Defence Science Journal, Vol. 60, No. 4, July 2010, pp. 387-391

Fuzzy Support Vector Machine-based Multi-agent Optimal Path Planning Approach to Robotics Environment

Gireesh Kumar T.[*], K.J Poornaselvan[#] and M. Sethumadhavan[*]

[*]*Amrita Vishwa Vidyapeetham, Coimbatore*
[#]*Govt. College of Technology, Coimbatore*
E-mail: [*]gireeshkumart@gmail.com, [#]poornakj@yahoo.co.in, m_sethu@cb.amrita.edu

ABSTRACT

A mobile robot to navigate purposefully from a start location to a target location, needs three basic requirements: sensing, learning, and reasoning. In the existing system, the mobile robot navigates in a known environment on a predefined path. However, the pervasive presence of uncertainty in sensing and learning, makes the choice of a suitable tool of reasoning and decision-making that can deal with incomplete information, vital to ensure a robust control system. This problem can be overcome by the proposed navigation method using fuzzy support vector machine (FSVM). It proposes a fuzzy logic-based support vector machine (SVM) approach to secure a collision-free path avoiding multiple dynamic obstacles. The navigator consists of an FSVM-based collision avoidance. The decisions are taken at each step for the mobile robot to attain the goal position without collision. Fuzzy-SVM rule bases are built, which require simple evaluation data rather than thousands of input-output training data. The effectiveness of the proposed method is verified by a series of simulations and implemented with a microcontroller for navigation.

Keywords: Fuzzy logic, fuzzy support vector machine, FSVM, optimal path planning, multi-agent, mobile robot, robotic navigation

1. INTRODUCTION

A robot is autonomous when it is able to move purposefully with no intervention from a human user in an un-engineered real-world environment. The development of techniques for autonomous navigation in a real-world environment constitutes one of the major trends in the current research on robotics. By incorporating multi-agent optimal path planning[3,9,10] and support vector machine (SVM) learning, special efficiency for robotic navigation can be achieved. One of the important aspects, that are still deemed important to consider in mobile robots, is obstacle avoidance. A pertinent problem in autonomous navigation is the need to cope with a large amount of uncertainty that is inherent of natural environments, and to respond reactively to unforeseen events as soon as these are perceived. Fuzzy logic with support vector machine is an adequate method to obtain certain and finite data in an optimal manner. However, this approach proves that it is enough for areas of highest danger coefficients to cause the robot to change direction, which consequently reduces the number of fuzzy rules that control the robot motion[9,10].

An SVM learns the decision surface from two distinct classes of the input points[7,8]. In many applications, each input point may not be fully assigned to one of these two classes. In this paper, a fuzzy membership has been applied to each input point and reformulated the SVMs such that different input points can make different contributions to the learning of decision surface.

2. SYSTEM STRUCTURE

Figure 1 shows the integration of a microcontroller (89C52) with the features of the robot. This architecture uses five sensors out of which the three proximity sensors are used to detect the obstacles, one to detect the colour and the last one to detect the pit in the navigation path[2,6]. Two stepper motors were used to navigate the robot which will be simulated by the stepper motor driver agent.

3. FUZZY LOGIC CONTROLLER

3.1 Decision-making System

For decision-making, they can use either type 1 fuzzy or type 2 fuzzy as per the requirement[4,5]. The four principal

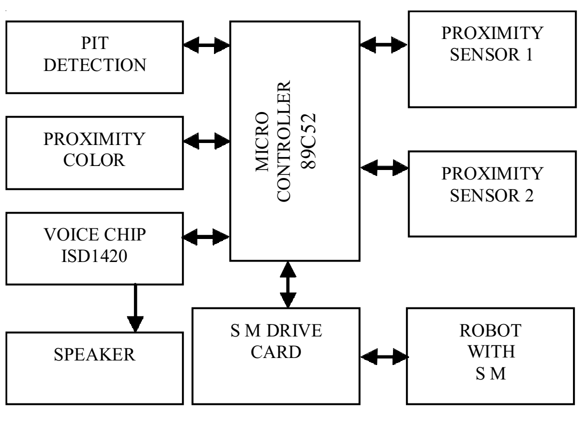

Figure 1. Integration of a microcontroller.

Received 22 September 2009, Revised 18 March 2010

components of type 1 fuzzy decision-making systems are:

a) *The fuzzification interface*: This determines the input and output variables and maps these into linguistic variables that are to be displayed on a universe of discourse.

b) *The knowledge base*: This is a part of expert systems that contains the domain knowledge. Membership functions and control rules are decided by the experts at this point, based on their knowledge of the system.

c) *The decision-making logic*: This treats a fuzzy set as a fuzzy proposition. One fuzzy proposition can imply another, and two or more fuzzy propositions can be associated by a Boolean connectivity relation to infer a final fuzzy proposition.

d) The defuzzification interface converts the fuzzy output into a crisp value.

A type 1 fuzzy set has a grade of membership that is crisp, whereas a type 2 fuzzy set has grade of membership that are fuzzy, so it is called fuzzy-fuzzy set. As the type 1 gives only sub-optimal solution, type 2 fuzzy is good in dealing with uncertainty.

4. SUPPORT VECTOR MACHINES

SVM is a machine learning technique developed on statistical learning theory. For machine learning tasks involving pattern classification, multi-sensors information fusion, nonlinear system control, etc, SVMs have become an increasingly popular tool. SVMs are a set of related supervised learning methods used for classification and regression. Viewing input data as two sets of vectors in an *n*-dimensional space, an SVM will construct a separating hyperplane in that space, which maximises the margin between the two data sets. To calculate the margin, two parallel hyperplanes are constructed, one on each side of the separating hyperplane, which are pushed up against the two data sets. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighbouring data points of both classes, since in general, the larger the margin, the better the generalisation error of the classifier[8,11].

4.1 Multi-class Support Vector Machine

For the conventional SVM, an n class problem is converted into n two-class problem and for the i^{th} two-class problem, class i is separated from the remaining class with decision function that classifies class i and remaining classes be

$$D_i(\boldsymbol{x}) = w^t{}_i \boldsymbol{x} + b_i \quad (1)$$

SVM, if for the input vector $\boldsymbol{x}$

$D_i(\boldsymbol{x}) > 0$ is classified for i, $\boldsymbol{x}$ is classified in to class i.

Let the decision function for class i against class, with the maximum margin[7,8], be

$$D_{ij}(\boldsymbol{x}) = w^t{}_{ij} \boldsymbol{x} + b_{ij} \quad (2)$$

where $D_{ij}(\boldsymbol{x}) = D_{ji}(\boldsymbol{x})$ for the input vector $\boldsymbol{x}$ the authors calculated

$$D_i(\boldsymbol{x}) = \sum_{i=1,\ldots,n}^{n} sign\left(D_{ij}(x)\right) \quad (3)$$

and classified x into the class

$$\arg\max_{i=1,\ldots,n} D_i(\boldsymbol{x}) \quad (4)$$

4.2 Fuzzy Support Vector Machine

An SVM learns the decision surface from two distinct classes of the input points. In many applications, each input point may not be fully assigned to one of these two classes. In this paper, a fuzzy membership was applied to each input point and reformulated the SVMs such that different input points could make different contributions to the learning of decision surface.

Class i is defined as a one-dimensional membership functions $m_{ii}(\boldsymbol{x})$ on the direction orthogonal to the optimal separating hyperplanes $D_j(\boldsymbol{x}) = 0$ as given below

1. For $i = j$

$$m_{ii}(\boldsymbol{x}) = \begin{cases} 1 & \text{for } D_i(\mathbf{x}) > 1 \\ D_i(\mathbf{x}) & \text{otherwise} \end{cases} \quad (5)$$

2. For $i \neq j$

$$m_{ij}(\mathbf{x}) = \begin{cases} 1 & \text{for } D_j(\mathbf{x}) < -1 \\ -D_i(\mathbf{x}) & \text{otherwise} \end{cases} \quad (6)$$

The class i membership function of $\boldsymbol{x}$ is defined using the minimum operator for

$$m_i(\boldsymbol{x}) = \min_{j=1,\ldots,n} m_{ij}(\boldsymbol{x}) \quad (7)$$

Now the datum $\boldsymbol{x}$ is classified into the class

$$\arg\max_{i=1,\ldots,n} m_i(\boldsymbol{x})$$

if $\boldsymbol{x}$ satisfied

$$D_k(x) \begin{cases} > 0 & \text{for } k = i, \\ \leq 0 & \text{for } k \neq i,\ k = 1, \ldots\ldots n \end{cases} \quad (8)$$

and $m_k(\boldsymbol{x})$ is given by

1. $k \in i_1, \ldots\ldots\ldots\ldots, i_1$

$$m_k(\boldsymbol{x}) = \min_{j=i1,\ldots,il \ \ j\neq k} -D_j(\boldsymbol{x}) \quad (9)$$

2. $k \neq j \ (j = i_1, \ldots, i_l)$

$$m_k(\boldsymbol{x}) = \min_{j=i1,\ldots,il} -D_j(\boldsymbol{x}) \quad (10)$$

Thus, the maximum degree of membership is achieved among $m_k(\boldsymbol{x})$, $k = i1, \ldots il$ (Fig. 2)[7,8].

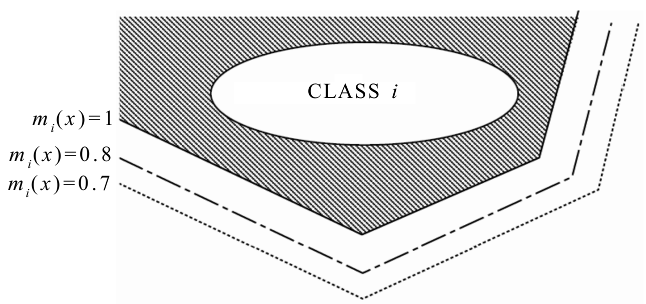

Figure 2. Contour lines of the class i membership function.

5. IMPLEMENTATION DETAILS

Architecture: The robotic system architecture used in this paper consists of two layers. The hardware layer is a collection of modules communicating with the robot's hardware devices such as infrared sensors and motors. One module can be shared by two or more agents, which reduces redundancy in coding. The development tools used is C language and Keil software, which is a part of Microsoft turbo C and 89C52 IC.

5.1 Multi-agent System

Figure 3 depicts the simplified diagram representing the multi-agent system using a interface agent. The agent basically interacts with the other components of the system by manipulating information on the interface agent. The information on the interface agent may represent facts, assumptions, and deductions made by the system during the course of solving the problem. An agent is a partial problem solver which may employ a different problem-

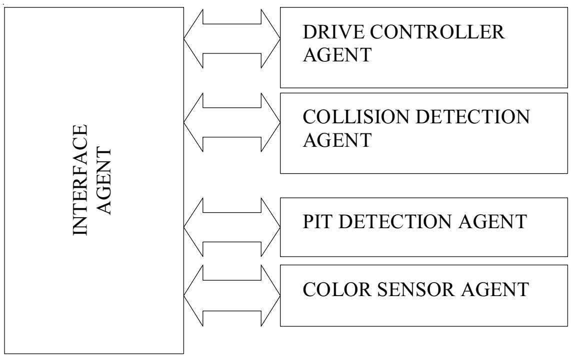

Figure 3. Multi-Agent system.

solving strategy and contribute to the solution by viewing the information on the interface agent. The system has four independent agents: fuzzy collision detector, pit detection agent, colour sensor, and drive controller. Arrows in Fig. 3 represent the flow of information. The diagram shows that all four agents are allowed to read/ write information on the interface agent. Each of the four agents basically executes their tasks independently using the information on the interface agent and posts any result back to the interface agent[4].

5.2 Intelligent Agents

5.2.1 Fuzzy SVM-based Collision Avoidance

The agent called the fuzzy collision detector[1,3] is a fuzzy SVM-based collision avoidance controller. The fuzzy logic controller has one input fuzzy set for the sensor value and three output fuzzy sets for linear distance, velocity, and turn-angle. Each set is defined by one or more membership functions that map numeric values onto linguistic terms; each input point may not be fully assigned to one of these two classes. In this paper, the authors have applied a fuzzy membership to each input point and reformulated the SVMs such that different input points can make different contributions to the learning of decision surface.

The fuzzy-based agent was fed with sensor values as an input, acquired from a set of infrared proximity detectors. The values were fuzzified with designated linguistic terms (near, medium, and far). Among three output fuzzy sets, the turn-angle fuzzy set has been uniquely defined. The angle lies between –30°and 30° which acts as a default. The total angle of 60° is divided into six amplitudes represented by six member functions, and each of which is associated with the following linguistic terms: positive-left (PL), negative-left (NL), positive-centre (PC), negative-centre (NC), positive-right (PR), and negative-right (NR), as shown in Fig. 4.

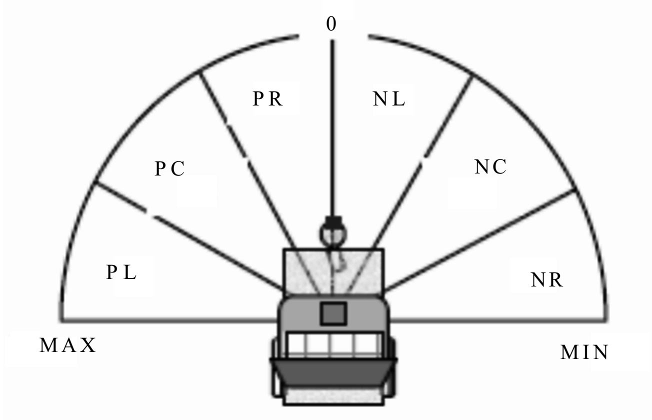

Figure 4. Turn angle fuzzy set.

5.2.2 Pit Detection Sensor

An apparatus and method for uniquely detecting pits on a smooth surface by irradiating an area of the surface; separately sensing radiation scattered from the surface in the near-specular region indicative of a pit and in the far-specular region indicative of a flaw and producing signals representative thereof normalising the near-specular signal with respect to the far-specular signal to indicate a pit[5].

5.2.3 Colour Sensor

Colour detection: Identifying the presence or absence of a specific color;

Colour measurement: Identifying a colour based on its red, green, and blue components;

Colour control: Using the colour sensor as part of a closed-loop feedback system to produce and maintain a required colour.

5.2.4 Drive Controller Agent

The agent primarily holds responsibility for the robot's actuator via the device driver that controls motors through the stepper control module. The agent is made of modules responsible for the motor initialisation and termination, the communication between layers, and the manoeuvring of the robot[5].

5.2.5 Interface Agent

The interface agent operates as a central repository for all shared information and a communication medium for all the agents.

5.3 Multi-agent Optimal Path Planning

Problems of multi-agent robot systems control have got significance[4]. Each multi-agent robot system has some transport subsystem, which consists of several mobile robots. The authors have developed a method based on graph optimisation algorithms to control such mobile robot group. Novelty of the developed multi-agent path planning algorithm[6,2] is as follows:

- Mobile robots are considered as dynamic obstacles.
- Graph representation of common environment mode is used for path planning.
- Each edge of the graph has two weights, distance and motion time (speed).
- Weights of edges can be modified during path planning.
- The quickest path is planned (time optimisation).
- Expert rules for speed and path correction are synthesised to provide collision avoidance.

These algorithms provide global optimality. Multi-agent path planning algorithm also provides robots collision avoidance. As the authors have two weights for graph such as distance and time required to travel, they can use *AO** algorithm to find shortest path[9,10]. They need to update the weight of graph during particular interval. Ant colony optimisation also can be used to control a group of robots.

6. EXPERIMENTATION AND RESULTS

The navigation technique described was implemented in C and interfaced with the microcontroller using Keil. For the test, the start point, the target point and velocity of the point robot were specified. It was repeated for different navigation tests in real time. Figure 5 shows an example of robot navigation in cluttered environments.

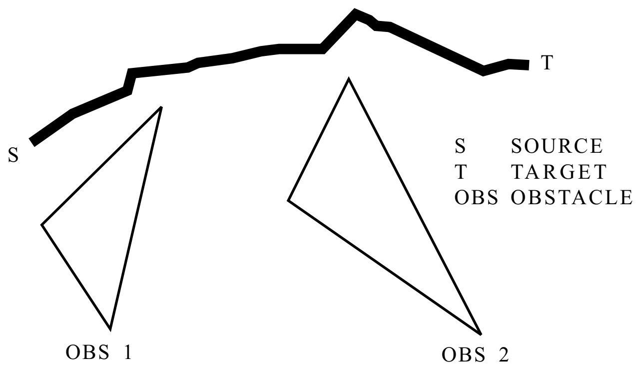

Figure 5. Sample navigation result.

7. CONCLUSIONS

The multi-agent optimal path planning approach provides an extra level intelligence. System can be enhanced with incremental FSVM[11] with Kernel will give better result in group robots. This paper deals with the real-time navigation of a mobile robot in a totally unknown environment. Fuzzy SVM is the best tool to incorporate human procedural knowledge into a robot. Type 2 fuzzy gives a better approach to solve uncertainty. Based on fuzzy logic and support vector machine, a collision-free technique has been proposed that partitions a robot space into eighteen zones of danger, where just four of them trigger the fuzzy rules. This proposal has been validated in different unknown environments cluttered with static and dynamic obstacles and has proven to give the robot, a means of safely reaching the target.

REFERENCES

1. Fayad, Carole & Phil, Webb. Optimised fuzzy logic based algorithm for a mobile robot collision avoidance in an unknown environment. Manufacturing Engineering. Department, University of Nottingham, UK, 2002.
2. Yantogo, Xiaolei Yin & Alan, Bowling. Navigability of multi-legged robots. *IEEE Explore*. Feb. 2008, 1-8.
3. Emmanuel, A.; Merchant-Cruz & Morris, Alan S. Fuzzy-GA-based trajectory planner for robot manipulators sharing a common workspace. *IEEE Transactions,* August 2006.
4. Huq, Rajibul; George, K.I.; Mann & Gosine, Raymond G. Behaviour-modulation technique in mobile robotics using fuzzy discrete event system. *IEEE Transactions.* October 2006.
5. Levine, S.P. *et al.* The NavChair assistive wheelchair navigation System. *IEEE Trans. Rehabilitation Eng.,* 1999, **7**(4), 443-51.
6. Miller, D. Assistive robotics: An overview. *In* Assistive technology and AI. LNAI-1458 edited by Mittal, *et al.* Berlin, Springer-Verlag, 1998. pp. 126-36.
7. Shigeo, Abe & Takuya, Inoue. Fuzzy support vector machines for pattern classification. *In* Proceeding of International Joint Conf. on Neural Networks (IJCNN'01'), Vol. **2**, July 2001, pp. 1449-1454.
8 Shigeo Abe, Takuya. Fuzzy support vector machines for multiclass problems. *In* Proceeding of the European Symposium on Artificial Neural Networks, Bruges (Belgium), April 2002. pp. 113-18.
9. Poornaselvan, K.J.; Gireesh Kumar, T. & Vijayan, V.P. Agent-based ground flight control using type-2 fuzzy logic and hybrid ant colony optimisation to a dynamic environment. 1st International Conference on Emerging Trends in Engineering and Technology, July 2008. pp. 343-48,.
10. Gireesh Kumar, T. & Vijayan, V.P. A multi-agent optimal path planning approach to robotics environment. *In* International Conference on Computational Intelligence and Multimedia Applications, Dec 2007. pp. 400-404.
11. Evgeniou,T.; Pontil, M. & Poggio, T. Regularisation networks and support vector machines. *In* Advances in Large Margin Classifiers edited by A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans. Cambridge, MA, MIT Press, 2000. **203**. pp.171.

Contributors

Mr Gireeshkumar T. received his MTech (Computer and Information Science) from Cochin University of Science and Technology, Cochin, Kerala, in 2002. He is currently pursuing PhD (Artificial Intelligence) Anna University, Chennai. He is now an Assistant Professor (Sr. Grade) with TIFAC CORE in Cyber Security at Amrita Vishwa Vidyapeetham University, Coimbatore. His research interests are artificial intelligence, machine learning and algorithms.

Dr K.J. Poornaselvan received ME (Elect and Electro Engg) from Bharathiyar University, Coimbatore. He obtained his PhD (Elect Engg) from Anna University, Chennai, in 2006. He is currently an Asst Prof in Electrical and Electronics Engg at the Government College of Technology, Coimbatore. His area of interest are in the field of control systems and artificial intelligence.

Dr M. Sethumadhavan received his PhD (Number Theory) from Calicut Regional Engineering College (presently a National Institute of Technology). He is a Professor of Mathematics and Computer Science in Amrita Vishwa Vidyapeetham University, Coimbatore and is currently the Head, TIFAC CORE in Cyber Security. His area of interest is cryptography.