IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 6, Ver. V (Nov – Dec. 2015), PP 59-63
www.iosrjournals.org

Design of Mobile Robot Navigation system using SLAM and Adaptive Tracking Controller with Particle Swarm Optimization for Indoor Environment Monitoring

Kapil Jajulwar[1] Dr.Amol Deshmukh[2]
[1](Research Scholar, Electronic Engineering G.H.Raisoni College of Engineering, India)
[2](Deputy Director & Professor, Electronic Engineering G.H.Raisoni College of Engineering, India)

Abstract: *Now a day mobile robot work as a autonomous machine which is used to control the locomotion and move around in environment by varying the physical location A self localization & mapping of mobile robot using fuzzy based tracking controller is design . The localization of mobile robot is calculated by slam technique. A AFTC with 9 various rule based is used to calculate the total unknown path of mobile robot in indoor monitoring system is calculate & the PSO is used to optimized the total path required for same. A system design with SLAM technique, Adaptive Fuzzy based controller & PSO which is used to solve the self localization of mobile robot.*

Keywords: *Fuzzy Controller, Particle Swarm Optimization, Simultaneous Location & mapping technique*

I. Introduction

The Navigation is an area of study that concentrates on the process of detecting and controlling the motion of a vehicle from one place to another. The area of navigation includes the four basics categories such as aeronautic navigation, marine navigation, space Navigation and land navigation. Locating the Navigator's position compared to known locations or patterns includes in navigational techniques.]Robot navigation means the robot's talent to figure out its own position in its reference frame and design a path for the goal location. Navigation is basically consisting of the three basic aptitudes: Self-localization, Path planning, Map-construction. Self-localization indicates the robot's capacity to confirm its own position and orientation within the reference frame. Path planning is efficiently denotation of localization, in that it requires the determination of the robot's present position and a position of a goal location, both within the same coordinates. Map construction can be in the status of a metric map or any symbols describing locations in the robot frame of reference. As long as we see Robot navigation has been a ground laying goal in both computer vision research and robotics. While the problem is largely solved for robots furnished with an active range-finding device, for a variety of reasons.Vision is an attractive sensor as it helps in the design of economically possible systems with simpler sensor limitations. Different sensors are used in conventional robot navigation systems, utilizing traditional sensor there are several drawbacks related to physical limitations of the sensor. Vision sensing has come forth as a popular choice where cameras can be used to minimize the overall cost and maintaining high degree of intelligence, robustness and flexibility. Robot navigation classified in to two parts:[5]5 Indoor navigation Outdoor navigation Indoor navigation is more complex as against the outdoor navigation. Because, indoor navigation has to deal with more dimensions (multiple floors of building), needs more level of detail to understand enclosing area around navigating subject or object, and has to consider the context of navigation for accurate sub spacing results. Keeping in view the complexity of indoor navigation we have to find solutions for applications (e.g., route planning) of indoor navigation that must take into consideration of the context of navigation and be more level of detail to have accurate and consistent results for users.The proposed methodology will use the color image acquired by the single camera module, which is stored in database. Then the image is converted in to gray image by image processing technique. And using Adaptive fuzzy tracking controller the reference path for the navigation of robot will be calculated. The Procedure to reach the Destination is Time Consuming.[2] Hence the Reseach work is to reduce the time Consumption using PSO. Directing the robot to the Destination is another major Task to be fullfilled as the path would consist of several Obstacle ,consider images stored in Database.[3] One of the image is consider from the Database. It Is Difficult to Controller work on RGB level. Therefore it is needed to convert in gray level image.

Then that gray image is fed to controller. It will calculate shortest path and Time required. Whatever time is needed to complete this simulation is reduce by applying PSO Technique. Working of Adaptive Tracking Fuzzy Controller (ATFC) and Particle Swarm Optimization (PSO) is discussed below.

Motion of Robot		Indication of Obstacle From Sensor		
Sr. No	Robot Decision	Sensor Right	Sensor Straight	Sensor Left
1	Straight	A	A	A
2	Easy Right	A	A	N
		A	N	A
		A	N	N
3	Strong Right	A	F	NS
		A	N	NS
4	Easy Left	N	A	A
		N	N	A
5	Strong Left	NS	A	A

Table No.1: Rule Used For Sensor Devices

II. Design Approach

Controllers which are based upon Fuzzy logic are adept control systems that fluently insinuate between rules. Rules fire to continuous degrees and the multiple resultant actions are mixed into an insinuate result. Processing of uncertain information and savings of energy using common-sense rules and natural language statements are the bases for fuzzy logic control. Fuzzy controller rule-bases typically consist of form of a set of if-then rules whose preceding ("if" parts) and following ("then" parts) are proportionally involve fuzzy membership functions. If X is input and Y is output assumed in a given discussion of a fuzzy controller with a rule-base of size n, then 'if-then' rule as follows:

IF x is $\tilde{A}_i$ THEN y is $\tilde{B}_i$ Where, x and y are respective input and output fuzzy real variables, and $\tilde{A}_i$ €X and $\tilde{B}_i$ €Y $(1 \leq i \leq n)$ are fuzzy sets representing exact values of x and y.A fuzzy controller or model uses fuzzy rules, which are linguistic if-then statements involving fuzzy sets, fuzzy logic, and fuzzy inference. Fuzzy rules play a key role in representing expert control/modeling knowledge and experience and in linking the input variables of fuzzy controllers/models to output variable (or variables). Two major types of fuzzy rules exist, namely, Mamdani fuzzy rules and Takagi-Sugeno (TS, for short) fuzzy rules. Let's start with the familiar Mamdani fuzzy systems. Out of these two here we are using Takagi-Sugeno for monitoring the Working of controller as shown in below.

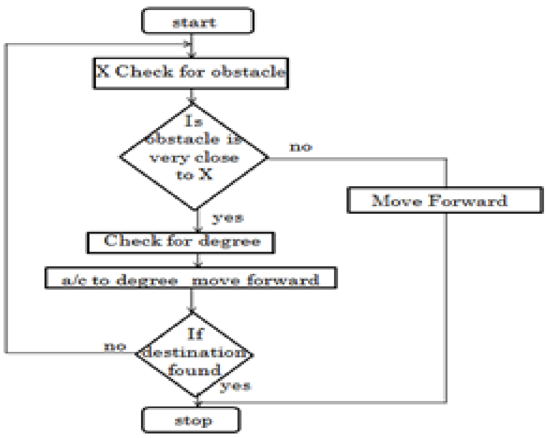


Figure No -1 Flow Chart of Adaptive fuzzy controllers

The fuzzy controller here Where; A-Away, N-Near, NS-Nearest Robot is considered as X.Our purpose is to design a FLC able to guide a robot in its navigation in an environment from a start position to a goal zone. So for this reason we adopted the following fuzzy logic system: In the Fuzzification, the inputs variables of our FLC are dl, dc, dr, dLr, dLl and _; where: dl, dc and dr: are respectively the distances to the nearest obstacles in the left, the center and the right sides of the robot, dLr, dLl: are respectively the distances to the nearest obstacles in the lateral right and lateral left sides of the robot and a: is the angle value between the robot direction and the target position.

The variation range of variables dl, dc and dr is [0, 10] in centimeters; 0 cm means that the obstacle is very near and a collision may be happen, 10 cm means that the obstacle is very far. Each of those three variables are represented by two trapezoidal Mfs: Near and Far. Whereas values of dLr and dLl are either 0 or 1 depending on the presence of obstacle or not. While the variation range of the input variable a is [-180^0, 180^0] in which negative values refer to the left side robot and positive ones to the right side.

Concerning the output variable of our FFLC we defined the angular velocity computed from the left and right robot's angular velocities values Vangular; the robot is supposed to have a fixed linear velocity equal to 100 mm/s. Regarding the inference part, in our controller we elaborated rules in such a way the robot behavior mimic the human comportment in driving.

III. Particle Swarm Optimization

Concept of Particle Swarm Optimization (PSO) was given by Eberhart and Kennedy in 1995. This technique was inspirited by bird flocking's social behavior, where it was observed that the motion of each bird is influenced by the motion of the nearest bird to it. PSO was used successfully in several applications. The algorithm initializes the population with random solutions and updates the generation to found optima. N number of particles exists in a swarm in M-dimensional search space. In PSO each individual or particle has a position and a velocity expressed by a position change which means the flying direction of the particle. It successively adjusts its position based on two factors: the best position of its neighbor (pbest) and the best position swing by the entire swarm (gbest). The best solution is determined using a defined fitness function. Each particle is represented by its instantaneous current position x and position change _(x) or also named velocity, which is generally calculated as below:v[] = c0 *v[] + c1 * rand() * (pbest[] - present[])+ c2 * rand() * (gbest[] –present)&present[] = present[] + v[]Where w indicates the inertia weight, t is the t^{th} iteration, rand() is a random function in [0 1], gbest is the global best of the whole swarm, $pbest_i$ is the local best of the i^{th} particle and fc1; c2g are cognitive and social parameters.

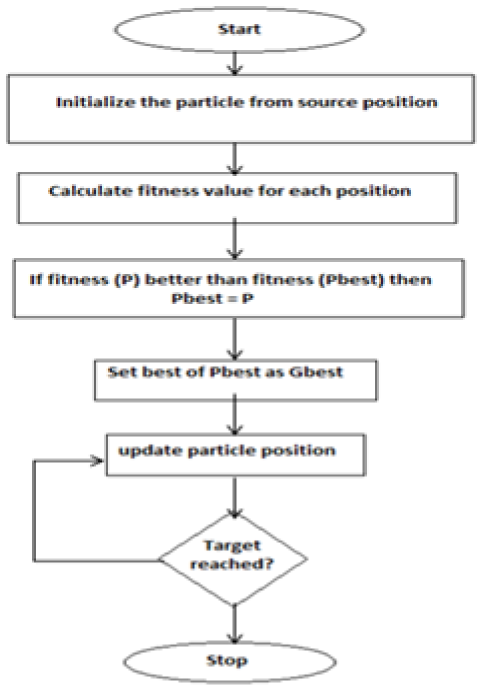


Figure :-2 Flow chart of particle swam optimization

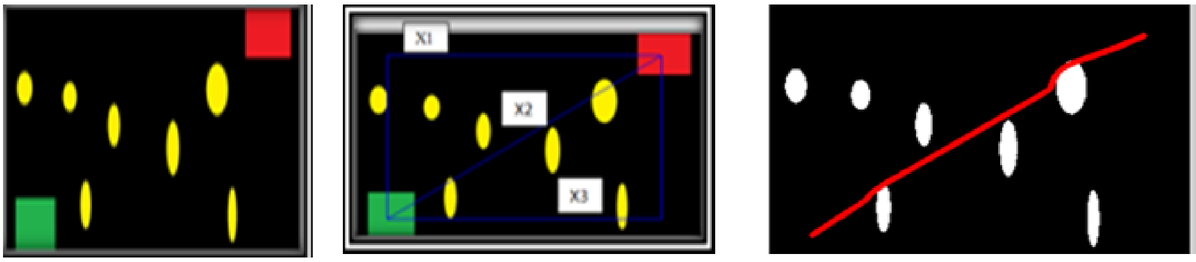
The swarm size is assumed to be D. Fig.4 illustrates the schematic view of updating position of particle in two successive iteration. Working of PSO explained in following flowcharts algorithms using population topologies, which can be smaller and localized subsets of the global best value, are familiar in these days. Neighboring particle may involve two or more particles which are predetermined to act together. Figure: Single-located, where particular particle only compare themselves to the next best.

IV. Simulation Results

Design of Adaptive Tracking Controller Using Particle Swarm Optimization (PSO) for Robot Navigation gives results as describes fallow. The starting direction is found from the start point and ends at the goal point. The inputs are three sensors for detection of obstacle which fuzzified on the basis of far, close or very close and the output is the navigation i.e. straight, soft right, hard right, soft left, hard left.

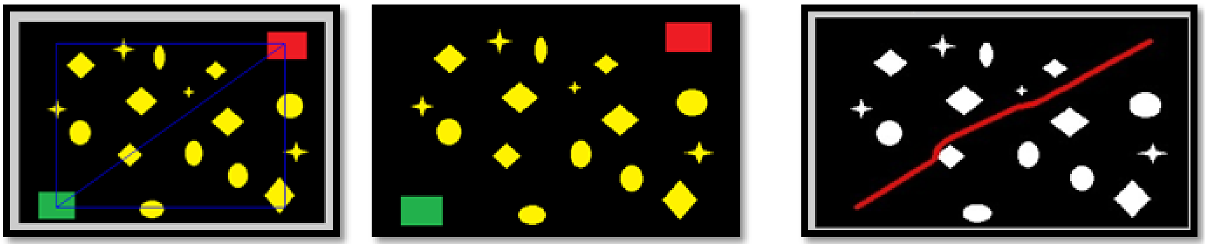
The distance of the obstacle is sensed from the sensor and depends on the current position and orientation of the moving robot. Figure 1, 2 consist of source goal along with obstacle, Figure Shows possible path here only three possible path considered out of which one is shortest one shown in Figure 2and 4. Table 2,

5 shows the Possible Path and Path with No. of obstacle. Whereas table2 Show Movement of Robot and Obstacle Position Here Distance between Source and Destination considered 10 units then it is divided into 3 parts according to the Ranges As for nearest, near and away.
Three sensors are considered in Left, Right and Straight direction respectively. Depending upon the distance and direction membership function has been calculated. Simulation Results by using different Enlivenments



(a) **(b)** **(c)**

Figure 3- Simulation result based on Enlivenment No-01 (a) Input image for indoor Environment (b) Input with no of possible path in Environment No -1 (c) Calculated Navigated Path using PSO in Environment No-1



(e) **(f)** **(g)**

Figure 4 - Calculated navigated path with avoiding the no. of obstacle & Target in Input Environment-2 (e) The no. of projected path in ENV. No. 2 (f)Input image for Indoor room of ENV No.-2 (g) Calculated Navigated Path using PSO.in ENV. No.2

V. Mobile Robot Navigation Using Pso

In this paper the navigation path was calculated for mobile robot in different indoor environment ,here two simulation result are calculated using two different environment with maximum obstacle & target. The Adaptive fuzzy tracking controller was design using 09 fuzzy rules based & Simulation location & mapping (SLAM) technique are used to find the mobile robot location with path planning & mapping. The Particle swarm optimization was used to optimized the time required in project path & Navigated path.

Navigation Time for Environment 1	
Navigated Path without PSO	11.14sec
Navigated Path With PSO	1.71 sec
Navigation Time for Environment 2	
Navigated Path without PSO	8.92 sec
Navigated Path With PSO	1.55 sec

Table No. 2 – Time Calculation for Mobile robot navigation in Environment 1 & 2.

VI. Conclusion

The adaptive fuzzy Tracking controller finds out the possible collision free and smooth navigation path. In the simulation Particle swarm optimization approaches have been implemented for robot navigation in indoor environments. Navigation Path for mobile robot calculate by Fuzzy tracking controller and particle swarm optimization (PSO). The particle swarm optimization approach is used to calculate the total time for navigated path and optimize path. In optimize path time delay with and without PSO is calculated. Efficiency of adaptive tracking fuzzy controller is increased by using particle swarm optimization. The SLAM technique is used to calculate the total localization & Mapping of the mobile robot in indoor monitoring

References

[1]. S. Rathinam; P. Almeida; Z. Kim; S. Jackson; A. Tinka; W. Grossman and R. Sengupta ," Autonomous Searching and Tracking of a River using UAV", Proceedings of the American control conference, new York city ,USA, July 11-13,2012.

[2]. D Santosh, SupreethAchar, C V Jawahar "Autonomous Image-Based Exploration for Mobil Robot Navigation", IEEE International Conference on Robotics and Automation Pasadena, CA, USA, May 19-23, 2013.

[3]. D. Wolf; C. Netto ,"Vision based outdoor Navigation Using Mobile Robots" IEEE Latin American Robotic symposium,pp35-39.
[4]. L. F. Posada; K.K. Narayan; F. Hoffmann and T. Bertram ,"Floor Segmentation of Omnidirectional Images for Mobile Robot Visual Navigation", IEEE/RSJ International Connefernce on Intelligent Robots and systems, Taipei,Taiwan .
[5]. Shankardas ,Deepti Bharat, Abdul Imran Rasheed , Viswanath K. Reddy.Dept. of EEE, MSRSAS, Bangalore, India," Design and ASIC Implementation of Image Segmentation Algorithm for Autonomous MAV Navigation", International Conference on Image Information Processing (ICIIP-2043).
[6]. Ricardo Neves, Anibal Matos, "Raspberry PI Based Stereo Vision for Small Size ASVs" IEEE, International Conference on Image Processing, 2013.
[7]. Chin-kai chang christian siagian laurent itti. ,"Mobile robot monocular vision navigation based on road region and boundary estimation" 2012 IEEE/RSJ international conference on intelligent robots and systems, october 7-12, 2012. Vilamoura, algarve, portugal.
[8]. Denis F. Wolf, CeiloNetto, "Vision-based outdoor navigation using mobile robots", IEEE Latin American Robotic Symposium, 2014.