Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 37–43

ISBN 978-83-60810-27-9
ISSN 1896-7094

PSO based modeling of Takagi-Sugeno fuzzy motion controller for dynamic object tracking with mobile platform

Meenakshi Gupta, Laxmidhar Behera, and K. S. Venkatesh
Department of Electrical Engineering,
Indian Institute of Technology Kanpur, Kanpur, India
Email: {meenug, lbehera, venkats}@iitk.ac.in

***Abstract*—Modeling of optimized motion controller is one of the interesting problems in the context of behavior based mobile robotics. Behavior based mobile robots should have an ideal controller to generate perfect action. In this paper, a nonlinear identification Takagi-Sugeno fuzzy motion controller has been designed to track the positions of a moving object with the mobile platform. The parameters of the controller are optimized with Particle swarm optimization (PSO) and stochastic approximation method. A gray predictor has also been developed to predict the position of the object when object is beyond the view field of the robot. The combined model has been tested on a Pioneer robot which tracks a triangular red box using a CCD camera and a laser sensor.**

I. Introduction

OBJECT detection and tracking is an essential ingredient of any motion planning controller employed for mobile robot navigation. Mobile robot navigation is known as the ability of a robot to act based on its knowledge and sensor values in order to reach its goal position as efficiently and as reliably as possible [1]. Wide variety of sensors such as sonar, laser range finder, infra-red, Global Positioning System (GPS) and vision are used for mobile robot navigation. The vision based navigation is widely used [2], since vision gives the rich information about the surroundings. Vision is an attractive sensor as it helps in the design of economically viable systems with simpler sensor limitations. It facilitates passive sensing of the environment and provides valuable information about the scene that is unavailable to other sensors [3].

An ample of work has been done on vision based object tracking. Ramesh et.al. in [4] proposed the Mean shift algorithm for object tracking that can be used for the images with static distribution. The Continuously Adaptive Mean Shift Algorithm (CAMShift), which is an adaptation of the Mean Shift Algorithm has been proposed by Bradski [5] to track the head and face movement using a one dimensional histogram (hue) consisting of quantized channel from the HSV color space. CAMShift operates on a probability density image obtained by histogram back-projection. In this paper, a hybrid CAMShift algorithm [6], that overcomes the assumption of single hue, has been used for object tracking. The computational cost of mean matching algorithm used in the hybrid CAMShift algorithm is high. So, instead of using mean matching algorithm for the detection of object when the CAMShift fails to detect the object, a gray predictor has been used to predict the position of the object. Gray predictor re-initializes the CAMshift window not only when CAMshift fails to detect the object but also when the object goes out from the robot's view.

The outputs of the hybrid CAMShift algorithm are the centroid coordinates of the object in the image frame. In order to obtain the global position and orientation of the object or even just to determine their relative pose, various algorithms of calibration and transformation are required. All the proposed approaches formulate the vision-based navigation problem as a two-step process: first, to transfer the visual features back to pose information, and then make a motion plan in the pose space. The calibration techniques, that transfer the visual features from image space to pose space introduce unnecessary uncertainty into the system. In this paper, a simple transformation technique has been proposed to transfer centroid coordinates of the object from image frame to the robot frame.

Once the coordinates of the object centroid are known in the robot frame, the next task is to design motion controller to effectively track the object. Since the primary focus of machine intelligence and advanced robotics is to capture the human faculties in the robot, fuzzy logic controllers are often a good choice. These controllers are developed to utilize human expert knowledge in controlling various systems and they have capability to express knowledge in the form of linguistic rules. Among various fuzzy modelling themes, the Takagi-Sugeno (T-S) model has been one of the most popular frameworks as it exhibits both high nonlinearity and simple structure [7][8]. In this paper, a T-S fuzzy controller has been modeled to control the motion of the robot while tracking the object. The structure identification of the premise part (i.e. membership functions) of rules of T-S fuzzy controller is carried out using PSO [9], while the identification of consequent part (i.e. weight parameters) of rules of T-S fuzzy controller is carried out using stochastic approximation method [10].

II. Outline of our approach

The goal of this work is to design a vision and laser sensor based optimized motion controller for the mobile robot to make it track the moving object in an effiecient manner. The outline of our approach for tracking the moving object is illustrated in figure 1. The first step inculdes the initialization of the robot and selection of object in the current frame. Next, frames are captured from the resulting video stream and the hybrid CAMShift algorithm is run over to detect the presence of the object in the scene. In case object is detected the algorithm returns the centroid coordinates of the object in image frame. If hybrid CAMShift fails to detect the object then the output of algorithm will be origin coordinates. Failure of the algorithm occurs either due to fast motion of the object or when the object goes out from the robot's view. In case of failure of the algorithm, a gray predictor is used to predict the centroid coordinates of the object. If the predicted x-coordinate lies in the range of image width, then the failure of the algorithm is due to fast motion of the object. In this case, the center coordinates of the CAMShift window are replaced by the predicted coordinates of the gray predictor. In case, when predicted x-coordinate does not lies in the range of image width, the reason of the algorithm failure is the absence of the object in the robot's view. The robot is then commanded to turn by an angle, calculated using the predicted x coordinate as follows:

$$\beta = -\frac{\phi}{2} + \frac{x}{x_{max}}.\phi \qquad (1)$$

where, β is the required turning angle, x_{max} is the maximum x coordinate of the image (width of the image in pixels), ϕ is the view angle of the camera.

The CAMShift window is then reinitialized with the center x coordinate of the window as $(x_{max}/2)$ and center y coordinate of the window as the predicted y coordinate of the gray predictor. In both the cases, the CAMShift window size is taken to be equal to its initial window size. Once the object centroid coordinates are obtained in the image frame, a coordinate converter, as described in section IV, is used to transform the object centroid coordinates from the image frame to the robot frame. The object centroid coordinates in the robot frame are then sent to an optimized T-S fuzzy motion controller, described in section VI, to generate desired translational and rotational velocities for the robot. The robot is then commanded to move with the translational and rotational velocities as generated by the controller.

Rest of the paper is organized as follows. In section III, the gray fuzzy predictor is discussed to predict the position of the object when it goes out from the robot's view or the hybrid CAMShift algorithm fails to detect the object. Section IV gives the details of the transformation of object coordinates from the image frame to the the robot frame. Section V briefly explains the PSO method and the T-S fuzzy model. Section VI presents modelling of the T-S fuzzy controller for the object tracking using PSO and stochastic approximation method. Experimental results are presented in section VI and finally the paper is concluded in section VII.

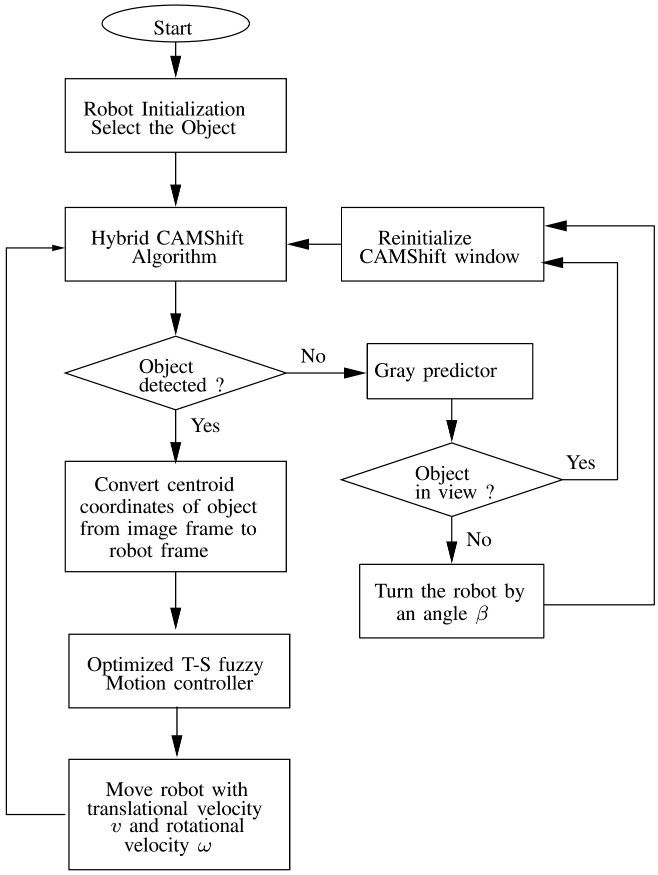

Fig. 1. flow diagram of our approach

III. Gray predictor

A system with partial known information and certain unknown information is defined as a gray system. The gray theory, originally developed by Deng [11], employed the method of data generation instead of statistic regulation, to obtain more regular generating sequence from those initial random data. The gray prediction is to establish a gray model extending from the past information to the future based upon the past and present known or undeterminate information. Then the gray model can be used to predict the future variation trend of the system.

Gray Prediction model for tracking The procedure of gray prediction model is as follows:

- Establish the initial sequence from observed data

$$\mathbf{x}^{(0)} = (x^{(0)}(1), x^{(0)}(2), x^{(0)}(3), ..., x^{(0)}(n) \qquad (2)$$

  where, $x^{(0)}(i)$ represent the base line data with respect to time i.
- Generate the first-order accumulated operation sequence (AGO) sequence $\mathbf{x}^{(1)}$ based on the initial sequence $\mathbf{x}^{(0)}$

$$\mathbf{x}^{(1)} = (x^{(1)}(1), x^{(1)}(2), x^{(1)}(3), ..., x^{(1)}(n) \qquad (3)$$

  where,

$$x^{(1)}(k) = \sum_{i=0}^{k} x^{(0)}(i) \qquad (4)$$

- Compute the mean value of the first-order AGO sequence:

$$z^{(1)}(k) = 0.5x^{(1)}(k) + 0.5x^{(1)}(k-1) \tag{5}$$

- Define the first-order gray-differential equation of sequence $\mathbf{x}^{(1)}$ as:

$$\frac{dx^{(1)}(k)}{dk} + az^{(1)}(k) = b \tag{6}$$

where, a and b express the estimated parameters of gray prediction model.

- Utilize the least square estimation, we can drive the estimated first-order AGO sequence $\hat{x}^{(1)}(k+1)$ and the estimated inversed AGO sequence $\hat{x}^{(0)}(k+1)$ as follows:

$$\hat{x}^{(1)}(k+1) = \left[x^{(0)}(1) - \frac{b}{a}\right]e^{-ak} + \frac{b}{a} \tag{7}$$

$$\hat{x}^{(0)}(k+1) = \hat{x}^{(1)}(k+1) - \hat{x}^{(1)}(k) \tag{8}$$

where, parameter a and b can be conducted by following equations:

$$\begin{bmatrix} a \\ b \end{bmatrix} = (B^T B)^{-1} B^T y \tag{9}$$

$$B = \begin{bmatrix} -\frac{1}{2}(x^{(1)}(1) + x^{(1)}(2)) & 1 \\ -\frac{1}{2}(x^{(1)}(2) + x^{(1)}(3)) & 1 \\ \cdots & .. \\ -\frac{1}{2}(x^{(1)}(n-1) + x^{(1)}(n)) & 1 \end{bmatrix} \tag{10}$$

$$y = \left[x^{(0)}(2), x^{(0)}(3), x^{(0)}(4), ..., x^{(0)}(n)\right]^T \tag{11}$$

In this paper, we modeled two gray predictor to predict the x and y coordinate of the object centroid. We used last six centroid coordinates to generate accumulated operation sequence.

IV. Transformation of the object coordinates from the image frame to the robot frame

To design any motion controller for the robot, it is necessary to transform the coordinates of the gravity center of the object from the image frame to the robot frame. As x coordinate of the object in the image frame is invariant to the size (width) of the object, it can be used to calculate the direction of the object with respect to the robot. But y coordinate of the object in the image frame varies with the size (height) of the object, so, it cannot be used to determine the distance of the object with respect to the robot. To get the accurate coordinates of the gravity center of the object in the robot frame, x co-ordinate of the object in the image frame and laser sensor data are used. As view angle of the camera and size of the image are known, direct relationship between the x coordinate of the object in the image frame and angle of the object w.r.t. the robot can be established as:

$$\alpha = -\frac{\phi}{2} + \frac{x}{x_{max}}\phi \tag{12}$$

α is the angle of the of the object w.r.t. the robot at a particular x coordinate of the object in the image frame. In our case, camera view angle is $40°$, and size of image is 640×480.

Once the angle of the object w.r.t. the robot is obtained, the distance of the object can easily be measured with the help of the laser data. Laser can give the distance reading from $-90°$ to $+90°$ with a resolution of $0.5°$. The nearest integer value of the angle α is choosen and the minimum of the three laser reading at $\alpha - 1$,α and $\alpha + 1$ is taken. As the object is in motion, so it may happen that at a particular angle α, the laser ray may not come back from the object. So, $1°$ offset is put to get the accurate reading. Using this transformation, the polar coordinates (α, r) of the gravity center of the object in the robot frame are obtained.

V. Description of particle swarm optimization and T-S fuzzy model

A. Particle swarm optimization

PSO is an optimization technique developed by Kennedy and Eberhart [9]. It is inspired by the formation of swarms by animals such as bird flocking and fish schooling. The principle behind PSO is that each individual in the swarm, called a particle, will move towards the best performing particle in the swarm while exploring the best experience each particle has [12]. The particle update their velocities as follows [13]:

$$\begin{aligned} v(k+1) &= \eta(k+1).v(k) \\ &\quad +c_1(k+1).r_1.(P^{lbest}(k) - P(k)) \\ &\quad + c_2(k+1).r_1.(P^{gbest}(k) - P(k)) \end{aligned} \tag{13}$$

where, k is the generation number, v denote the particle velocities, η denotes the inertia weight, $r_1$and r_2 are random numbers between 0 and 1, c_1 is the cognitive parameter, c_2 is the social parameter, P^{lbest} is the local best solution and P^{gbest} is the global best solution of the group.

The inertia weight η represents the degree of momentum of the particles. This parameter is used for balancing between local and global explorations. In early generations, it is set higher, so that the particles are allowed to have much exploration capability and pursue an aggressive search of the solution space. Once the algorithm is found to converge towards the optimum, this coarse tuning is gradually converted to finer tuning by making η smaller in later generations. In this paper, a linearly adaptable inertia weight is employed [14], which starts with a high value η_{max} and linearly decreases to η_{min} at the maximum number of generations. This means that $\eta(k+1)$ is calculated from

$$\eta(k+1) = \eta_{max} - \frac{\eta_{max} - \eta_{min}}{Gen_{max}}.Gen \tag{14}$$

where, Gen_{max} is the maximum number of generations and Gen is the current generation number.

The constants c_1 and c_2 represent the weights of the stochastic acceleration terms that pull each particle toward the local best and global best positions. With a large cognitive component and small social component at the beginning, particles are allowed to move around the search space, instead of moving toward the best solution. In the latter part of the optimization, a small cognitive component and large social component are used, to allow the particles to converge on the global optima. In this paper, we use linearly time-varying acceleration coefficients over the evolutionary procedure. Therefore, the acceleration coefficients $c_1(k+1)$ and $c_2(k+1)$ can be expressed as follows [15]:

$$c_1(k+1) = c_{1max} - \frac{c_{1max} - c_{1min}}{Gen_{max}}.Gen \tag{15}$$

$$c_2(k+1) = c_{2min} - \frac{c_{2min} - c_{min}}{Gen_{max}}.Gen \tag{16}$$

To limit the searching space v is limited to be within a certain range of $v_{min} \leq V \leq v_{max}$.

The particle positions is updated as:

$$P(k+1) = P(k) + v(k+1) \tag{17}$$

Where, P is the positions of the particle. P should also be limited to be within a certain range of $P_{min} \leq P \leq P_{max}$ for limiting the searching space.

The evaluation of the particle performance is based on a problem specific fitness function that decides the 'closeness' of the particle to the optimal solution. The particle which has the best fitness in any generation till the current generation is known as global best particle and its position is known as global best solution (P^{gbest}). For each particle, there is a local best solution (P^{lbest}), which is the position of the particle at generation g in which that particle has the best fitness till the current generation.

B. The T-S Fuzzy Model

The T-S fuzzy model constructs a map from input space to output space through a fuzzy average of local models. The local models can be either linear or nonlinear. In this paper, the map is built using local linear models. The i^{th} rule in a T-S fuzzy model with k inputs has the following form:

$$R^i : IF \ x_1 \ is \ A_1^i \ and \ x_2 \ is \ A_2^i \ ....x_k \ is \ A_k^i$$
$$THEN \qquad y_i = w_i^T x + b_i$$

where, R^i is the i^{th} rule (i = 1,2....m); m-is the number of rules; x =[x_1,x_2,.....x_k]^T is a input vector; $w_i \ \epsilon \ \mathbf{R}^{1\times k}$; b_i is the constant; A_1^i,A_2^i,....,A_k^i are fuzzy sets and y_i is the consequence of the i^{th} rule.

The possibility that the i^{th} rule will fire is given by the minimum of all the membership functions associated with the i^{th} rule (Mamdani's implication [16]).

$$\mu_i = \min(\mu_1^i, \mu_2^i, ...\mu_k^i) \tag{18}$$

where, μ_i is the membership value for the i^{th} rule and μ_k^i is the membership value of the k^{th} input in the A_k^i fuzzy set. The weighted membership value for the i^{th} rule is given by:

$$\sigma_i = \frac{\mu_i}{\Sigma_{i=1}^k \mu_i} \tag{19}$$

By using center of gravity method for defuzzification, the overall output of the T-S fuzzy system is given by:

$$y = \sum_{i=1}^{k} \sigma_i * y_i \tag{20}$$

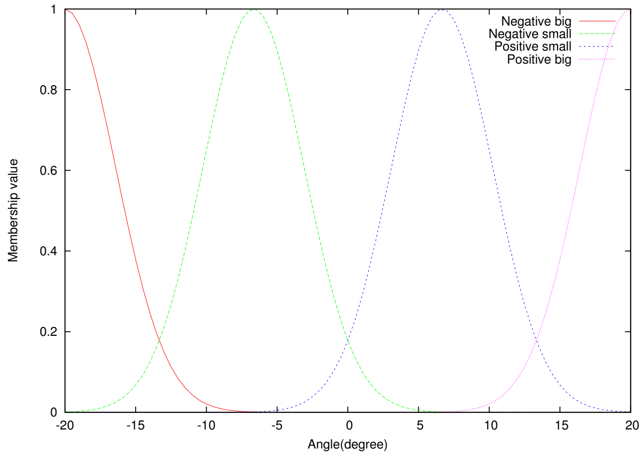

Fig. 2. Fuzzy zones for angle α

VI. Modelling of the T-S fuzzy controller for object tracking

Modelling of the T-S fuzzy controller required the followings steps:

1) Acquisition of real-time data for training and testing the T-S fuzzy model.
2) Design and develop the T-S fuzzy model using training data.
   - Identification of the parameters of the premise part of the rules.
   - Identification of the parameters of the consequent part of the rules.
3) Verification of the designed T-S fuzzy controller to demonstrate the desired object tracking behavior on the Pioneer robot.

A. Acquisition of real-time data for training and testing

To collect the training and testing data, the robot had driven manually using a joystick to set both translational and rotational velocities, guiding the robot to follow the moving red box. The human driver had no visual contact with the object and the robot, he used the robot's camera video stream and his sensor motor coordination to steer the robot towards the box.

The robot was driven in this manner for one hour. During this time, the placement angle of the object w.r.t. the robot and the distance of the object from the robot, and the robot's translational and rotational velocities were logged every $500ms$. The robot's maximum translational velocity is set to 750 mm/sec.The object position w.r.t. the robot was estimated with the hybrid CAMShift algorithm and laser data, as discussed in section II and IV.

B. Design of T-S fuzzy controller

For object tracking behavior the inputs to the T-S fuzzy model are the object position in the robot frame, i.e., object placement angle w.r.t. the robot and the object distance from

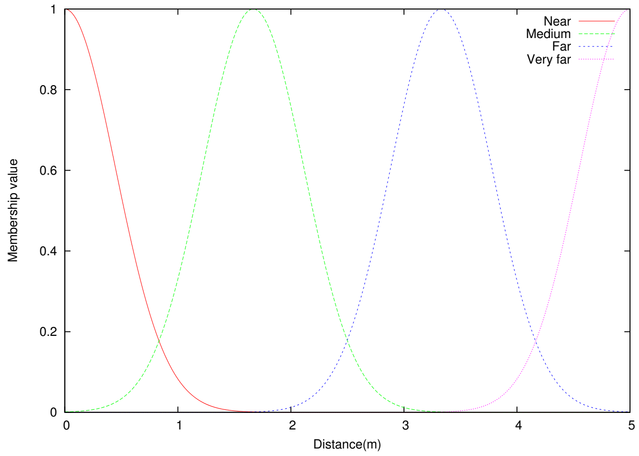

Fig. 3. Fuzzy zones for distance d

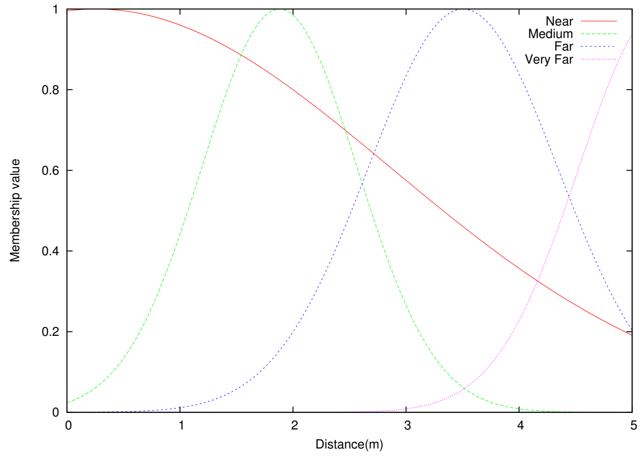

Fig. 4. Fuzzy zones for distance d

the robot, and the current translational and rotational velocities ($V(t)$,$\omega(t)$) of the robot. Outputs of the T-S fuzzy model are desired translational and rotational velocities ($V(t+1)$,$\omega(t+1)$) of the robot required to reach the object.

Placement angle (α) and distance (d) of the object are fuzzified into four zones each as shown in figure 2 and 3 respectively. Initially, a Gaussian membership function is taken for each zone and later on membership functions are updated using particle swarm optimization. The total number of rules (k) is equal to the product of number of zones for α and number of zones for d ,i.e., 4 $\times$4 = 16. For each rule the membership value is calculated as follows:

$$\mu_i = \prod(\mu_{\alpha_i}, \mu_{d_i}) \tag{21}$$

where, μ_{α_i} is the membership value of α in i^{th} rule and μ_{d_i} is the membership value of d in i^{th} rule.

For each rule the outputs are given as:

$$V_i(t+1) = w_i^T x + b_i \tag{22}$$

$$\omega_i(t+1) = w_i^{'T} x + b_i' \tag{23}$$

where, x = [d,α,$V(t)$,$\omega(t)$]^T is a input vector; w_i and w_i' are parameter vectors to be updated to make the model for a given behavior and b_i and b_i' are constants.

The overall outputs of the T-S fuzzy system are given as:

$$V(t+1) = \sum_{i=1}^{k} \sigma_i * V_i(t+1) \tag{24}$$

$$\omega(t+1) = \sum_{i=1}^{k} \sigma_i * \omega_i(t+1) \tag{25}$$

After defining the structure for the T-S fuzzy controller, the parameters of the structure are identified as follows:

1) Identification of the parameters of the premise part of the rules using particle swarm optimization: The membership functions used in the premise part of the T-S fuzzy controller are all of Gaussian forms. The parameters that define the Gaussian membership function are mean m and the deviation σ. The Gaussian membership function is defined as:

$$G_{mf}(x) = e^{-\frac{(x-m)^2}{2\sigma^2}} \tag{26}$$

Since, we have defined 2 inputs each having 4 zones, so there are 8 membership functions and a total of 16 parameters that need to be updated. Therefore, in the PSO, each particle is to have 16 dimensions. We have defined 20 particle in the swarm and total searching iterations has been set to 2000. The minimum and maximum value of inertia weight have been set to be 0.1 and 0.9 respectively. For weights of the stochastic acceleration minimum and maximum value have been set to 0.5 and 2.5 respectively. The fitness function that evaluates the fitness of each particle has been defined as:

$$f(x(k) = \sum_{g=0}^{G_{max}} \epsilon^2 \tag{27}$$

where, $x(k)$ is the k^{th} particle of the swarm, $Gmax$ is the maximum number of generation and ϵ is the output error. At the completion of the all iterations, the membership functions for the inputs of the T-S fuzzy controller have been modified significantly as shown in figure 4 and 5.

2) Identification of parameters of the consequent part of the rules using stochastic method: There are several methods described in the literature for the parameters estimation [17]. Least-mean square algorithm based on the idea of stochastic approximation is widely used. It was developed by Widrow and Hoff [18] and is used for adjusting the weights in a liner adaptive system. For consequent part parameters identification we have used stochastic approximation method, as described by the Jelena in [10]. Once the training is over, the learned T-S fuzzy model with 16 rules is validated using test data. One sample rule of the learned T-S fuzzy model is given here:

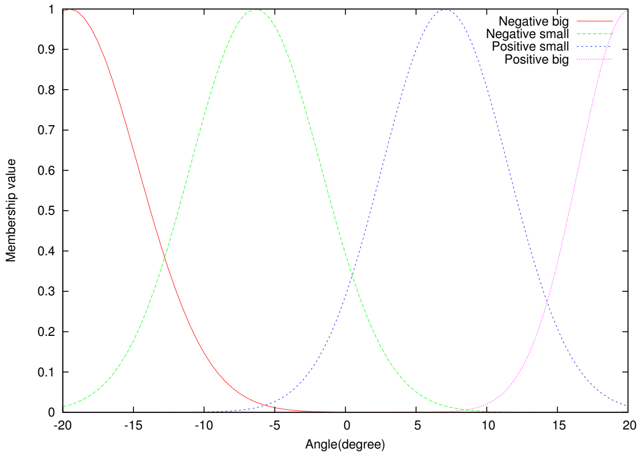

Fig. 5. Fuzzy zones for angle α

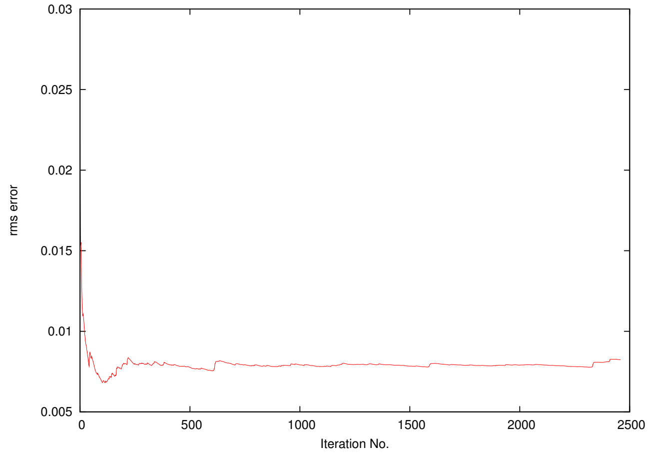

Fig. 6. RMS error during testing

- If α is Negative big and d is Near, then

$$\begin{aligned} V_1(t+1) &= 0.6506d - 0.00009\alpha - 0.00173V(t) \\ &\quad +0.00007\omega(t) - 0.0169 \\ \omega_1(t+1) &= 0.0046d + 1.37\alpha - 0.0295V(t) \\ &\quad -0.026\omega(t) - 0.173 \end{aligned}$$

C. Experimental results and Observations

We applied the testing data set and observed the error. The RMS error during testing of the model is shown in figure 6. Figure 7 shows the desired and actual translational velocities of the robot for the object tracking behavior during testing. Desired and actual rotational velocities of the robot for the object tracking behavior during testing is shown in figure 8. As compared to our previous work [19], the rms error has reduced significantly. The reason of error reduction is the updation of the parameters of the membership functions in this work. Searching mode for the robot, which is a slow process is not required in this case, as the grey predictor predict the position of the object quite efficiently, when the algorithm fails to detect the object. A test run of the designed model in action can be seen in the following video [20].

VII. Conclusion

This paper has presented a T-S fuzzy model based sensor-motor coordination scheme for object tracking. The object is detected using vision sensor and the laser is used to transform the image coordinates of the object into the robot frame coordinates. Particle swarm optimization is used to optimized the parameters of membership functions of the T-S fuzzy model. Since the robot behavior is expressed as if-then rules, the behavior modelling can be easily interpreted. A gray predictor is developed to predict the position of object, when it is not in the view of the robot. The control scheme has been implemented on a Pioneer robot for tracking a triangular box. The experimental result shows that the robot is able to track any object in any arbitrary trajectory using the proposed controller.

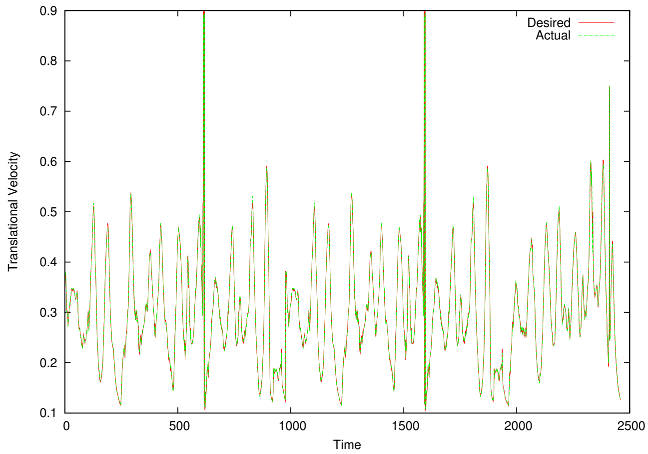

Fig. 7. Desired and actual Translational Velocity

References

[1] R. Siegwart and I. Nourbakhsh, *Introduction to Autonomus Mobile Robots*. The MIT Press, Massachusetts Institute of Technology Cambridge, 2004.

[2] A. Ohya, A. Kosaka, and A. Kak, "Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing," *IEEE Transcations on Robotics and Automation*, vol. 14, pp. 969–978, December 1998.

[3] D. S. Kumar, "Vision-based robot navigation using an online visual experience," Master's thesis, Center for Visual Information Technology, International Institute of Information Technology, Hyderabad, India, June 2007.

[4] V. Ramesh, D. Comanciu, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2000, pp. 142–149.

[5] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal*, 1998, 2nd Quarter.

[6] M. Gupta, B. Uggirala, and L. Behera, "Visual navigation of a mobile robot in a cluttered environment," in *Proceedings of the 17th International Federation of Automatic Control World Congress*, July 2008, pp. 14 816–14 821.

[7] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[8] M. Sugeno, "Structure identification of fuzzy model," *Fuzzy Sets and Systems*, vol. 28, pp. 15–33, 1988.

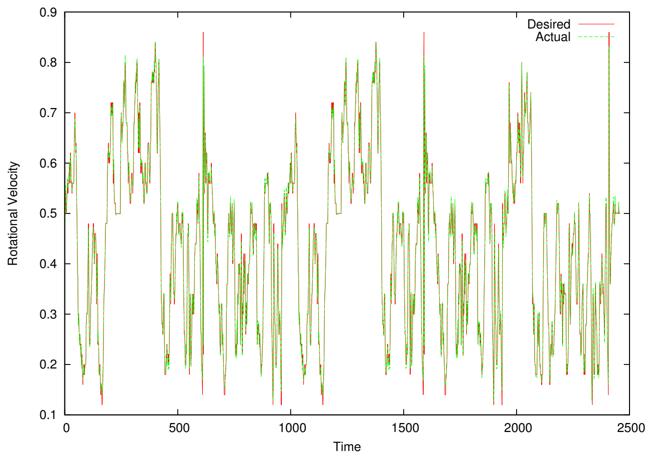

Fig. 8. Desired and actual Rotational Velocity

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE Internatinal Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[10] J. Godjevac and I. f Ecublens, "A learning procedure for a fuzzy system: application to obstacle avoidance," in *Proceedings of the International Symposium on Fuzzy logic*, 1995, pp. 142–148.

[11] D. Julong, *Grey Forecasting Control, In Grey System*. China Ocean Press, 1988.

[12] G. Fang, N. M. Kwok, and Q. Ha, "Automatic fuzzy membership function tuning using the particle swarm optimisation," in *Pacific-Asia Workshop on Computational Intelligence and Industrial Application*. IEEE, 2008, pp. 324–328.

[13] C. N. Ko and C. J. Wu, "A pso-tuning method for design of fuzzy pid controllers," *Journal of Vibration and Control*, December 2007.

[14] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1998, pp. 611–616.

[15] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[16] T. J. Ross, *Fuzzy logic with engineering applications*. McGraw-Hill, International Editions, 1995.

[17] K. J. Astrom and P. Eykhoff, "System identification - a survey," in *Aotomatica*, 1971, pp. 123–168.

[18] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*, A. S. Willsky, Ed. The MIT Press Signal Processing, Optimization, and Control Series, 1983.

[19] M. Gupta, T. N. Kumar, L. Behera, K. S. Venkatesh, and A. Dutta, "Environment modelling in mobile robotics through takagi-sugeno fuzzy model," in *Irish Signal and System Conference*, June 2009.

[20] M. Gupta, "Object tracking," http://www.youtube.com/watch?v=HICmUFJRgyY.