Fuzzy Logic Control of an Obstacle Avoidance Robot

Sng Hong Lian
Electronics and Communication Department
Singapore Polytechnic
500 Dover Road
Singapore
snghl@sp.ac.sg

Abstract

A fuzzy controller is used to control an obstacle avoidance mobile robot. In this classical problem, the aim is to guide a mobile robot along its path to avoid any static obstacles in front of it. Obstacle avoidance in real-time is a mandatory feature for mobile robots in a dynamically unknown environment. This controller presented here uses three sub-controllers. The outputs are summed to produce a concerted effort to control the motors, steering the robot away from obstacles. This fuzzy controller was implemented on a miniature robot. This robot is able to overcome its limitation on range accuracy to follow a left wall, maintaining a short distance from it, to avoid obstacles in front of it, and to decide whether a gap is wide enough for a "side-step" maneuver.

Keywords : fuzzy control, wall following, obstacle avoidance.

1.0 Introduction

Obstacle avoidance in mobile robot has been investigated in many instances [8][9]. The objective of this report is to show a novel technique to implement fuzzy control system for obstacle avoidance in real-time control. The motivation of using fuzzy control to solve this problem is that knowledge on avoiding obstacle could be easily expressed in linguistic form, and subsequently coded as fuzzy rules in the controller.

The mobile robot being used, as shown in figure 1, is called the Khepera[1]. In its basic configuration, is an excellent choice for investigation of robot behaviour. It comprises of a Motorola 68331 processor, driven by two d.c. motors coupled to the wheels through a 25:1 reduction gear. Incremental encoders are placed on the motor axes that give 24 pulses per revolution of the motor shaft. This allows a resolution of 600 pulses per revolution of the wheel, which is 0.08mm. Eight infra-red proximity sensors are placed round the robot as shown in figure 2. It has a small size, with diameter of 55 mm, and height of 30mm.

2.0 Fuzzy Controllers

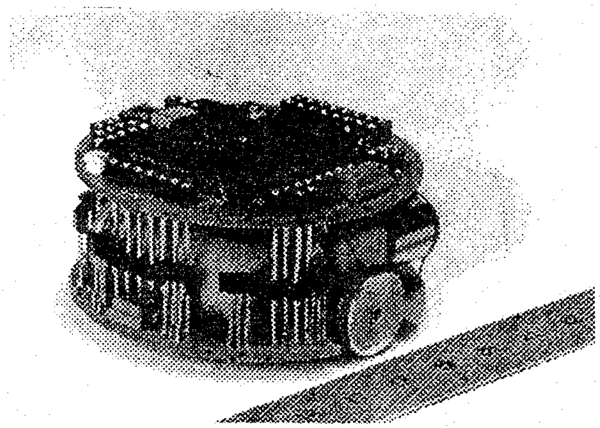


Figure 1 Khepera, the mobile robot

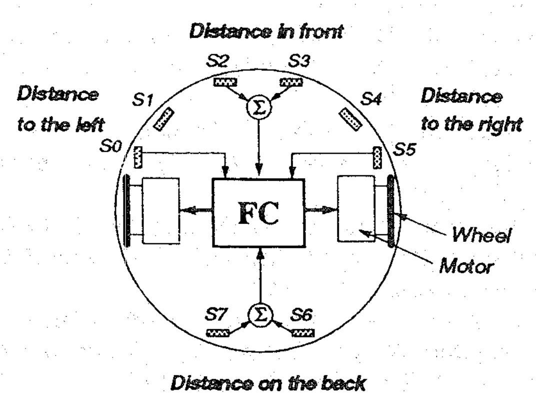


Figure 2 Position of sensors in Khepera

As the robot move, it is continuously monitoring for obstacles on it's left, front and right side. As it moves, information from it's sensors invariably change due to the dynamic unknown environment. With such multitude of input data, a hierarchical structured controller is designed. The structure of the controller is as shown in figure 3.

There are several reasons why a fuzzy controller should be designed in a hierarchical manner. The first, as demonstrated below, is that a hierarchical system has a small fraction of the number of fuzzy control rules compared to a non-hierarchical system. The second reason, as demonstrated, is that the structure of the rules in a hierarchical system tends to be simpler, based on fewer parameters. The hierarchical structure can also be implemented step by step designing each part of the system separately. Reliability and stability of each component of the controller could be tested individually. The behaviour of the whole system can then be found by examining the combined effect of the individual subsystems. This means that the stability and reliability results of the hierarchical control model are easier to ensure as compared to a corresponding non-hierarchical system [6]. A non-hierarchical system would be difficult to design and test. As larger control systems that control many more parameters are developed the difficulties in ensuring reliability and stability will become greater for non-hierarchical systems.

Rules in hierarchical fuzzy controllers[5]

Hierarchical fuzzy controllers require a fraction of the number of fuzzy rules that are required by a corresponding non-hierarchical system. Let *Left distance* be the distance of obstacle to the left of the robot, *Front distance* be the distance of obstacle in front and *Right distance* be the distance of obstacle on the right. Consider a fuzzy controller for the obstacle avoidance robot based on these parameters, which are represented by three fuzzy sets respectively. To specify fully the input space, 27 fuzzy rules are required for the non-hierarchical fuzzy associated memory (FAM) controller [7][8]. In [7], Kosko gives a detail discussion and in [8], Leon et al implemented such a controller. A hierarchical fuzzy controller based on three sub-controllers requires just 9 rules to specify the system behaviour, a significant difference.

Hierarchical control fuzzy rules[5]

A non-hierarchical FAM controller would have fuzzy rules of the form :

if (*Left distance* is Near) and (*Front distance* is *Medium*) and (*Right distance* is *Far*) then (*Left motor* is *Medium Positive*, *Right motor* is *Small Positive*) etc, 27 similar rules.

A hierarchical FAM controller would have fuzzy rules of the form :

if (*Left distance* is *Near*) then (*Left motor* is *Medium Positive*, *Right motor* is *Large Positive*);
if (*Front distance* is *Small*) then (*Offset* for *Left motor* is *Small Negative*, *Offset* for *Right motor* is *Large Negative*) etc, 9 similar rules.

It can be seen from the example above that the rules for the controllers in the hierarchical system depend on fewer parameters than the non-hierarchical system. This makes the rules easier to formulate and the system easier to test.

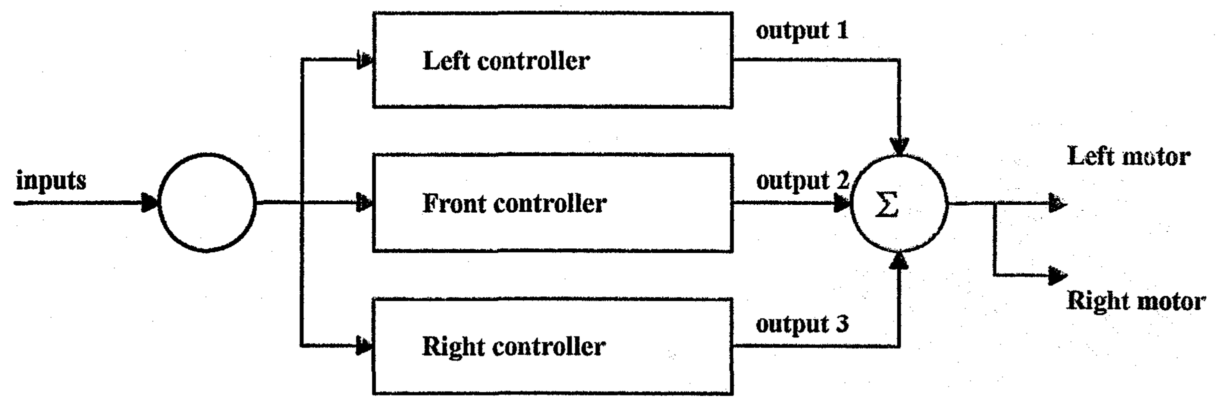


Figure 3. Structure of the fuzzy controller

2.1 Left controller

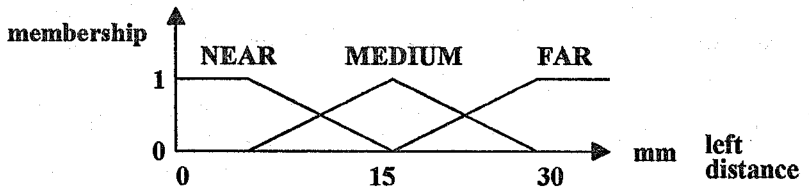
Three fuzzy sub-controllers are used. One for steering the robot away from obstacles on the left side, one for the front and the third for the right. The fuzzy rules are as shown in figure 4. The robot uses the Left controller for guiding itself to follow a left wall while moving along a corridor. It maintains a small distance from the wall and steers away from obstacles on the left. The Left controller has three fuzzy sets for its only parameter, *Left distance*, and it has three fuzzy rules. The robot will be moving at a *Medium Positive* speed and at a *Average distance* from the left wall. If the robot deviates further than the desired distance, the Left controller's output will correct the deviations by increasing the speed of the right motor. It the robot moves nearer to the wall, the Left controller's output will decrease the speed of the right motor.

2.2 Front controller

The Front controller likewise has three fuzzy sets for its only parameter, *Front distance*, and has three rules as shown in figure 5. As the robot must avoid any obstacle in front, the Front controller output offset values to the motors output computed from the Left controller once front obstacles are detected. This offset value is dependent on the nearness of the obstacle; if the obstacle is far, there will be little or no offset and if the obstacle is near, the offset will increase proportionally.

2.3 Right controller

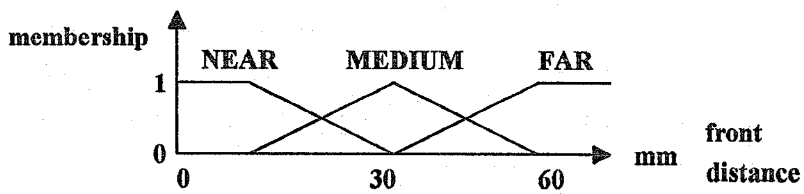
It is quite different from the Left and Front controller because the offset action is only exerted if and only if the obstacle on the right is very near such that by moving forward, the robot will crash into the obstacle. This will



	NEAR	MEDIUM	FAR
Left motor	MP	MP	MP
Right motor	SP	MP	LP

SP : Small positive speed;
MP : Medium positive speed;
LP : Large positive speed

Figure 4. Fuzzy control rules for wall following



	NEAR	MEDIUM	FAR
output 2 Left motor	SN	ZE	ZE
output 2 Right motor	LN	MN	ZE

ZE : No offset;
SN : small negative offset;
MN : medium negative offset;
LN : large negative offset.

Figure 5. Fuzzy control rules for steering away from front obstacles

allow the robot to execute a side-step maneuver for the situation where there is no front obstacle but a gap between the wall and a right obstacle. If the gap is big enough, the robot will move forward going through the gap else it must take a side-step. The right controller will output a large offset to turn the robot to the right.

3.0 Tuning

In this project, there are three fuzzy sub-controllers which are interconnected. All the controllers have fuzzy members with triangular functions. The tuning of the robot is performed first on the Left controller, then the Front controller and finally, the Right controller. Tuning is performed by adjusting contribution weights connected to each inference rule in each controller [4][5]. These weights will determine how much the rules affect the speeds of the motors. Initially, all weights are set to 1.

To tune the Left controller, a straight wall is erected. The movement of the robot is observed and the weights connected to the output fuzzy sets are tuned. The movement of the robot must be parallel to the straight wall. To tune the Front controller, a front wall is erected perpendicular to the left wall. When the robot approaches the front wall, it will make a 90 degrees turn. The turning action is observed and the rules are tuned by adjusting its corresponding weights again. Finally, the Right controller is tuned. Obstacle is positioned to the right of the robot and a front gap is created. Tuning is performed for a desired gap size just enough for the robot to go through. If the gap size is greater than or equal to this desired size, the robot will go through. If the gap is lesser than the desired size, the robot must executed a side-step maneuver by continuously turning to the right until it is able to circumvent the obstacle.

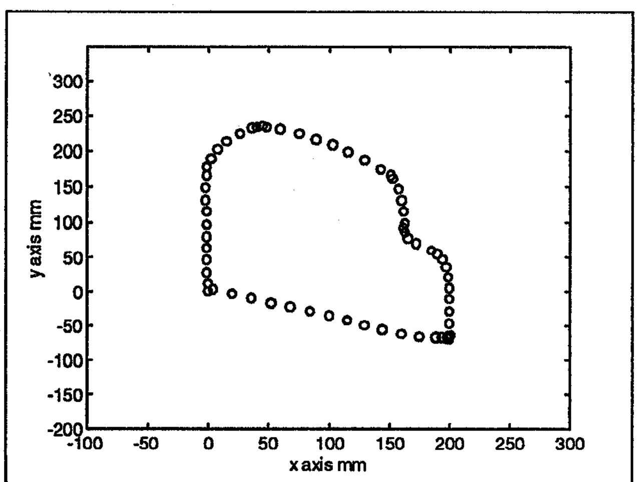


Figure 6. Trajectory of obstacle avoidance robot in the test platform.

4.0 Results

The performance of the robot is tested. An enclosed region made of expanded polystyrene is created on a platform with surrounding walls. The platform is constructed with corridors and walls which have gradual bends and sharp corners. The robot is positioned in the platform and allows to make a clockwise trajectory. The enclosed platform and the robot trajectory is as shown in figure 6. The robot is able to make a clockwise trajectory within the test platform maintaining a short distance from the left wall. It can negotiates gradual bends and sharp corners gracefully. It could also decides whether a gap between the wall and the right obstacle is big enough to go through or to make a turn. On experimenting with dead-end passage, it could also smoothly make an about turn. Figure 7 shows the consistent behaviour of the robot when it goes for two rounds in an enclosed rectangular premise. The trajectories for both rounds are identical and they coincide with each other.

5.0 Conclusions

A hierarchical structured fuzzy controller is implemented in an obstacle avoidance robot. The hierarchical nature has greatly reduced the number of inferencing rules required and make the system easier to tune and test. With dynamic real life environment created on a test platform, which comprises of gradual and sharp corners, the movement of the vehicle is smooth and consistent in maintaining a fixed distance from the surrounding walls. The robot is also able to make an about turn when it meets with a dead-end.

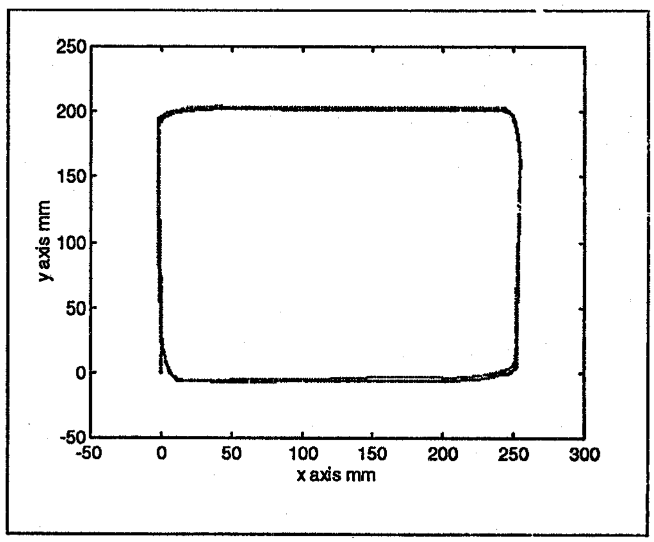


Figure 7. Trajectory of robot in a rectangular premise.

Acknowledgement

This work is completed in LAMI-EPFL (Swiss Federal Institute of Technology, Lausanne), Switzerland. I would like to specially thank Professor J.D. Nicoud of LAMI-EPFL for his kind assistance and special arrangement, Mr. Masakazu Yamamoto of Dept. of Precision Machinery Engineering, Faculty of Engineering, University of Tokyo and Ms Godjevac Jelena of LAMI-EPFL for their assistance to make this project possible.

References

[1] F.Mondada, E.Franzi and P.Ienne, Mobile robot miniaturization : A tool for investigation in control algorithms. Informatik, pp17-20, Feb.1994.

[2] Tay, S.L., Nah, S.L. & Ng, W.T. (1993), Self Organising Control of an Inverted Pendulum using a Neuro-Fuzzy Controller, Proceedings First Asian Fuzzy Systems Symposium 1993, Singapore, pp 567- 573.

[3] Messom,C.H. & Sng, H.L. (1994),The Development of a Fuzzy Hierarchical Servo Controller for the Inverted Pendulum, 2nd Singapore International Conference on Intelligent Systems, November 1994. pp B147-B152.

[4] Cox, E. (1993), Adaptive Fuzzy Systems, IEEE Spectrum, February 1993, pp 27- 31.

[5] Sng, H.L. & Messom, C.H. (1994), Adaptive Fuzzy Pole Balancing Robot, 2nd Singapore International Conference on Intelligent Systems, November 1994. ppB141-B146.

[6] Kawaji, S. & Maeda, T. (1991), Fuzzy Servo Control System for an Inverted Pendulum, Fuzzy Engineering toward Human Friendly Systems, IFES '91.

[7] Kosko, B. (1992), Neural Networks and Fuzzy Systems, Prentice Hall International Inc, Singapore.

[8] Deist L.A. & Fourie C.J. (1993), Fuzzy Algorithm for the Control of a Mobile Robot, 2nd International workshop on Emerging Technologies and Factory Automation Proceedings p42-50.

[9] Li, Wei, (1994) Perception-action Behaviour Control of a Mobile Robot in Uncertain Environments Using Fuzzy Logic, International Conference on Intelligent Robots and Systems, p439-446, Vol 1.