

A FUZZY-BASED MANOEUVRE MANAGEMENT SYSTEM FOR AN AUTONOMOUS UNDERWATER VEHICLE

Divas Karimanzira, Peter Otto, Juergen Wernstedt *

** Technical University of Ilmenau
Institute of Computer- and System engineering
Department of System analysis
Postfach 100565
Ilmenau, Germany
E-mail: divas.karimanzira@tu-ilmenau.de*

Abstract: The problem domain in this work is a three-dimensional simulation of an underwater vehicle (AUV) that must navigate through obstacles towards a stationary goal point. The AUV has a limited set of sensors, including sonar, and can set its speed and direction each decision cycle. We wish to learn a strategy that is expressed as a set of reactive rules, (i.e. stimulus-response rules) that map sensor readings to actions to be performed at each decision time step. Note that the system does not learn a specific path, but a set of rules that reactively decide a move at each time step allowing the vehicle to reach its goal and avoid the obstacles. *Copyright 2005 IFAC*

Keywords: Machine learning, Adaptive control, Navigation, Fuzzy Logic, Rule generation

1. INTRODUCTION

Manoeuvring through obstacles such as mines is one of many important capabilities or behaviours of autonomous underwater vehicles. One way to produce robust behaviour is to perform projective planning. However, real-time performances is often a critical requirement for many of the capabilities needed in autonomous vehicles. Reactive systems, where stimulus-response rules drive the behaviour of the vehicle, can achieve real-time performance and can perform well in a wide variety of situations. An interesting and challenging problem is how to develop or obtain the rules for these reactive systems. In previous work, FuzzyMod[®], a learning system based on ID3 algorithm, was used to learn classification problems (Otto and Wernstedt, 2002). In this work, FuzzyMod[®] is used to learn high-performance reactive strategies

for navigation and collision avoidance. This task domain requires an AUV to navigate through a randomly generated, dense obstacle field and then rendezvous with a stationary object. The AUV has a limited set of sensors, including sonar, and must set its speed and direction each decision cycle. The strategy, or plan, that is learned is exposed as a set of reactive rules (i.e. stimulus-response rules) that map sensor readings to actions to be performed at each decision time step. Simulation results demonstrate that the AUV can navigate through a randomly generated, dense obstacle field and reach a given goal.

2. APPROACH TO LEARNING STRATEGIES

In response to the knowledge acquisition bottleneck associated with the design of expert systems,

research in machine learning attempts to automate the knowledge acquisition process and to broaden the base of accessible sources of knowledge. The choice of an appropriate learning technique depends on the nature of the performance task and the form of available knowledge. If the performance task is classification, and a large number of training examples are available, then inductive learning techniques can be used to learn classification rules. If there exists an extensive domain theory and a source of expert behaviour, then explanation-based methods may be applied (Ramsey, 1998). Many interesting practical problems that may be amenable to automated learning do not fit either of these models. One such class of problems is the class of sequential decision tasks, there exists neither a database of examples nor a complete and tractable domain theory that might support traditional machine learning methods. In these cases, one method for automatically developing a set of decision rules is to manually collect data using a simulation model of the task environment, and then to use inductive learning techniques to learn the classification rules.

The approach described here reflects a particular methodology for learning via a simulation model. The motivation behind the methodology is that making mistakes on real systems may be costly, dangerous or in this case the real system was not yet existent. Since learning may require experimenting with tactical plans that might occasionally produce unacceptable results if applied to the real world, we assume that hypothetical plans will be evaluated in a simulation model (see Figure 1).

Previous studies have illustrated that knowledge learned under simulation is robust and might be applicable to the real world environment if the simulation is more general (i.e. has more noise, more varied conditions, etc.) than the real world environment.

3. SYSTEM DESCRIPTION

The main objective of the experimental system for the expert to drive through obstacles in a virtual reality environment, while collecting data. This data is then used for teaching the controller to learn the behaviour of the expert. After learning, the mimic controller will then be able to drive the AUV independently in an unknown environment.

The experimental setup for data collection and testing purposes is illustrated in Figure 2. The model can be driven in two modi, one for data collection (switch F1 is connected to C) and the second mode is for testing the designed controller (switch F1 is connected to T). As can be seen the simulation setup is composed of four components, namely: (1) *Data preparation module*

(DPM), (2) *Mimic controller or Fuzzy Inference System (FIS)*, (3) *Expert*, (4) *The AUV and its Environment*. The above mentioned components, including the data collection and the learning processes will be explained in detail in the next section.

4. DATA PREPARATION MODULE

This module is required to convert absolute auv-state-variables (i.e. *absolute position, orientation, sonar parameter*, etc) to relative variables (i.e. *distance to goal, goal bearing, distance to obstacle, obstacle bearing*, etc) (see Figure 5a). This drastically reduces the amount of experiments to be performed for data collection. We assume that the AUV knows its own position with some margin of error, and that the position of the stationary target is known. The AUV also has some internal, or virtual, sensors that give the AUV certain information about its own state. The sensors required to optimally control the AUV are: *range, speed, goal bearing, last turn* and *active sonar cell*.

The DPM process the data in two steps. First of all the distances and the directions of the surface points (a_{new}) are calculated from the inputs, which include the obstacle position (x_h, y_h, z_h), the AUV position (x, y, z), a division angle (ang_x), obstacle radius (r_h) and obstacle height (h_h) as illustrated in Figure 3. This is done in the following procedure: Points on the surface of the obstacle are selected systematically using ang_x and then for each selected point on the obstacle the distance and its direction relative to the AUV (a_z, a_y) are calculated. In the next step the sonar view area (see Figure 4) is divided in cells (1-24) and for each selected point on the surface of the obstacle the distance and its direction relative to the AUV (a_z, a_y) are calculated and if this direction lies in the sonar cell(i) then the minimum distance of the selected points is assigned to this cell. If no obstacle lies in the direction of the sonar cell then this cell is assigned the maximum value of 220.

Figure 4 (b and c) shows the results of the system for two examples, when the obstacle is positioned right in front and on the left of the AUV, respectively. The diagrams show the minimum distances of the obstacle in the respective cells.

5. MIMIC CONTROLLER AND DATA COLLECTION

To imitate the behaviour of an expert a mimic controller is required. The mimic controller used here is based on a fuzzy inference system. That means it includes functions for fuzzification, inference and defuzzification. The controller is composed of the inputs and outputs as illustrated in

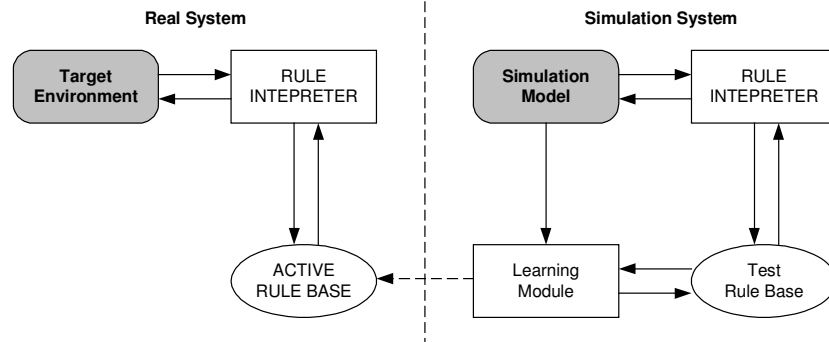


Fig. 1. Learning with a simulation model

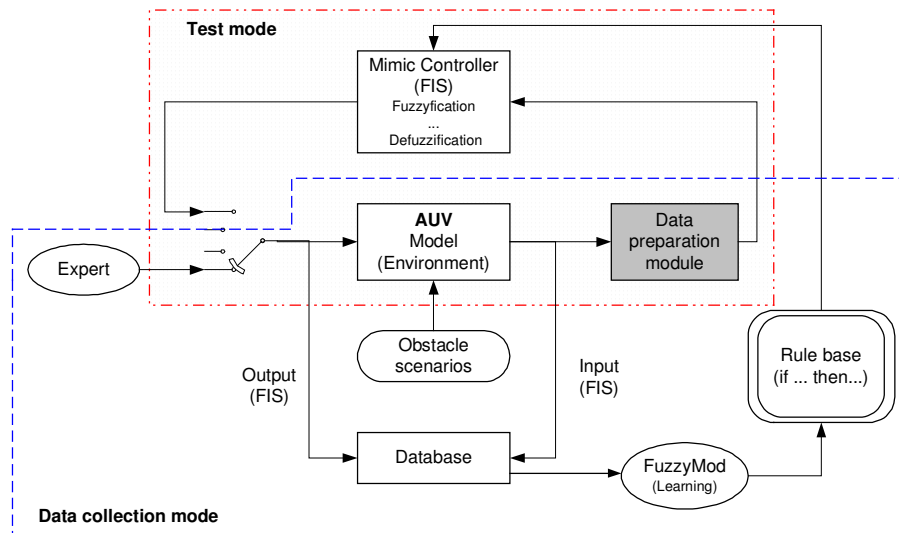


Fig. 2. Experimental setup for data collection and testing purposes

Figure 5b. As the controller is based on a fuzzy system it requires a rule base. The rule base is generated automatically from data samples using FuzzyMod[®] as will be shown in the next section.

The training/validation data will be collected from a series of experiments. Each experiment consists of reading a number of samples from sensors on the AUV while it is placed in the simulation environment. The AUV will be moving during each experiment. The data read from the sensors during an experiment will be stored in files. After the experiment is completed, software on the computer will be used to convert the data into a form that can be used for training/validating the mimic controller. The objective of each experiment is to collect data from sensors that represent a specific state of the auv's world. The primary parameters,

whether to turn to the left, right, down or upwards and at what speed when an obstacle is detected, is what is wished the mimic controller to learn. Here are the parameters to be varied in each experiment, and a brief explanation for each parameter on how it will be implemented in the experiment: *relative obstacle position, obstacle distance, goal position.*

In all experiments the AUV will be controlled by an expert using a joystick. It is recommended that the expert have only the sight of the sonar view area, i.e. the expert can only see what the AUV can see and no extra information, so that the sensors used by the mimic controller will be the same as those used by the expert. When data is collected it can then be used for teaching the mimic controller offline using FuzzyMod[®].

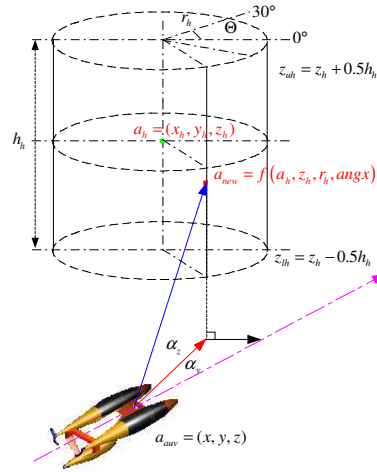


Fig. 3. Distances and directions of selected surface points on the obstacle

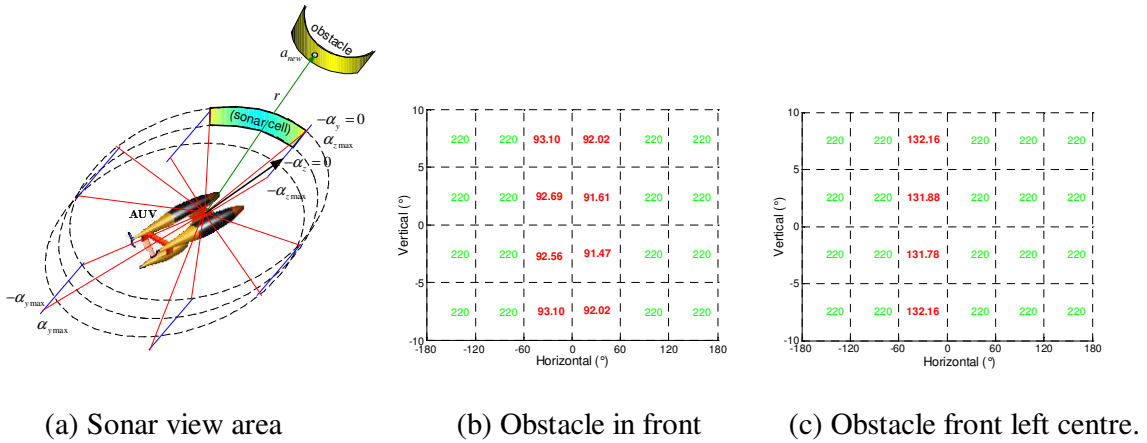


Fig. 4. Division of the sonar view area in cells

6. RULE GENERATION USING FUZZYMOD[®]

FuzzyMod[®] is a tool developed at the TU-Ilmenau for generating fuzzy rules from training data samples (Dung and Otto, 1997). Herewith, the control actions made for different manoeuvre situations by an expert in the virtual world were generalized in the form of IF - THEN rules, so that they can be used in the future, in order to make an autonomous guidance of the vehicle in the real world in similar situations.

The decision behavior of an expert and the static and/or dynamic behavior of systems can be described by the following equations:

$$y = f(u_1, u_2, u_3, \dots, u_n) \quad (1)$$

and

$$y(k) = f(y(k-1), \dots, y(k-m), u(k-1), \dots, u(k-n)) \quad (2)$$

where y is the output and $u_1, u_2, u_3, \dots, u_n$ are the inputs of a static MISO-system, $y(k)$ is the output and $y(k-1), \dots, y(k-m), u(k-1), \dots, u(k-n)$ are the sampled output and input values of a dynamic SISO-System and f is a nonlinear function. The functional dependencies in equation (1) and (2) can be expressed in form of rules, if the input and output values of the system are described by linguistic attributes as a function of the measured values. In principle, there are two ways to determine the attributes for the inputs. In the first ap-

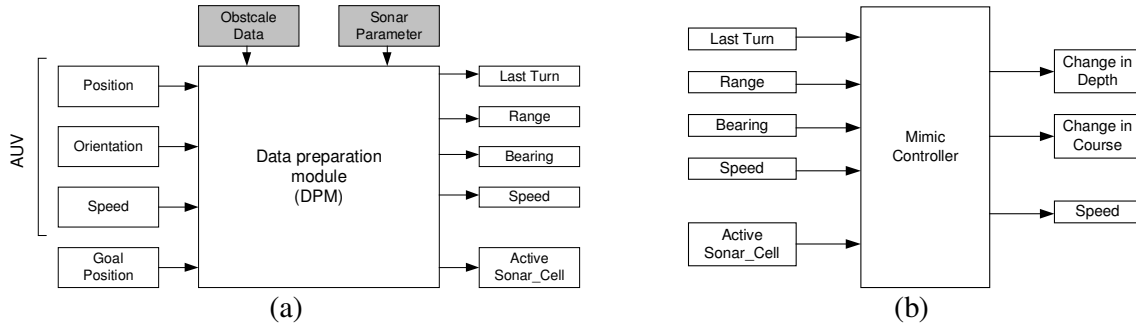


Fig. 5. (a) Data preparation module, (b) Mimic controller

proach, the attributes are determined by dividing the entire range of the process inputs and outputs into a given number of n equal intervals. In the second approach only the output is divided into equal intervals, while a machine learning method optimally specifies the attribute borders for the inputs. For the fuzzification of the process variables, triangular membership functions are assigned to the intervals in such a way that they intersect at the interval borders by $m = 0.5$. The edge intervals get one-side open membership functions. Thus, a measured value with different membership values belong to two intervals. The measured values are then transformed according to the fixed intervals into linguistic attributes, so that a description of the static and/or dynamic expert and/or process behavior in form of linguistic expressions results. The ID3-algorithm (Quilan, 1992) generates an optimal decision tree from the linguistic examples. Production rules of the form:

$$IF(u1 = vs \& u3 = b) THEN y = m$$

$$IF(y(k-1) = vb \& u(k-2) = s) THEN y(k) = vvb,$$

which describe the static or dynamic system/expert behavior, can then be derived from the decision tree. The decision tree is built in such a way that the information content of the attributes for the decision-making process drops with increasing depth and the irrelevant attributes remain unconsidered. After starting the procedure with a primary configuration, the generated fuzzy sets are linked in the rules, and the latter are then checked for their "correctness" (mean information content). After this the fuzzy sets are optimized using Genetic algorithms so that the mean squared error between model and system output is minimized.

7. FUZZY INFERENCE SYSTEM FOR THE MIMIC CONTROLLER

To simplify learning, data was generated for 3 different modules, module responsible for navigating the AUV to goal, module for avoiding obstacles and the other module for calculating the required speed. For the module to go to goal only two input variables are required, goal bearing and distance to goal. For the module for obstacle avoidance only the sonar cell distance values are required as inputs and for the module responsible for speed only the distance to obstacle, distance to goal and the previous speed are required as inputs. In every step if an obstacle is detected the module for obstacle avoidance is activated and if there is no obstacle registered by the sonar cells the go-to-goal module is activated and in either case the speed is calculated by the speed module accordingly. After learning and validation using FuzzyMod[®] the mimic controller was then used to replace the joystic (expert) and the results illustrated in the next section were obtained for several unknown environmental situations.

8. RESULTS

The simulation with the AUV-Model for Matlab[®], developed at the TU Ilmenau, shows the results illustrated in Figure 6 and 7 for selected environment scenarios. The tracking trajectories obtained with the control system show reactive obstacle avoidance. The obstacles were positioned, so that they represent a number of situations, which can occur in the real world environment.

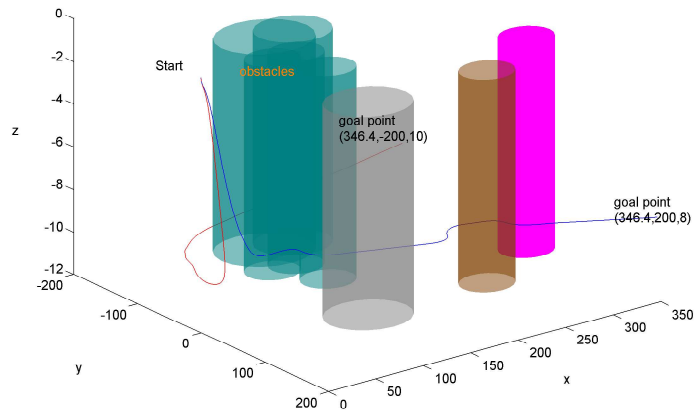


Fig. 6. Simulation results, example 1 & 2

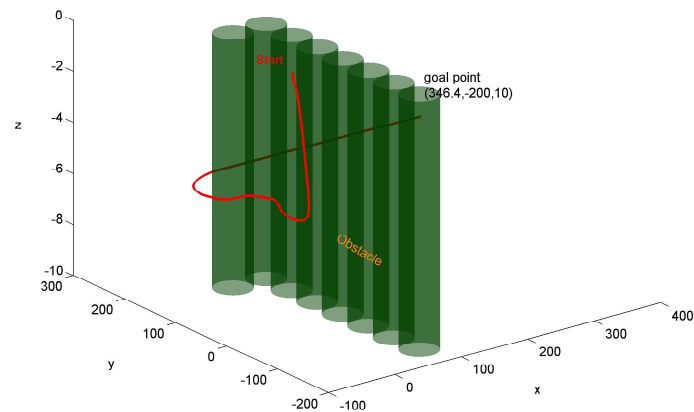


Fig. 7. Simulation results, example 2

9. CONCLUSION

A manoeuvre management system based on fuzzy-rules was presented. The management system learns from an operator by induction using ID3-Algorithm. This can be done offline or online. Simulation results show the performance of the manoeuvre management system in an unknown environment. Now the system has been tested successfully in the real AUV.

REFERENCES

- Dung, L. T. and P. Otto (1997). Fuzzy-opt - ein werkzeug zum entwurf optimaler fuzzy-systeme. *at - Automatisierungstechnik* **45**, 555–556.
- Otto, P. and J. Wernstedt (2002). Fuzzy methods for the optimal emulation of expert decision behaviour and for the optimal controller design. *East-West Fuzzy Colloquium 2002, 10th Zittau Fuzzy Colloquium 2002, September 4-6* **10**, 351–367.
- Quilan, J. R. (1992). *Programs for Machine Learning*. Morgan Kaufmann Publishers. San Mateo, California.
- Ramsey, C. L. (1998). Simulation-assisted learning competition. *Proceedings on Machine learning* **7**, 211–215.