Chapter 3.26
Computational Intelligence for Modelling and Control of Multi-Robot Systems

M. Mohammadian
University of Canberra, Australia

ABSTRACT

With increased application of fuzzy logic in complex control systems, there is a need for a structured methodological approach in the development of fuzzy logic systems. Current fuzzy logic systems are developed based on individualistic bases and cannot face the challenge of interacting with other (fuzzy) systems in a dynamic environment. In this chapter a method for development of fuzzy systems that can interact with other (fuzzy) systems is proposed. Specifically a method for designing hierarchical self-learning fuzzy logic control systems based on the integration of genetic algorithms and fuzzy logic to provide an integrated knowledge base for intelligent control of mobile robots for collision-avoidance in a common workspace. The robots are considered as point masses moving in a common work space. Genetic algorithms are employed as an adaptive method for learning the fuzzy rules of the control systems as well as learning, the mapping and interaction between fuzzy knowledge bases of different fuzzy logic systems.

INTRODUCTION

Fuzzy logic systems are increasingly used in control applications. Their ability to cope with uncertainty inherent in complex systems makes them an attractive method for solving complex, uncertain and dynamic systems. Current fuzzy logic systems are developed based on the individualistic systems. These systems are unable to face the challenge of interaction that might be necessary between different fuzzy logic systems solving a complex problem. There is a need for a structured approach to design fuzzy logic systems for controlling complex systems consisting of multiple fuzzy logic systems and fuzzy knowledge bases as is the case in a hierarchical fuzzy logic system.

In general hierarchical fuzzy logic systems consist of several fuzzy logic systems, each performing a specific task which are combined to form a system to solve a complex task. These controllers interact with each other to solve the problem at hand. The output of each fuzzy logic controller in a hierarchical fuzzy logic system has an effect on other fuzzy logic systems and consequently to the final output of the system.

The division of individual fuzzy systems required to solve complex problems demands that those problems be decomposed and distributed among different fuzzy logic systems. One of the main limitations on the application of hierarchical fuzzy logic systems in complex problem-solving domains is the lack of methods to structure the development of the hierarchical fuzzy logic systems. This is an important issue when considering the robustness and efficiency of the system. A well structured hierarchical fuzzy logic system can perform its tasks more efficiently.

The design of the fuzzy knowledge base of complex fuzzy logic systems are based upon human experience and the operator's knowledge of the system to be controlled (Lee, 1990). The fuzzy rules are formulated by a trial and error method which is not only time consuming but also does not guarantee 'optimal' fuzzy rules for the system. Incorporating genetic algorithms into the design of a fuzzy logic system ensures automatic generation of fuzzy rules for a fuzzy logic system.

This chapter is organised as follows: in the next section the learning of fuzzy rules of fuzzy logic systems using genetic algorithms are described. The application of this learning method to control a simulated multi-robot system using hierarchical fuzzy logic systems is considered and simulation results are presented. Conclusions are then drawn and further research directions are given.

LEARNING OF FUZZY LOGIC SYSTEM USING GENETIC ALGORITHMS

Earlier in the following papers (Mohammadian, 1994; Mohammadian, 1996; Stonier, 1998) an integrated architecture consisting of genetic algorithms and fuzzy logic for automatic rule generation of fuzzy logic systems was proposed. A block diagram of the fuzzy rule generation architecture is shown in Figure 1.

Let us consider a fuzzy logic controller with two inputs (x and y) and a single output (z). As a first step to generating the fuzzy rules, the domain intervals of the input and output variables are divided into different regions, called fuzzy sets. The number of fuzzy sets is application dependent. Assume that x, y and z are all divided into

Figure 1: Fuzzy-GA rule generator architecture