EUFIT 1996, Aachen, pp. 1095 - 1099, 2-5 Sep. 1996

Automatic Behaviour Adaption for Mobile Robots with different Kinematics

Jörg Huser, Hartmut Surmann and Liliane Peters

GMD - German National Research Center for Information Technology
53754 Sankt Augustin, Germany, Email: {huser,surmann,peters}@gmd.de

Abstract: Industrial applications of mobile robots require the same behaviour for different types of autonomous robots. If this at a first glance seems easy, it is in fact a rather complicated task as it means to adapt a system with a different mechanical model and thus different behaviours to a prescribed reaction to the environment. If in addition this goal should be reached automatically the complexity of the problem increases. In this paper we present an adaptive learning method - mother-child teach-in - through which from a given well tuned prototype a new rule base for a mobile platform with different kinematics is generated automatically. At the end of the teaching phase, the new platform shows the same behaviour as the prototype.

Keywords: Fuzzy rule-based systems, autonomous mobile robots, kinematic adaption, redesign, robot mechanics, robot navigation

1 Introduction

It has been shown that neural networks or genetic algorithms can automatically design or adapt simple fuzzy rule-based systems (FRBS), e.g. [1, 2]. For dynamical processes of higher order the rule base is extended to a FRBS with internal variables [3]. The optimization of higher order systems is very time consuming and sometimes senseless. What type of adaption is needed, if the input universe of discourse is nearly the same but the system and the output universe of discourse change? In our case this would correspond to two different robot types, with different kinematic models, that have to fulfill the same task, e.g. parcel carriage, in the same environment. We consider as input to our system the sensors and as output the expected reaction or behaviour in the same environment. Different kinematic models need different control strategies. The adaption in this case refers to the automatic strategy for converting the rule-base of the first trained robot into a rule-base for the second. In this paper we will discuss an approach for the automatic "teach-in" of a second robot (child) based on the knowledge contained in a given one (mother). First we will shortly present the employed strategy for the first robot. Afterwards we introduce the teach-in method through imitation, taking into consideration the problems which came up due to different kinematic models of the two robots. Thirdly we present the first results of our approach, tested on our simulation environment. Some final remarks are concluding the paper.

2 Autonomous Mobile Robot

The main goal of our current research project is the development of a collision-free autonomous vehicle for unknown indoor environments [3]. Our test platform[1] is an autonomous vehicle 157cm x 74cm x 75cm size, weighting 400kg with a payload of 150 kg. It is equipped with ultrasonic sensors which are the main source for the environmental information. The measured sensor values are converted independantly of their accuracy into linguistic information like "wall near", "wall far away", "junction detected at left". This information is passed to the navigator to control the vehicle trajectory. At the same time the linguistic information is given to a planner module. The planner operates in two modes: exploration and goal-orientation. In the exploration mode a map of the environment is built.

[1]Our thanks to the TZN GmbH Unterlüß for the appropriation of the vehicle.

Figure 1: The autonomous vehicle MORIA.

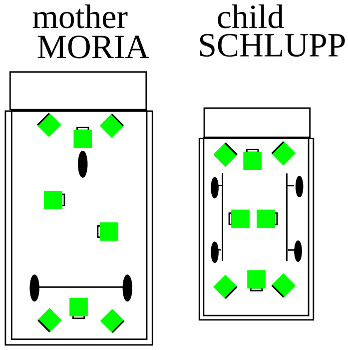

Figure 2: MORIA and SCHLUPP.

During the goal-orientation mode this map is updated. Through the linguistic structured description of the map the needed memory space of the complete environment is relatively small compared to a geometric description of the same environment [4]. A cockpit-like simulation, developed in-house, monitors the system. It enables the user to switch between manual/interactive command mode and autonomous mode (coordinated by the planner).

The simulation models implemented in our environment take into consideration sensory noise as well as the specific geometric and kinematic features of each platform. The "mother" robot is MORIA, an autonomous system with one motor for direction - backward, forward as well as the steering-angle, and the "child" is SCHLUPP a robot with two motors on the left and right sight. The second system is driving forward or backward when the motors get the same speed. Through a difference between the left and right velocity a turning to the left or right is reached. Both robots have different kinematics, sensor locations and geometries (Fig. 2).

3 Mother-child Teach-in approach

As mentioned earlier various autonomous systems can be used for the same or different services within an environment, e.g. [5]. For all these systems the used strategy, in our case the reactive navigator and the global planner, remain the same. Until now human intelligence was needed to adapt a given knowledge base to a new type of an autonomous vehicle with e.g. different kinematics. We developed an automatic tuning method for adapting a well-tuned fuzzy rule-base of a given kinematic model to a new one. We presume that the input to the system (sensors) remains nearly same for both robots and the expected output (reaction) of the system has to be adapted from the mother.

In our first approach SCHLUPP starts with the rule-base of MORIA, where the considered output variables change from *speed* and *angle* to *left speed* and *right speed*. Both variables started with equally distributed membership functions (Fig.3). The considered output variables changed from $SPEED$ and $ANGLE$ to *left speed* and *right speed*. An example of a typical rule looks like the following:
IF COMMAND is C_i AND STATE(1) is $ST1_{l1}$ ... AND STATE(n) is STn_{ln} AND INPUT(1) is I_{h1} ... AND INPUT(m) is I_{hm} THEN LEFT-SPEED is LS_j RIGHT-SPEED is RS_j

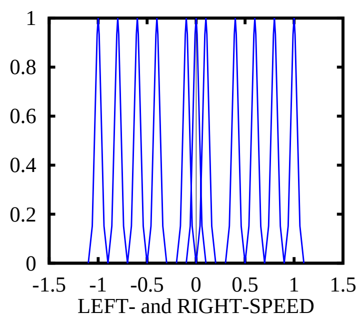

Figure 3: Equally distributed membership functions.

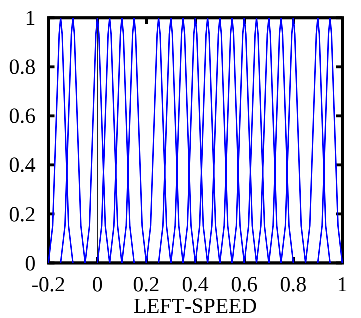

Figure 4: Adapted membership functions for *left speed*.

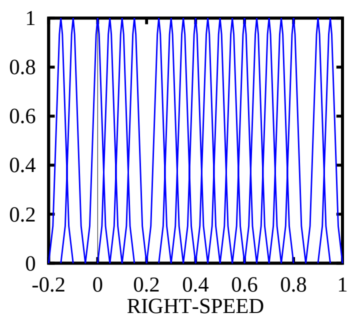

Figure 5: Adapted membership functions for *right speed*.

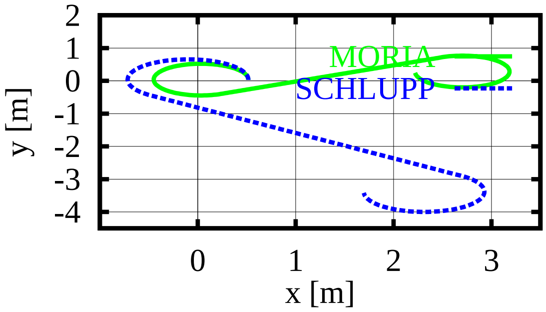

Figure 6: Approximated behaviour of SCHLUPP derivated from MORIA with distributed membership functions.

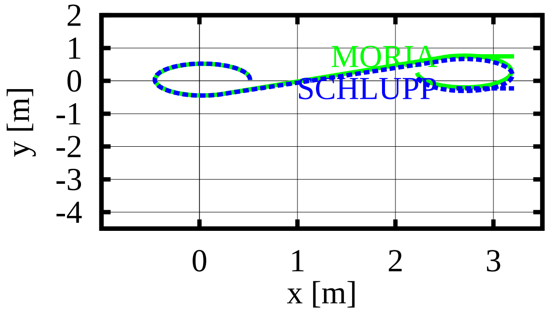

Figure 7: Approximated behaviour of SCHLUPP derivated of MORIA with adapted membership functions.

The adaption of the membership functions is performed through an iterative approach. For each fuzzy rule, the kinematics are compared and the closest predefined membership function of SCHLUPP is selected to reach the same behaviour of MORIA. So, the first rule base is initialized. Figure 6 shows the differences between the track of both robots. Depending on accuracy and desired simulation time the behaviours of the two robots are more or less different. Of course, predefined membership functions do not gain as good results as individually specified membership functions.

Therefore, in our second approach the position and width of membership function is generated depending on the remaining response in each fuzzy rule which includes motor values. The membership functions are sorted to avoid multiple membership functions. The knowledge and the structure of the rules are transmitted from the mother. Interesting is that the membership functions for *speed left* and *speed right* are also nearly normally distributed (Fig.4 and 5) which results from the symmetric rule set. Figure 7 shows the behaviour of both robots for the same situation as in figure 6. The behaviour of SCHLUPP is much more similar to MORIA. SCHLUPP only has problems in difficult situations like blocked corridors because such situations activate more fuzzy rules which are combined with the compositional rule of inference [6].

Now, for further optimization only the activated rules are considered. The optimization algorithm shown in figure 9 is derivated from genetic algorithms [7, 2]. The complete algorithm (fig.8) is subdivided into three loops. A driving path for the robot is given through a command list. The 1st loop terminates when the robot has worked out the command list. In each driving step and for all activated rules (2nd loop) a child population (3rd loop) is simulated and the child with the best fitness is selected for the next driving step. After each fitness calculation, the robots are reset to the same position and orientation to compare the behaviours of mother and child.

```
begin(1) initialize child(t), mother, simulation
    while (command list not empty)
    begin (2) step = step + 1
       while (activated rule set not empty)
       begin(3) t = 1
          while (child population not empty)
          begin(4) simulate(mother,child(t))
             exchange membership functions(child(t))
             fitness(t) = calc fitness(child(t))
             t = t + 1
             set old position(child(t))
          end (4)
          child(1) = select best child(child(1..t), fitness(1..t))
       end (3)
    end (2)
end(1)
```

Figure 8: The algorithm.

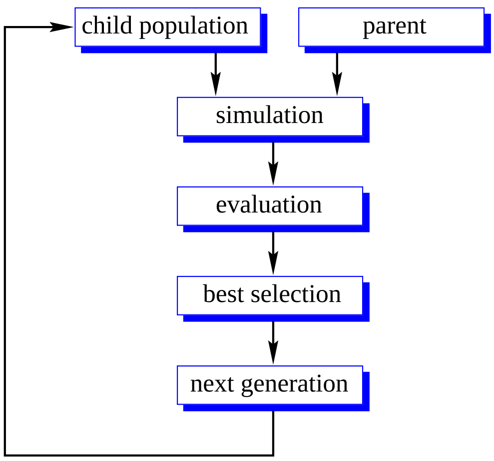

Figure 9: Optimization process.

The new child population is generated through the exchange of the membership functions of the activated rules in each control step (fig.10 and table 10).

No. of permutation	LEFT SPEED	RIGHT SPEED
1	LS_j	RS_j
2	LS_j	RS_{j-1}
3	LS_j	RS_{j+1}
4	LS_{j-1}	RS_j
5	LS_{j-1}	RS_{j-1}
6	LS_{j-1}	RS_{j+1}
7	LS_{j+1}	RS_j
8	LS_{j+1}	RS_{j-1}
9	LS_{j+1}	RS_{j+1}

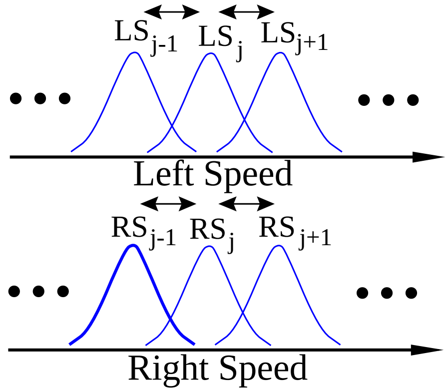

Figure 10: Neighbourhood exchange of membership functions for the activated rules.

In each controller step a population of nine mutated childs is generated. All nine permutations (table 10) with the neightbourhood membership functions of the output variables *speed left* and *speed right* are simulated together with MORIA's fuzzy controller. The fitness of the childs is determined by the metric distance to the mother:

$$\delta x(i) = x(i)_{MORIA} - x(i)_{SCHLUPP}, \quad \delta y(i) = y(i)_{MORIA} - y(i)_{SCHLUPP} \tag{1}$$

$$\bar{s} = \bar{v}_{MORIA} * \overline{time}_{elapsed} \approx 3.5 \tag{2}$$

$$Q = \begin{cases} 0 & \text{, if robot touched obstacle} \\ \sum_{i=1}^{nr_{steps}} \sqrt{\delta x(i)^2 + \delta y(i)^2} / (nr_{steps}\ \bar{s}) & \text{, else} \end{cases} \tag{3}$$

Obstacle hits or deadlocks are rated with a very low fitness value. The rule selection process is random but fair, i.e. a rule is selected once a time in a random order. After evaluating the child population, the combination with the best fitness is selected. The conversion of the rules is shown by the following example:

Rule Set MORIA (Initial)

IF COMMAND is left-rotation THEN SPEED is very-small, ANGLE is negative-very-big
IF FRONT-SENSOR is very-small AND FRONT-LEFT-SENSOR is very-small AND FRONT-RIGHT-SENSOR is small .... THEN SPEED is very-small, ANGLE is negative-very-big

Rule Set SCHLUPP (adapted)

IF COMMAND is left-rotation THEN SPEED-LEFT is negative-a, SPEED-RIGHT is positive-f
IF FRONT-SENSOR is very-small AND FRONT-LEFT-SENSOR is very-small AND FRONT-RIGHT-SENSOR is small .... THEN SPEED-LEFT is negative-a, SPEED-RIGHT is positive-f

The 1st rule is a manual command for left rotation. The 2nd rule is activated while the robot turns right. The new membership functions for SCHLUPP are denoted alphabetically in ascending order. Figure 11 shows the driving trajectory of SCHLUPP after the whole optimization process. The behaviour of both robots is nearly the same in all situations also in more difficult situations like “turning left” or “turning right” where more rules are activated and the total behaviour is essential.

4 Conclusion

In this paper, we have presented an adaptive learning method (mother-child teach-in). From a given well tuned prototype a new rule-base for a mobile platform with different kinematics is generated automatically. The new platform has the same behaviour as the prototype at the end of the teaching. The presented approach has been tested in our simulation environment.

Our first test results have proofed that the automatic adaption with predefined or adapted membership functions of a well tuned rule-base is possible. Through the two approaches applied, we have shown that a better solution is reachable. The behaviour adaption improves the fuzzy system by systematic exchange of the membership functions of the output variables *speed left* and *speed right*. The behaviour of the child is rated by the divergent movements to the mother.

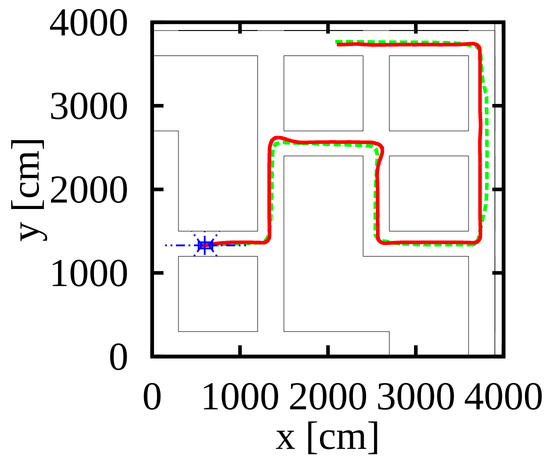

Figure 11: Driving trajectory of *MORIA* (dotted) and SCHLUPP. after optimization

It has been shown that the driving behaviour of a complex recurrent fuzzy system (child) can be improved. Our approach takes over the knowledge and the structure of the rule base and adapts the required mechanics. This makes the rule set reusable for robots with different kinematics, sensors and geometrical constraints, but with this approach the child will not reach a better behaviour than her mother, because the fitness value is oriented on the mothers behaviour.

Future work will concentrate on improving the behaviour of the child after the teach-in and adapting process. Therefore, the rating of the child behaviour has to be made independant of the mother. We estimate that further improvements can be reached through tuning the input membership functions as well as the state detection within the fuzzy system. But a lot of work remains to be done to decrease optimization time and increase the maintenances and comprehensibility when improving such complex and dynamic control systems. We think that all these constraints have to be considered in the structure logic of a system.

References

[1] Saman Halgamuge, *Advanced Methods for fusion of fuzzy systems and neual networks in intelligent data processing*, Dissertation, Technisch Hochschule Darmstadt, VDI-Verlag Düsseldorf, 1995.

[2] Hartmut Surmann, Andreas Kanstein, and Karl Goser, "Self-organizing and genetic algorithms for an automatic design of fuzzy control and decision systems", in *Proceedings of the First European Congress on Fuzzy and Intelligent Technologies, EUFIT'93, Aachen*, 7–10 Sept. 1993, pp. 1097–1104.

[3] Hartmut Surmann, Jörg Huser, and Liliane Peters, "Guiding and controlling mobile robots with a fuzzy controller", in *Fourth IEEE International Conference on Fuzzy Systems, Yokohama, Japan*, 20–24 Mar. 1995, pp. 83–88, Distinguished with *Robot Intelligence Award*.

[4] Hartmut Surmann, Jörg Huser, and Jens Wehking, "Path planning for a fuzzy controlled autonomous mobile robot", *Fifth IEEE International Conference on Fuzzy Systems, New Orleans, USA, 08. - 11.9.1996, to be published*, 1996.

[5] Alessandro Saffiotti, Enrique H. Ruspini, and Kurt Konolige, *A Multivalued Logic Approach to Integrating Planning and Control*, Artificial Intelligence Center, SRI International, Technical Note No. 533, Menlo Park, CA 94025, USA, 4/1994.

[6] E. H. Mamdani, "Application of fuzzy algorithms for control of a simple dynamic plant", *Proc. IEE*, vol. 121, Nr. 12, pp. 1585 – 1588, 1974.

[7] C. L. Karr, L. M. Freeman, and D. L. Meredith, "Improved fuzzy process control of spacecraft autonomous rendezvous using a genetic algorithm", *SPIE Intelligent Control and Adaptive Systems*, vol. 1196, pp. 274–288, 1989.