IORD Journal of Science & Technology
E-ISSN: 2348-0831
Volume 3, Issue 3 (MAR-APRIL 2017) PP 56-63
IMPACT FACTOR 1.719
www.iord.in

"Research work on put into excute for path planning using particle swarm optimization PSO "

Miss.Chetna Ashtakar
Department of Computer Science and Engineering
Vidarbha Institute of Technology
Nagpur, Maharashtra,India

Prof.Pravin Kulukar
Department of Computer Science and Engineering
Vidarbha Institute of Technology
Nagpur, Maharashtra, India

***Abstract*—**The goal of my paper providing an algorithim for path planning to a target for mobile robot in unknown environment. The proposed algorithm allows a mobile robot to navigate through static obstacles, and finding the path in order to reach the target without collision. This algorithm provides the robot the possibility to move from the initial position to the final position (target). Now a day's Computer Vision Based System is gaining more importance due to its use in wide area. When developing robot path planning, it is difficult to make a balance between efficiency and practicability. Here an improved particle swarm optimization algorithm is proposed, which integrates nonlinear inertia parameter and PID control strategy to intelligentize the particles. Based on the algorithm, the convergence and steady state of the algorithm is analyzed rigorously and the convergence conditions were given. To verify the efficiency of the proposed algorithm, it is applied to a classical robot path planning problem, and the simulation results demonstrate both the efficiency and the practicability of the proposed method. The algorithm developed here provides a new way for the robot path planning. The proposed research article illustrates the path planning of an intelligent mobile robot in unknown dynamic environment using Particle Swarm Optimization (PSO) technique. The main objective of the problem is to make the robot move from starting position to goal position while avoiding all obstacles and follow the possible shortest path. For solving the path planning problem, a new objective function is developed. By optimizing the developed objective function the global best particle is determined among the swarm. Thereby, the robot moves further towards the global best position. The robot will continuously move to these intermediate positions until it reaches the final destination.

Keywords*—(*PSO Algorithim, Path Planning Problem, PSO Parameter control)

1. INTRODUCTION

Robot navigation means the robot's ability to determine its own position in its frame of reference and then to plan a path towards some goal location. Navigation can be defined as the combination of the three fundamental competences:

- Self-localization
- Path planning
- Map-building and map interpretation

Vision-based robot navigation has long been a fundamental goal in both robotics and computer vision research. While the problem is largely solved for robots equipped with active range-finding devices, for a variety of reasons, the task still remains challenging for robots equipped only with vision sensors. Vision is an attractive sensor as it helps in the design of economically viable systems with simpler sensor limitations. The conventional robot navigation systems, utilizing traditional sensors like ultrasonic, IR, GPS, laser sensors etc., suffer several drawbacks related to either the physical limitations of the sensor or incur high cost. Vision sensing has emerged as a popular alternative where cameras can be used to reduce the overall cost, maintaining high degree of intelligence, flexibility and robustness.

IORD Journal of Science & Technology
E-ISSN: 2348-0831
Volume 3, Issue 3 (MAR-APRIL 2017) PP 56-63
IMPACT FACTOR 1.719
www.iord.in

Fig1. Particle Swarm Algorithim Steps

2. Path Planning Algorithim

A path planning task usually takes several input values or parameters: a start position (or start configuration): this is the initial configuration of a device (e.g. robot). a goal position (or goal configuration): this is the desired configuration for the device or robot. path planning module allows handling path planning tasks in 3D-space, and in 2D-space for vehicles with non-holonomic motion constraints. The path planning module does not include motion planning for kinematic chains, which is handled by the motion planning module. A novel path planning algorithm based on Parallel Particle Swarm Optimization (PSO) is proposed in this paper to solve the real-time path planning problem in dynamic multi-agent environment. This paper first describes the advantages of PSO algorithm in real time search problems, i.e. path finding problems. Then considering the development trend of multiprocessors, the parallel PSO (PPSO) was proposed to speed up the search process.

A path planning task usually takes several input values or parameters: a start position (or start configuration): this is the initial configuration of a device (e.g. robot). a goal position (or goal configuration): this is the desired configuration for the device or robot. obstacles: those are the objects that the device (or robot) shouldn't be colliding with, while following a path from the start to the goal configuration.

A path linking the start configuration to the goal configuration can be specified (or restricted to be) in a configuration space with a specific number of dimensions (e.g. the X, Y configuration space). Moreover additional constraints are usually needed that make the task more complicated (e.g. keeping a certain distance threshold to the obstacles, or moving only in one direction).

Several objects (or entities) have to be specified to the path planning module in order to define a path planning object:

a collidable or measurable robot entity: this entity represents the device that should avoid obstacles and is referred to hereafter as robot.

a start dummy: the start dummy represents the initial configuration of the robot. The robot entity should be built on top of the start dummy. Make sure that the center of the robot approximately matches the position of the start dummy.

a goal dummy: the goal dummy represents the desired configuration of the robot. The path planning algorithms will try to move the start dummy towards the goal dummy while avoiding collisions between the robot and obstacle.

a collidable or measurable obstacle entity: this entity represents the obstacles that the robot should avoid. It is referred to hereafter as obstacle.

a path: the path acts as a container for a trajectory calculated by the path planning algorithm. The path will also enable the robot to be moved along the trajectory.

3. Path Planning Problem

Many versions of the path planning problem exist. An exhaustive classification of these problems and of the methods developed to solve them can be found in a survey by Hwang and Ahuja . We choose to illustrate our discussion with a particular case. A robot arm is placed among a set of obstacles. Given an initial and a final position of the robot arm, the problem is to find a set of motions that will lead the robot to move between the two positions without colliding with the obstacles. To drive the robot amidst the obstacles, early methods directly used the 3D CAD models of the robot and of the obstacles to find a solution, i.e., they considered the ``operational 3D space''. In this space, the path planning problem consists of finding the movements of a complex 3D structure (the robot) in a cluttered 3D space. A major advance was to express the problem in another space known as the configuration space, denoted by tex2html_wrap_inline1298 [Lozano-Pérez 1987]. In this space, the position (or configuration) of a robot is completely determined by a single point having n independent parameters as coordinates. The positions that are not physically legal (because of a collision) are represented by particular regions of tex2html_wrap_inline1298 ,and are called tex2html_wrap_inline1322 . In the configuration space, the path planning problem consists of finding a continuous curve (representing a path for a single geometrical point) that

(i) connects the points representing the initial and the final configuration of the robot, and
(ii) does not intersect any tex2html_wrap_inline1322 . This method trades a simplification of the path

planning problem (it searches a path for a single point) against a higher-dimensional search space (the dimension of tex2html_wrap_inline1298 is the number DOF of the robot) and against more complex shapes of obstacles (very simple physical obstacles may result in very complex tex2html_wrap_inline1322).

2. Proposed Work

The proposed methodology will use the color image acquired by the single camera module, which acts as a vision sensor, by using the image processing technique and path planning algorithm the reference path for the navigation of robot will be calculated. The Procedure to reach the Destination is Time Consuming. Hence the Reseach work is to reduce the time Consumption. Directing the robot to the Destination is another major Task to be fullfilled as the path would consist of several Obstacle.

- Design a proposed system to navigate the Robot using image processing .
- To design fuzzy tracking controller for robot localization.
- To Calculate and reduce the time delay using Particle Swarm Optimization Technique.
- The proposed algorithm generated optimal paths for various configuration space and the approach in designing the simulation environments for local path planning are encouraging in development of newer algorithms in path planning.

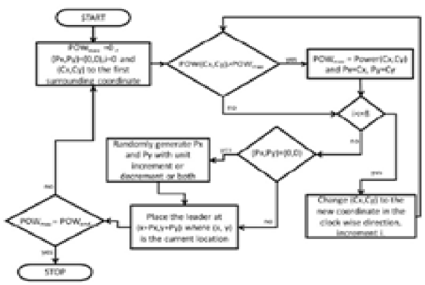


Fig2 : Flowchart for leader path

4. Particle swarm optimization

As stated before, PSO simulates the behaviors of bird flocking. Suppose the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is in each iteration. So what's the best strategy to find the food The effective one is to follow the bird which is nearest to the food.

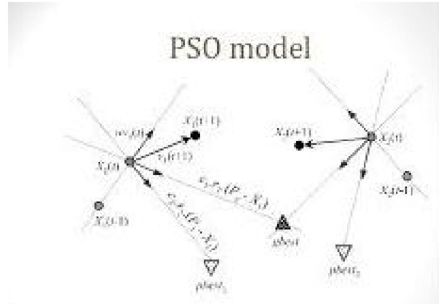


Fig3. PSO MODEL

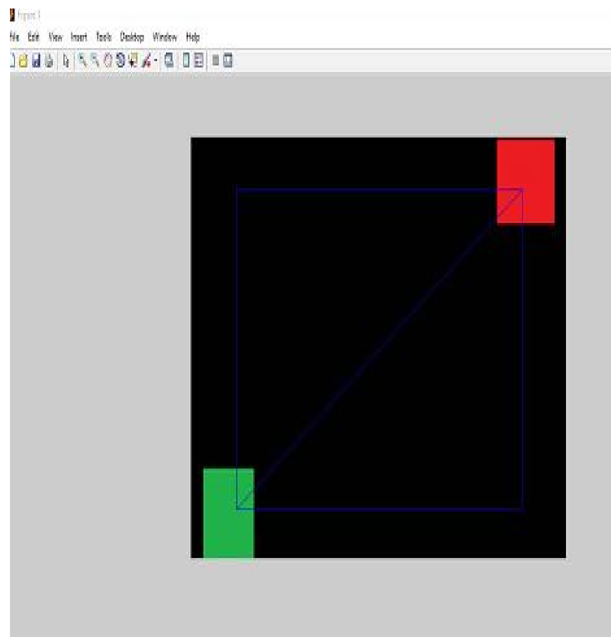


Fig4. PSO Module 1

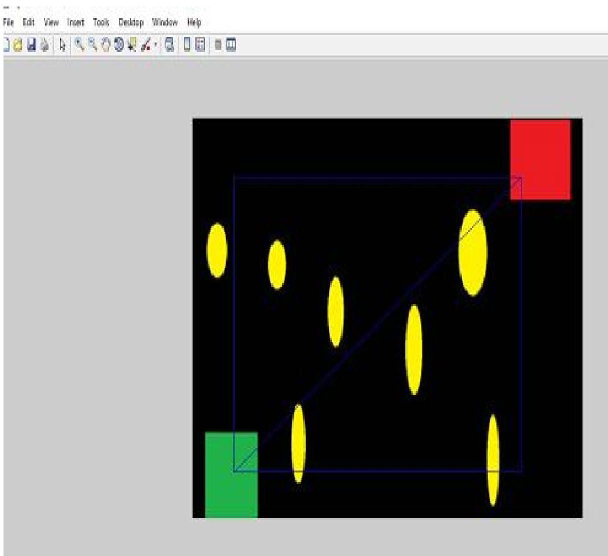


Fig4. PSO Module 2

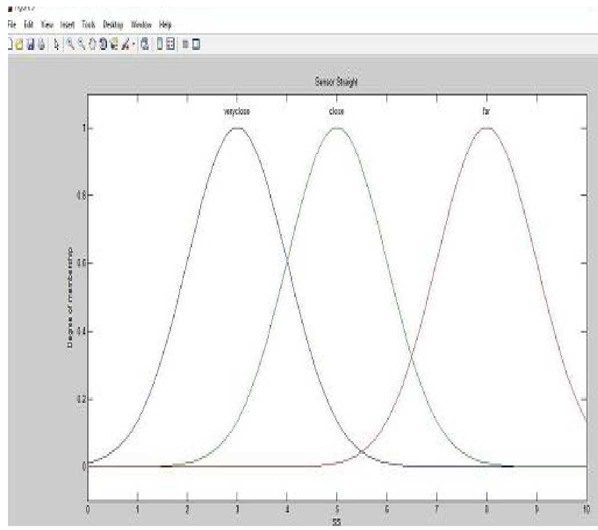


Fig6. PSO Module4

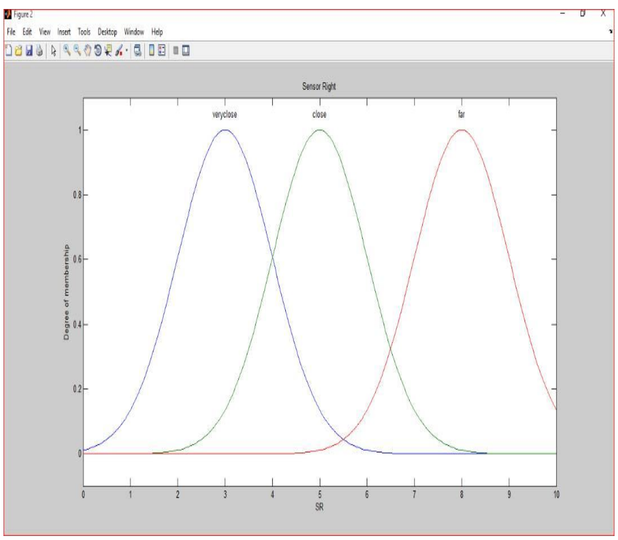


Fig5. PSO Module3

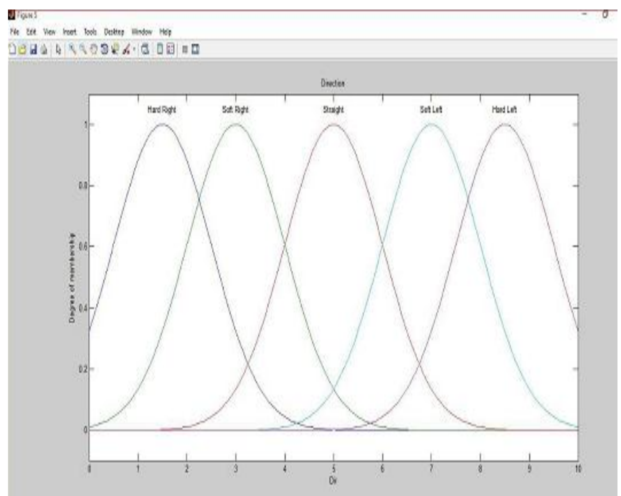


Fig7. PSO Module5

PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. We call it "particle". All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles.

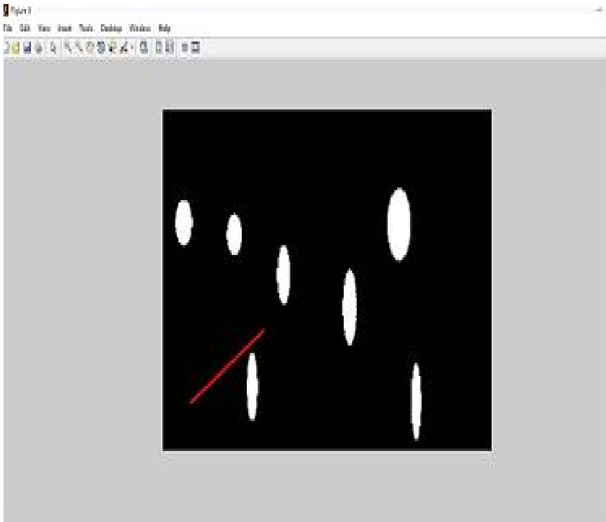


Fig8. PSO Module 6

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest.

After finding the two best values, the particle updates its velocity and positions with following equation (a) and (b).

v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[]) (a)
present[] = persent[] + v[] (b)

v[] is the particle velocity, persent[] is the current particle (solution). pbest[] and gbest[] are defined as stated before. rand () is a random number between (0,1). c1, c2 are learning factors. usually c1 = c2 = 2.

The pseudo code of the procedure is as follows

```
For each particle
    Initialize particle
END

Do
    For each particle
        Calculate fitness value
        If the fitness value is better than the best fitness value (pBest) in history
            set current value as the new pBest
    End

    Choose the particle with the best fitness value of all the particles as the gBest
    For each particle
        Calculate particle velocity according equation (a)
        Update particle position according equation (b)
    End
While maximum iterations or minimum error criteria is not attained
```

Particles' velocities on each dimension are clamped to a maximum velocity Vmax. If the sum of accelerations would cause the velocity on that dimension to exceed Vmax, which is a parameter specified by the user. Then the velocity on that dimension is limited to Vmax.

From the procedure, we can learn that PSO shares many common points with GA. Both algorithms start with a group of a randomly generated population, both have fitness values to evaluate the population. Both update the population and search for the optimium with random techniques. Both systems do not guarantee success.

However, PSO does not have genetic operators like crossover and mutation. Particles update themselves with the internal velocity. They also have memory, which is important to the algorithm. Compared with genetic algorithms (GAs), the information sharing mechanism in PSO is significantly different. In PSO, only gBest (or lBest) gives out the information to others. It is a one -way information sharing mechanism. The evolution only looks for the best solution.

5. PSO parameter control

From the above case, we can learn that there are two key steps when applying PSO to optimization problems: the representation of the solution and the fitness function. One of the advantages of PSO is that PSO take real numbers as particles. It is not like GA, which needs to change to binary encoding, or special genetic operators have to be used. For example, we try to find the solution for f(x) = x1^2 + x2^2+x3^2, the particle can be set as (x1, x2, x3), and fitness function is f(x). Then we can use the standard procedure to find the optimum. The searching is a repeat process, and the stop criteria are that the maximum iteration number is reached or the minimum error condition is satisfied.

There are not many parameter need to be tuned in PSO. Here is a list of the parameters and their typical values.

The number of particles: the typical range is 20 - 40. Actually for most of the problems 10 particles is large enough to get good results. For some difficult or special problems, one can try 100 or 200 particles as well.

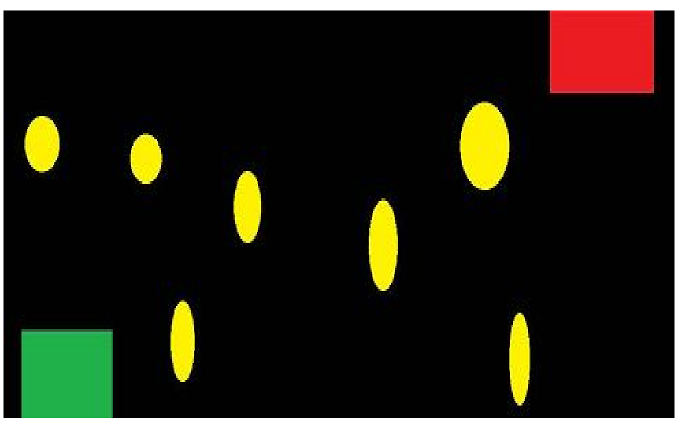
Dimension of particles: It is determined by the problem to be optimized,

Range of particles: It is also determined by the problem to be optimized, you can specify different ranges for different dimension of particles.

Vmax: it determines the maximum change one particle can take during one iteration. Usually we set the range of the particle as the Vmax for example, the particle (x1, x2, x3)
X1 belongs [-10, 10], then Vmax = 20

Learning factors: c1 and c2 usually equal to 2. However, other settings were also used in different papers. But usually c1 equals to c2 and ranges from [0, 4]

The stop condition: the maximum number of iterations the PSO execute and the minimum error requirement. for example, for ANN training in previous section, we can set the minimum error requirement is one mis-classified pattern. the maximum number of iterations is set to 2000. this stop condition depends on the problem to be optimized.



Fi94. PSO Module 7

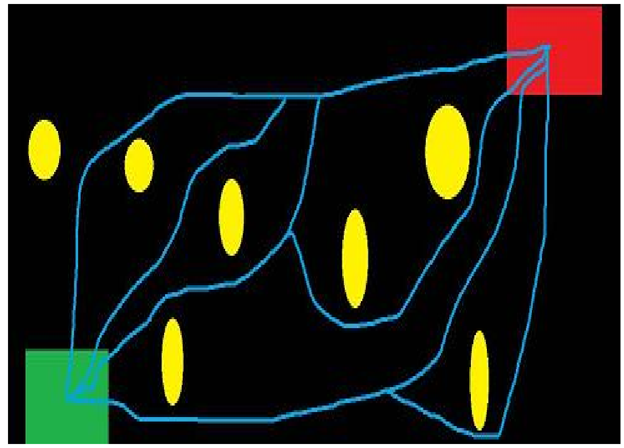


Fig10. PSO Module8

6. LITERATURE REVIEW

6.1:"Single-Objective Path Planning for Autonomous Robots Using Reconfigurable Analog VLSI" , Scott Koziol, Member, IEEE, Richard Wunderlich, Jennifer Hasler, Senior Member, IEEE, and Mike Stilman. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS 2016 IEEE.

Path planning using reconfigurable analog very large scale integrated (AVLSI) circuits. Existing research has shown that custom AVLSI circuits known as application specific integrated circuits (ASICs) can theoretically be used for path planning an algorithm for mapping a

robot's environment onto an FPAA, and then presents hardware results using an FPAA to implement the path-planning algorithm. Experimental results and analysis are presented for 24 environment scenarios. Digital search methods like breadth-first search have solutions which scale on the order of O(4d) whereas this paper will show our analog solution is on the order of O(d) whered is the depth of the solution.

6.2. "A PARTICAL SWARM OPTIMIZATION–Lyapunov Hybrid Stable Adaptive Fuzzy Tracking Control Approach For Vision-Based Robot Navigation", Kaushik Das Sharma, Amitava Chatterjee, And Anjan Rakshit IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 61, NO. 7, JULY 2012

Literature [1] proposes a method for vision based mobile robot navigation which uses stable fuzzy tracking controller design which is based on lyapunov theory and it also uses

PARTICAL SWARM OPTIMIZATION hybrid methodology. Work has been done on two factors:

1. By using single camera based vision sensor which captures images periodically.

2. Secondly it utilizes the method path planning to investigate reference path.

In this literature certain factor were taken in to consideration GA based &PARTICAL SWARM OPTIMIZATION based design approaches and two variants of hybrid design approaches for optimal controller design have been implemented for robot navigation in unknown environments. PARTICAL SWARM OPTIMIZATION based hybrid methodology is considered to be superior approach for this study.

6.3.“Autonomous Mobile Robot System Concept Based On PARTICAL SWARM OPTIMIZATION Path Planner and vSLAM", Abdullah Zawawi MOHAMED &Sang Heon LEE, Hung Yao HSU Department of Advanced Manufacturing and Mechanical Engineering University of South Australia South Australia, Australia

Literature [2] proposes scheme of situating mobile robots to move around in large complex or building for criminal or theft location detection prompt for some time. This study approaches to a mathematical model which helps in development of a scheme for an autonomous low cost mobile robot system using visual SLAM (simultaneous localization & mapping) & PARTICAL SWARM OPTIMIZATION planner. The study could provide a cost effective solution for problem in indoor mobile robot navigation

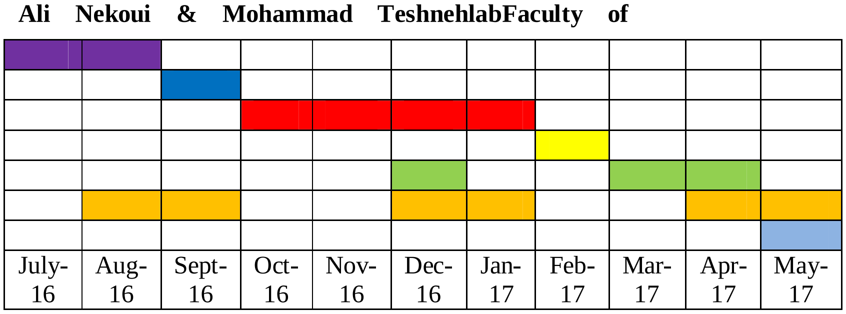
6.4. Adaptive Neuro-Fuzzy Extended Kalman Filtering for Robot Localization, Ramazan Havangi, Mohammad Ali Nekoui & Mohammad TeshnehlabFaculty of Electrical Engineering, K.N. Toosi University of TechnologyTehran, Iran Ramazan Havangi, Mohammad Ali Nekoui and Mohammad TeshnehlabFaculty of Electrical Engineering, K.N. Toosi University of TechnologyTehran, IranIJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 2, March 2010

The EKF (extended Kalman filter) is well known choice for localization when a prior knowledge is well known. Literature [3] proposes develop adaptive Neuro-Fuzzy inference system for localization based EKF when a priori knowledge is incorrect. The propose system aims to start the system with incorrect knowledge of statistics on the basis of adaptive Neuro-Fuzzy inference system which tries to minimize the mis-matching between theoretical and actual values .this method is beneficial because of consistency in this approach is more than localization based EKF

6.5 Qualitative Evaluation of a PID Controller for Autonomous Mobile Robot Navigation Implemented in an FPGA Card Rocio Alba-Flores, Fernando Rios-Gutierrez, and Christopher Jeanniton Dept. of Mechanical and Electrical Engineering

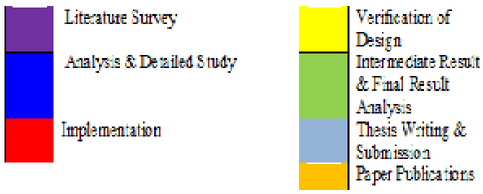
Georgia Southern University, Statesboro, GA, USA 2011 Seventh International Conference on Natural Computation

Literature [4] uses PID controller for autonomous mobile robot Navigation was implemented in PID controller using FPGA card. The PID controller was first simulated in simulink Then FPGA card was used to construct the PID controller in combination with stamp micro controller. Finally, performance of PID controller used to navigate robot..



7.Plan of Research work

Duration in Months



8. CONCLUSIONS

In this paper, Proposed providing an algorithim known as Particle Swarm Optimization. The system is initialized with a population of random solutions and searches for optima by updating generations. The potential solutions, called particles, fly through the problem space by following the current optimum particles. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations.

REFERENCES

[1] “Single-Objective Path Planning for Autonomous Robots Using Reconfigurable Analog VLSI” **,** Scott Koziol, *Member*, *IEEE*, Richard Wunderlich, Jennifer Hasler, *Senior Member*, *IEEE*, and Mike Stilman. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS 2016 IEEE.

[2] Kaushik Das Sharma, Amitava Chatterjee, and Anjan Rakshit, “A PARTICAL SWARM OPTIMIZATION–Lyapunov Hybrid Stable Adaptive Fuzzy Tracking Control Approach for Vision-Based Robot Navigation” IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 61, NO. 7, JULY 2012.

[3] Abdullah Zawawi MOHAMED Sang Heon LEE, Hung Yao HSU Department of Advanced Manufacturing and Mechanical Engineering University of South Australia Autonomous “Mobile Robot System Concept Based On PARTICAL SWARM OPTIMIZATION Path Planner and vSLAM”.

[4] Ramazan Havangi, Mohammad Ali Nekoui and Mohammad Teshnehlab Faculty of Electrical Engineering, K.N. Toosi University of Technology Tehran, Iran “Adaptive Neuro-Fuzzy Extended Kalman Filtering for Robot Localization” IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 2, March 2010

[5] Rocio Alba-Flores, Fernando Rios-Gutierrez, and Christopher Jeanniton Dept. of Mechanical and Electrical Engineering Georgia Southern University, Statesboro, GA, USA Qualitative Evaluation of a PID Controller for “Autonomous Mobile Robot Navigation Implemented in an FPGA Card” 2011 Seventh International Conference on Natural Computation.

[6] N. N. Singh, “Vision based autonomous navigation of mobile robots,” Ph.D. dissertation, Jadavpur University, Kolkata, India, 2010.

[7] C. C. Chou, F. L. Lian, and C. C. Wang, “Characterizing indoor environment for robot navigation using velocity space approach with region analysis and look-ahead verification,” *IEEE* Trans. Instrum. Meas., vol. 60, no. 2, pp. 442–451, Feb 2011.

[8]I. A. R. Ashokaraj, P. M. G. Silson, A. Tsourdos, and B.A. White, “Robust sensor-based navigation for mobile robots,” IEEE Trans. Instrum. Meas., vol. 58, no. 3, pp. 551–556, Mar. 2010.

[9]Jianguo, W., Yilong, Z., Linlin, X., Adaptive Genetic Algorithm Enhancements for Path Planning of Mobile Robots, Proceedings on International Conference on Measuring Technology and Mechatronics Automation 2010, pp. 416-419.