

A NEURO-FUZZY APPROACH FOR HIERARCHICAL BEHAVIOUR LEARNING

Dilek ARSLAN
darslan@aselsan.com.tr

Supervisor: Ferda Nur ALPASLAN
alpaslan@ceng.metu.edu.tr

Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
December 2003

Abstract

In autonomous navigation of mobile robots, the dynamic environment is a source of problems. Because it is not possible to model all the possible conditions, the key point in the robot control is to design a system that is adaptable to different conditions. This paper first describes a behaviour hierarchy, which decomposes behaviours of a robot, needed to achieve a goal, into simpler behaviours. Then explains a neuro-fuzzy approach to learn fuzzy rules and to model fuzzy systems. Finally a combination of these two approaches for learning hierarchical fuzzy behaviours is discussed.

TABLE OF CONTENTS

1	Introduction.....	3
2	Background.....	3
2.1	Fuzzy Modelling, Fuzzy Control and Mobile Robots	3
2.2	Fuzzy Behaviour Hierarchies.....	4
2.3	A Neuro-Fuzzy Approach for Learning Fuzzy Rules.....	6
3	Related Work	8
4	Conclusion and Future Work	9
5	References.....	9

1. Introduction

Over the past few years, research in autonomous mobile robots gained an interest due to various tasks they're used such as tasks in industrial environments and on remote planetary surfaces. Methods used to control autonomous mobile robots can be classified into two groups; deliberative and reactive.

Deliberative approach uses planning in a completely known environment. But need for exact model of the world makes usage of deliberative approach very difficult in some real world tasks. Because, in most of the problems, the environment is open to modifications and is not well known.

In reactive approach, model of the world is not needed. Actions are determined according to information gathered from sensors. Drawback of reactive approach is limited and uncertain sensor data. Many modelling methods have been proposed to overcome these uncertainties. Fuzzy modelling is one of these reasoning methods and is used widely.

In recent years, fuzzy modelling, as a complement to the conventional modelling techniques, has become an active research topic and found successful applications in many areas. However, most fuzzy models are presently built based only on operator's experience and knowledge, but when a process is complex there may not be an expert available. In this kind of situation the use of unsupervised learning techniques is of fundamental importance. Recently, several approaches have been proposed for automatically generating fuzzy if-then rules from numerical data without domain experts. Hybrid systems like neuro-fuzzy systems are one way of generating fuzzy if-then rules automatically.

Lin's fuzzy-neural network [5] is one of these hybrid systems. It's a simple but effective fuzzy-rule based model. It uses a four-layer architecture (input, fuzzification, inference, and defuzzification layers) to implement a fuzzy-rule based model of a real system from input-output data.

This report is organized as follows; section 2.1 gives an introduction to fuzzy modelling, fuzzy control, and use of fuzzy control in autonomous mobile robots, section 2.2 explains fuzzy behaviour hierarchies, section 2.3 describes Lin's neuro-fuzzy approach for fuzzy modelling. In section 3, other work about learning hierarchic behaviour is given and finally in section 4. future work is discussed.

2. Background

2.1 Fuzzy Modelling, Fuzzy Control and Mobile Robots

Using fuzzy logic in design of mobile robots gives us a useful tool for dealing problems like uncertain sensor data, and a dynamic environment that cannot be modelled in the beginning of robot navigation. So it has become a popular tool for control applications in recent years.

Fuzzy set theory, developed by Lotfi Zadeh [6], classifies information into sets that do not have crisp boundaries. Fuzzy sets are describes by linguistic terms such as "high",

“low”, “small”, “large”, etc. In fuzzy logic, being a member of a set does not take only the values “true” and “false”. Membership value determines how much a value is member of a set. Membership values can be real numbers in interval $[0,1]$.

Fuzzy modelling is an approach used to form a fuzzy system’s model. It is based on the idea of finding a set of local input-output relations describing a process. A fuzzy model consists of a number of fuzzy if-then rules. Premise and consequent parts of the rules are formed by combination of fuzzy sets defined in the different domains of the variables. Each input-output relation is described by a fuzzy rule. A fuzzy rule can easily capture the knowledge of a human expert. Number of system’s inputs determines total number of rules in the model.

Fuzzy control systems produce actions according to fuzzy rules based on fuzzy logic. The basic units of a fuzzy logic controller are; fuzzifier, fuzzy rule base, fuzzy inference engine, and defuzzifier. In fuzzifier, crisp input values are mapped to fuzzy sets. Fuzzy rule base contains fuzzy if-then rules, which specifies behaviour of the system. Fuzzy inference engine maps input fuzzy sets to output fuzzy sets using the fuzzy rule base. Defuzzifier maps the fuzzy output sets to crisp output value.

In a fuzzy control system of a mobile robot, first crisp sensor values are translated into linguistic classes in the fuzzifier, and then appropriate rules are fired in the fuzzy inference engine that generates a fuzzy output value. Finally this value is translated into a crisp value and is used as speed or moving angle of the mobile robot.

Fuzzy logic controllers are a useful choice when a precise linear model of the system cannot be easily obtained. They also show robustness to variability and uncertainty in inputs. These characteristics fit the needs of autonomous robot navigation.

2.2 Fuzzy Behaviour Hierarchies

When a system requires many rules and complex decision-making, hierarchical rule structures are a viable alternative for dealing with autonomous mobile robots. Behaviours can be arranged as a hierarchical network of fuzzy rule bases, each responsible for some part of system functionality. Each of these rule bases corresponds to a behaviour of the mobile robot.

In behaviour hierarchy, behaviours are grouped into two categories; primitive behaviours and composite behaviours. Primitive behaviours are elementary tasks, which resides at the lowest level of behaviour hierarchy. These are encoded as fuzzy rule bases with distinct control policies governed by fuzzy inference. They are typically simple and self-contained behaviours that serve a single purpose while operating in a reactive fashion. Examples of primitive behaviours are; obstacle avoidance, motion towards a given location, wall following, etc.

When operating alone, primitive behaviours would be insufficient for performing complex navigation tasks. They are building blocks for higher-level coordination behaviours, called composite behaviours. Examples of composite behaviours are goal seeking and route following. Capabilities of composite behaviours can be combined to produce composite behaviour(s) suitable for goal-directed navigation.

Figure 1 is a conceptual illustration of a general behaviour hierarchy consisting of a primitive level of individual motion behaviours, β_i , coordinated by higher-level composite behaviours, B_j . The interconnecting circles between composite behaviours and the primitive level represent weights and activation thresholds associated with primitive behaviours by composite behaviours. Each primitive behaviour maps inputs to a vector of fuzzy control outputs. Higher-level behaviours act as fuzzy decision systems that map goal information and sensor data to dynamically adaptive, scalar weights associated with each primitive behaviour.

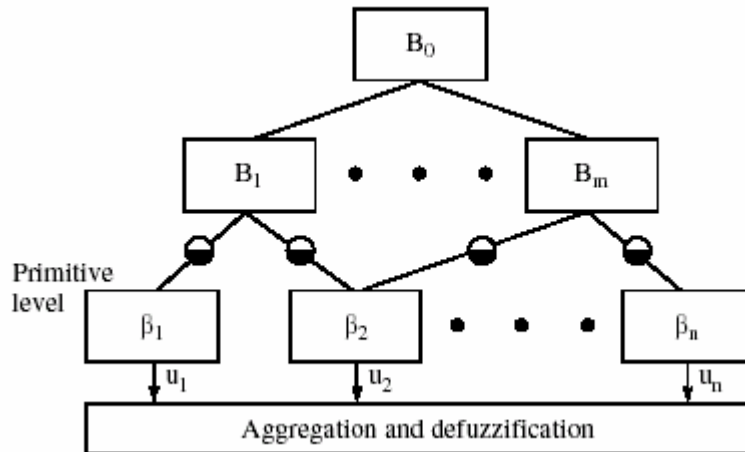


Figure 1-Generic Fuzzy Behaviour Hierarchy (Adapted from [3])

An example behaviour hierarchy for indoor navigation of a mobile robot might be organized as in Figure 2. It implies that goal-directed navigation can be decomposed as a behavioural function of **goal-seek** (collision-free navigation to some location) and **route-follow** (assuming some direction is given in the form of waypoints or a path plan). These behaviours can be further decomposed into the primitive behaviours shown, with dependencies indicated by the adjoining lines. **Avoid-collision** and **wall-follow** are self-explanatory. The **doorway** behaviour guides a robot through narrow passageways in walls; **go-to-xy** directs motion along a straight-line trajectory to a particular location. The circles represent weights and activation thresholds of associated primitive behaviours. Fluctuations in these weights are at the root of the intelligent coordination of primitive behaviours, which leads to adaptive system behaviour.

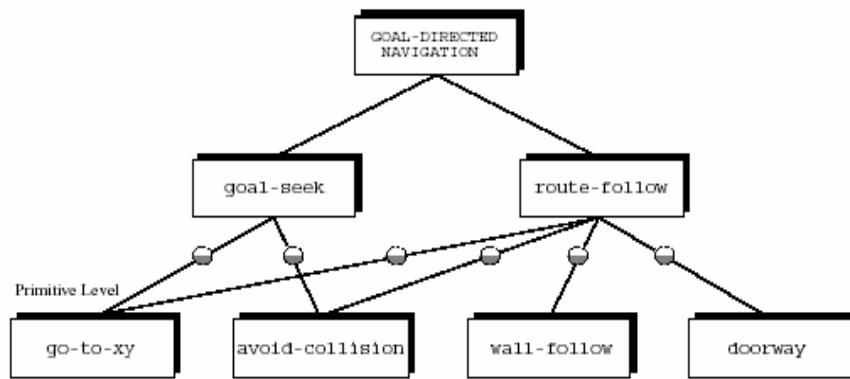


Figure 2- Hierarchical decomposition of indoor navigation of a mobile robot (Adapted from [3])

When more than one primitive behaviours are active, interactions can take the form of behavioural cooperation or competition. Coordination of multiple behaviours is achieved by weighted control decision-making. This control is called degree of applicability (DOA)—a measure of the instantaneous level of activation of a motion behaviour. Fuzzy rules of composite behaviours are formulated to include weighting consequents that modulate the DOAs of behaviours at a lower level. These are called applicability rules. DOA is defined over the closed unit interval $[0, 1]$, and defaults to zero if unspecified by a composite behaviour. In general, a composite behaviour, c , will include applicability rules for each primitive behaviour p modulated by c . Thus for all p , DOA of behaviour p , $\alpha_p \in [0, 1]$ is determined by fuzzy inference as the output of an associated composite behaviour. This feature allows certain robot behaviours to influence the overall behaviour to a greater or lesser degree as required by the current situation and goal. It serves as a form of adaptation since it causes the control policy to dynamically change in response to goal information and sensory input. The behaviour hierarchy, then, is a dynamic non-linear mapping from situations to actions rather than a static non-linear mapping represented by a fixed set of fuzzy rules.

After determining DOA for each primitive behaviour, outputs of primitive behaviours are multiplied by their DOA values and the results are combined by using vector summation. That is; result of the composite behaviour and results of each primitive behaviours are defined as a vector of velocity and angle of the robot. Outputs of the primitive behaviours, produced by different primitive behaviours, are combined by vector summation.

2.3 A Neuro-Fuzzy Approach for Learning Fuzzy Rules

Several researches have been made on using neural networks for implementing fuzzy models. Lin [5] proposed one of these studies for fuzzy-neural system modelling. In the fuzzy-neural network he proposed, there are four layers for input, fuzzification, inference, and defuzzification. Figure 3 shows architecture of the network.

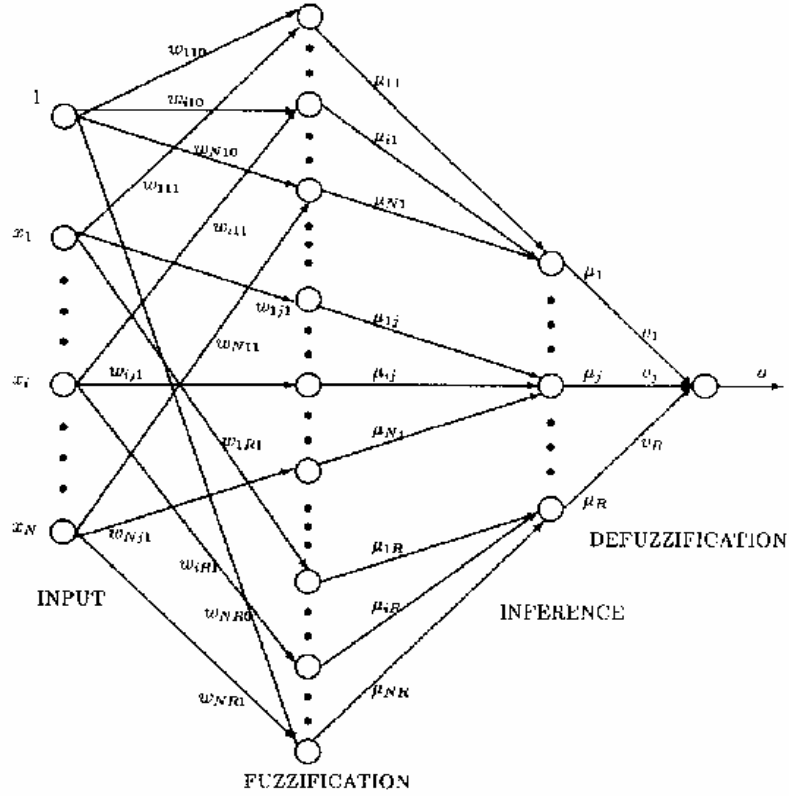


Figure 3- The Architecture Of The Fuzzy-Neural Network (Adapted from [5])

There are N inputs, with N neurons in the input layer, and R rules, with R neurons in the inference layer. There are $N \times R$ neurons in the fuzzification layer. The first N neurons (one per input variable) in the fuzzification layer incorporate the first rule, the second N neurons incorporate the second rule, and so on. Every neuron in the fuzzification layer represents a fuzzy membership function for one of the input variables.

The set of weights between the input and the fuzzification layer are labelled by $W = \{\{w_{ij0}, w_{ij1}\} : i=1, \dots, N; j=1, \dots, R\}$. The connecting weights between the third layer and the fourth layer are the central values, v_j , of the fuzzy membership functions of the output variable. These set of weights are labelled by $V = \{v_j : j=1, \dots, R\}$. The neural network weights in V and W determine the fuzzy rules. The network can represent any continuous function $f: R^N \rightarrow R$.

A concept called “fuzzy curves” is used for;

- Identification of the significant input variables,
- Estimation of the number of rules needed in the fuzzy model,
- Determination of the initial weights for the neural network.

To produce fuzzy curves in a multiple-input, single-output system, first, for each input variable x_i , data points are plotted in x_i - y space (y is the system output). Then a fuzzy membership function for input x_i is plotted. This membership function is defined by

$$\phi_{ik}(x_i) = \exp\left(-\left(\frac{x_{ik} - x_i}{b}\right)^2\right), \quad k = 1, 2, 3, \dots, m$$

where m is number of training points x_i , and b is about %20 of the length of the input interval of x_i .

Each pair of ϕ_{ik} and the corresponding y_k provide a fuzzy rule for y with respect to x_i . Then defuzzification is used on these rules to produce a fuzzy curve c_i for each input variable x_i . If the fuzzy curve for given input is flat, then this input has little influence in the output data and is not a significant input. Initial weights in V are set to the centres of output variable fuzzy membership functions. For this, range of the desired output data is divided into R intervals, and initial weights in set V are set to be the central value of these R intervals.

3. Related Work

Work by Brooks [1], introduced layered control approach to the robotics community. Rather than decomposing the autonomous control problem into separate layers in a deliberative scheme, Brooks proposed an architecture where each layer has direct effect on the action. Behaviours are represented as layers in subsumption architecture and multiple behaviours are coordinated in parallel.

Gachet [2] and Tunstel [3] proposed two works based on Brooks' approach. Gachet presented a learning method, which is hybrid of reinforcement learning and neural networks. His work is an implementation of reinforcement learning algorithm through the use of a special neural network topology called AHC (Adaptive Heuristic Critic). The AHC is used as a fusion supervisor of primitive behaviours in order to execute more complex robot behaviours like path following or going to goal.

Outputs of primitive behaviours are appropriate values for linear velocity and curvature. Primitive behaviours are not learned. The primitive behaviours and the information they take into consideration is fixed during the execution.

The hybrid learning is used to find a coefficient for each primitive behaviour and output of the primitive behaviours are multiplied by these coefficients. Vector summation is used for behaviour modulation.

In his work, Tunstel presented an approach to hierarchical control design for the case where the collection of subsystems is comprised of fuzzy logic controllers and fuzzy knowledge-based decision systems. He used fuzzy rule bases for both primitive behaviours and composite behaviours. These rule bases are distinct and distributed. Rule bases of composite behaviours have rules, which determine coefficients of the primitive behaviours. Outputs of each primitive behaviour are multiplied by its coefficient and then modulated by using weight counting. Tunstel gives an example application in which this fuzzy model for autonomous mobile robots is extended for multi-robot control. No learning was used in this work. Rules in the rule bases of primitive and composite behaviours are found by trial and error.

In another work, Tunstel [4] used evolutionary approaches for learning rules of composite behaviours. Since genetic algorithms are slow to converge, the learning

was done offline. Again he used trial and error to form rule bases of primitive behaviours.

4. Conclusion and Future Work

So far, I've presented an introduction to behaviour hierarchies, a fuzzy approach to behaviour hierarchies, and Lin's neuro-fuzzy approach for fuzzy modelling. In my thesis, I want to combine these two methods and develop a system, which uses Lin's neuro-fuzzy learning approach for learning rules in both primitive and composite behaviours' knowledge bases. Since training of Lin's neuro-fuzzy network is quite fast, this learning can be done online. Since output of primitive behaviours represent velocity vector of the robot, modulation of these behaviours, after multiplying their result by coefficients form composite behaviour, can be done by using vector summation.

I will use a simulation to test performance of the system. Software agents will represent robots. I will use multi-agent concept for just sharing information in the first phase. Other agents will serve to an agent as remote sensors only. Then as a second phase, I will try to extend fuzzy behaviour hierarchy for multi-agent control.

As an infrastructure for the simulation, I'm planning to use HLA (High Level Architecture). HLA is a framework for creating computer simulations out of component simulations. It provides interoperability among component simulations, called federates. Each agent will be defined as a federate of HLA and will use HLA's communication facilities for information sharing among the agents.

5. References

- 1) Brooks RA. A robust layered control system for a mobile robot. IEEE J Robot Automat 1986;RA-2(1):14–23.
- 2) Gachet D, Salichs MA, Moreno L, Pimentel JR. Learning emergent tasks for an autonomous mobile robot. In: IEEE Int Conf on Intelligent Robotic Systems, 1994. pp 290–297.
- 3) Tunstel E, Oliveira M, Berman S Fuzzy Behaviour Hierarchies for Multi-Robot Control, 2002, International Journal Of Intelligent Systems, Vol.17, 449-470.
- 4) Tunstel E, Lippincot T, Jamshidi M, Behaviour Hierarchy for Autonomous Mobile Robots: Fuzzy-Behaviour Modulation and Evolution, 1997, International Journal Of Intelligent Systems, Vol.3, 37-50.
- 5) Lin Y, Cunningham G, A New Approach to Fuzzy-Neural System Modelling, 1995, IEEE Transactions On Fuzzy Systems, Vol. 3, No. 2.
- 6) Zadeh L.A, Fuzzy Sets, Information and Control, No:8, pp.338-353, June 1965.