon that environment through effectors. Most of the researchers in RL focus on algorithms that are inspired to solving this type of problem efficiently. In RL, the agent learning activity is called a "*policy*". A policy is a mapping from perceived states of the environment to actions of those states which maximize some function of the reward. The reward function defines what are the good and bad actions for the agent (where rewards are positive reinforcements, while punishments are negative reinforcements) (Sutton and Barto, 1998; Gosavi, 2009).

The study to find the optimal policies that solve the problem of designing a controller to minimize a dynamical system's behaviour over time with respect to some objective function is the objective of an optimal control. Techniques to solve RL problems are part of "adaptive optimal control" (Sutton and Barto, 1998; Gosavi, 2009). Current RL implementations can be divided into three approaches.

The earliest is Dynamic Programming (DP) which was developed by Richard Bellman in the 1950s. DP refers to a collection of algorithms that can be used to compute optimal policies for a given model of the environment which are perfectly known in order to iteratively estimate the value of states. Classical DP algorithms are of limited utility because of their assumption of a perfect model and because of their great computational complexity, but they are still important theoretically (Sutton and Barto 1998; Gosavi, 2009).

The second approach is known as Monte Carlo methods which do not require a perfect model and are conceptually simple, but require only experience (sample sequences of states, actions, and rewards) from on-line or simulated interaction with an environment. Monte Carlo method divides experience into episodic tasks, and all episodes eventually terminate no matter what actions are selected. They require that a sequence of steps between states has a defined termination, at which point the values of the states leading to that final state can be calculated (Sutton and Barto, 1998; Gosavi, 2009).

The third approach is a combination of Monte Carlo methods and DP, and is known as Temporal Difference (TD) learning. TD methods require no model of the environment and are fully incremental, but are more complex to be analyzed. TD methods update value estimate based in part on other learned estimates, without waiting for a final outcome (they bootstrap). TD methods can be divided into two categories: (i) *On-policy* methods that evaluate the values of a policy while using it for control. (ii) *Off-policy* methods that use a different policy to choose their actions from the one which is being evaluated. *Sarsa* is an on-policy TD control method which learns an action-value function rather than a state-value function. As it learns the value of state-action combinations, it can be used to control an agent's movements around an environment. *Q-learning* is an off-policy method. The learned action-value function Q directly approximates the optimal action-value function Q^* independent of the policy (Sutton and Barto, 1998; Gosavi, 2009).

All the previous approaches aim to learn a value function, that is trying to associate a value to each state or to each state-action pair so that it can be used to implement a control policy (Dorigo and Colombetti, 1998).

In LCS, a rule-based system with adaptive capabilities was build to overcome the brittleness shown by traditional handmade expert system. The learned LCS controller consists of a set of rules called classifiers that associate control actions with sensory input in order to perform the desired behaviour. However, Dorigo *et al.* (1994b) have obtained by operating a strong simplification on an LCS, a system equivalent to Q-learning where LCS learns the usefulness ("strength") of classifier by exploiting the "Bucket Brigade Algorithm" (BBA) which is a temporal difference technique strongly related to Q-learning. This makes the connection between LCSs and adaptive optimal control to be much tighter.

On the other hand, LCS learns new classifiers by using genetic algorithm (GA). Therefore a LCS learns a value function (the strength of classifiers), and at the same time it searches the space of possible rules by exploiting one evolutionary algorithm.

2.5 Genetic Algorithm

Genetic algorithm (GA) is a search procedure which is based on the idea of natural evolution in natural selection and natural genetics. GA allows an effective search in very large search space (Michalewicz, 1998; Goldberg, 2007). GAs behave in a similar manner to the biological genetics in their structure and function. In GA, the representation of solutions depends on the problem to be solved. Usually these solutions are represented as strings of bits (i.e. ones and zeros) of fixed length and