Proceedings of the 13th
Mediterranean Conference on Control and Automation
Limassol, Cyprus, June 27-29, 2005

Experimental Validation of a MATLAB Based Control Architecture for Multiple Robot Outdoor Navigation

L. Doitsidis, A. L. Nelson, K. P. Valavanis, M. T. Long, R. R. Murphy

Abstract— **Design, implementation and experimental validation of a MATLAB based autonomous robot control framework is presented, supported by, and integrated into a distributed field robot architecture known as distributed-SFX. The MATLAB based framework is composed of multi sensor fuzzy logic robot controllers that utilize laser, GPS and odometer data, fusing such sensor data and filtering out noise, to improve collision free navigation. Extensive outdoor environment experiments with single and multiple mobile robots are performed to demonstrate waypoint and goal point navigation, and raster scan search patterns in unknown environments with static and dynamic obstacles. Results and videos are provided to justify the proposed approach.**

I. Introduction

A distributed robot control architecture, both during its design as well as during its implementation and testing phase, brings together concepts and ideas -under a common framework- from the diverse fields of Distributed Artificial Intelligence, Human Robot Interaction and Multi Agent Systems (higher levels/layers), coupled with Control Theoretic approaches (lower level real-time control), operating in unison to provide heterogeneous system functionality, adaptability and flexibility.

This paper focuses mainly on the implementation and experimental validation of the control theoretic *MATLAB* based lower level of the architecture, highlighting briefly for completeness purposes its integration with the overall system architecture, referred to from now on as *distributed-SFX* architecture. Design and implementation details of the MATLAB based framework are discussed; multi sensor fuzzy logic robot controllers are derived and implemented using laser, GPS and odometer data as inputs; details related to fusing such sensor data (laser, GPS and odometer) filtering out noise, are presented; single and multiple robot waypoint navigation, goal point navigation, random and raster scan search patterns along with collision avoidance are also demonstrated experimentally.

A major novel aspect and contribution of this research is related to the fact that although the *distributed-SF*X architecture is implemented in *Java* and uses *Jini* to manage distributed objects and services between robots and computers [39], the *MATLAB* environment is supported within the *Java / Jini* framework using the *JMatLink* library [30] allowing for mathematical control theoretic research and experimentation and for rapid prototyping of behavioral and control modules and services. Wrapping the *MATLAB* workspace environment with *JMatLink*, in conjunction with the *Jini* distributed object platform, allows modules and services implemented as native interpreted *MATLAB* code to be accessed as remote and distributed objects. Although it is the case that many behavioral architectures and languages support implementation of modules in different languages, this approach allows *MATLAB* to be directly incorporated into behavioral architectures resulting in speed up of development and added flexibility of implementation. The combination of *Java* based distributed architectures using *MATLAB* in its interpreted form for autonomous robot navigation and control is not well represented in the literature, and up to now, no complete approach has been published.

The derived fuzzy logic robot controllers utilize three different sensors (lasers, GPS and odometers), they are capable of moving to one or more goal positions regardless of initial position or unpredicted deviations and obstacles along the paths traversed; they may be dynamically redirected to reach a new set of goal or waypoints regardless of the robot current state and location; the robots move continuously without stopping to evaluate terrain or process sensor data and navigate to goal points.

The presented approach differs considerably from related work in [18] and [19]; the reported research in [18] uses dead reckoning requiring a-priori knowledge of initial position, it is not robust against cumulative odometer errors, and at times the robot remains stationary during execution. The approach followed in [19] for outdoor navigation uses small lab-based robots to prototype controllers for large

This work has been partially supported by a grant form ONR, N00014-03-01-786 (with USF 2132-033-LO). L. Doitsidis is supported partially by "HERAKLITOS" fellowships for research from the Technical University of Crete, EPEAEK II – 88727.

L. Doitsidis was with University of South Florida. He is now with Department of Engineering and Management, Technical University of Crete, Greece (e-mail: ldoitsidis@ depm.tuc.gr).

A. L. Nelson, was with the Center for Robot Assisted Search and Rescue, University of South Florida.

K. P. Valavanis, M. T. Long, R. R. Murphy are with the Center for Robot Assisted Search and Rescue, University of South Florida.

0-7803-8936-0/05/$20.00 ©2005 IEEE

outdoor vehicles in agricultural environments; indoor robots use hierarchical fuzzy controllers having a set of different behaviors including obstacle avoidance, goal seeking, and edge following, which are then transferred to outdoor vehicles; however it is applied mainly to corridor and edge following using an IR beacon for homing on goal positions. Further, compared with reported research in [5], [6], [7], the fuzzy controllers introduced here use GPS, odometer and laser data and they are applied in outdoor environments, as opposed to using only sonar sensor data for indoor navigation.

The experimental testbed consists of a fleet of custom made *ATRV-Jr* skid steering differential drive robots equipped with 1 GHz P III processors and 2 Gb of RAM. On board computer includes 10 auxiliary serial ports, *firewire* and *UBS* to accommodate diverse sensors and accessories; robots run *RedHat Linux 9.0*. They are capable of running locally any standard PC software and they are accessible via wireless *Ethernet (802.11g)*. The sensor suite includes a *SICK LMS 200 scanning planar laser* mounted on the front of each vehicle, a *pan/tilt* unit with *FLIR* and *standard video cameras* attached to the front-top equipment rack, *IMU* and *GPS* mounted on the center-top equipment rack. The GPS system is a *Synergy Systems LLC M12+* with evaluation board and a *HAWK GPS antenna* connected via serial link to the robot's main computer. The robots are powered from *2 lead-acid 720 Watt/hour* batteries and can be operated for two to three hours continuously. A custom made *24V power supply box* has been designed and installed in each robot to provide power to added devices.

Looking at the literature, there exist several mobile robot control architectures in the literature, including behavioral robotics architectures [2], [3], [4], [10], [11], [13], [20], [31], [34], [38], [39], designed to support robot control within methodological and conceptual constraints as well as within the behavior-based paradigm. Utilization of *MATLAB* as the primary computing environment for robot control experimentation allowing access to all sensors and actuators in the form of functions that appear as standard *MATLAB* function calls has been reported in [32], [33]. Integration of *MATLAB* and *Java* in engineering systems is the topic addressed in [35], [38].

Handling uncertainty in autonomous robot control is the topic of [1], while methods employing fuzzy logic mobile robot navigation (mostly indoors) include [5], [7], [10], [11], [12], [13], [14], [15], [16], [17], [18] and [19]. Waypoint navigation forcing vehicles to move in predefined environments is reported in [26] and [27]. Outdoor environment navigation using odometer data only has been proven inadequate due to significant cumulative odometer errors [21], [22]; the use of GPS or even other sensors like INS and gyro is essential as shown in [23], [24], [25].

Section 2 presents essential information on the *distributed – SFX* architecture and its interface with *MATLAB*, while Section 3 discusses sensor data processing and robot position detection. Section 4 presents the fuzzy logic robot control approach. Experimental results are the topic of Section 5. Section 6 concludes the paper.

II. Distributed SFX and Matlab

The *Distributed-SFX* architecture [37] is a service-based distributed robot architecture designed to support behavior based control. It is a distributed Java/Jini based descendant of the Sensor Fusion Effects architecture (SFX) [34]. The system makes use of modules to implement robot sensing and control related services. The architecture also supplies high-level service managers for the coordination and functional integration of low level modular services. Modules are exported to a distributed run-time system as services with certain attributes and types. Service may then be searched for (using a distributed-object lookup service) based on functional attributes rather than details of actual implementation or physical location. This type of architecture allows a decoupling of client and server. Clients, in an abstract sense, are interested in services with particular attributes that provide or consume certain types of data.

The JMatLink class library [30] is used to connect Java to *MATLAB*. JMatLink includes methods and objects that allow Java to initialize a workspace, write data members of any format to the workspace, read from the work space, and execute command line functions. *MATLAB* scripts and functions may run locally on the robots as interpreted code without the need to be compiled into stand-alone executables. *MATLAB* is supported at the driver module implementation level and it may be used as the native server implementation of a service. Within Java and Jini, the associated server and proxy handle the remote overhead and interaction with other services [29].

III. Sensor data processing

Sensor data are pre-processed, filtered and fused. In particular, data from the laser scanner are fused with odometer data and are filtered to reduce noise and ghost readings caused by bouncing of the laser unit, by variations in grass and vegetation as well as other unforeseen outdoor environment conditions. Integration of past GPS sensor readings fused with odometer readings results in smoothing of the GPS errors providing more accurate robot position detection; this is accomplished with the aid of a simple heuristic design filter. This section presents details for the laser scan filtering and position detection, both essential to designing and implementing in real-time the robot fuzzy logic controllers.

A. Laser Scan Filtering

The range of the laser scan is 180 degrees, centered on the robot body-attached reference frame in which each consecutive point is offset by 1 degree (181 total points per scan). Laser data is used to identify and avoid static and dynamic obstacles during navigation. Without loss of generality, for all practical purposes offering a trade off between computational complexity and real-time control update cycle, the effective laser scan area is divided in three radial sectors labelled as *Left Area*, *Center Area*, *Right Area*, denoted by W_i $i=1, 2, 3$, each one including further division in *Close*, *Medium* and *Far* regions. as shown in Fig. 1. This division is also used to derive the fuzzy logic robot controllers (discussed in the next Section) and to calculate the corresponding collision possibilities reflecting the presence / absence of obstacles in the robot path. Laser effective range is experimentally verified to be about 8 meters (25 feet). The left and right areas have a width of 70^0 each and the center area of 40^0.

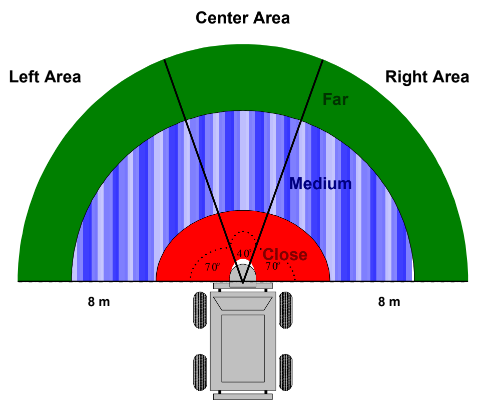

Fig. 1: Laser scan area radial sectors divided into close, medium and far areas

The laser scan filtering process integrates information from the most recent scan at time *k*, with information from up to *n* previous scans, *k*-1, *k*-2, …, *k*-n. Fig. 2.shows two laser scans, one taken at time *k* (the most recent one) and another taken at time *k-n* (a previous one), with α and l the relative angular and linear offsets of the scans at time *k* and time *k-n*, respectively.

Each laser scan is represented as a vector **r** of 181 points in polar form with elements r_i, $i\in\{1,181\}$ representing the Euclidian distance from the origin of the robot body-attached reference frame, with angles implicitly defined by position within **r**:

$$\mathbf{r} = [r_1 \quad r_2 \quad \cdots \quad r_{181}]$$

To integrate information from multiple scans, each of the (previous) past $n\in\{1, N\}$scans is transformed into the reference coordinate frame of the most recent scan $\mathbf{r}(k)$ using a standard rotation and translation transformation **T**, calculated from the time history of robot wheel odometer readings:

$$\mathbf{r}_{\mathbf{T}}(k-n) = \mathbf{r}(k-n)\mathbf{T}$$

A matrix **R** of transformed scans represents the last *n* laser scans with all range readings transformed into the current robot reference frame can then be constructed:

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}(k) \\ \mathbf{r}_{\mathbf{T}}(k-1) \\ \vdots \\ \mathbf{r}_{\mathbf{T}}(k-n) \end{bmatrix}$$

Hence, objects detected in multiple current and past scans appear as range readings of similar values in multiple rows of R.

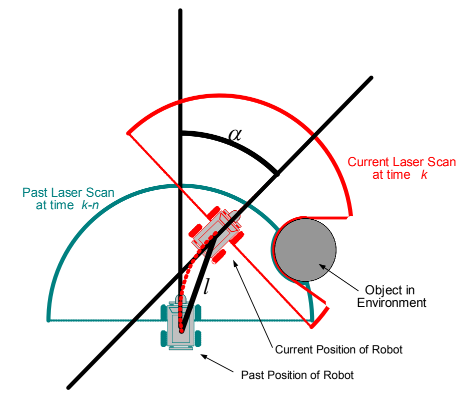

Fig. 2: Laser scans taken at time *k* (most recent scan) and at time *k-n* (a previous scan).

A statistical filter is derived considering the variance, the mean value and the maximum range readings of the elements of each of the columns of **R**. These are given by the vectors of r_{var} r_{μ}, and r_{max} respectively, each of which is of length 181. The final filtered laser vector is given by:

$$\mathbf{r}_{\Phi} = [r_1 \quad r_2 \quad \cdots \quad r_{181}]$$

$$r_i = \begin{cases} r_{i,\mu} & \text{if} \quad r_{i,\text{var}} < v \\ r_{i,\max} & \text{otherwise} \end{cases}$$

$\mathbf{r}_{\Phi}$ is the final filtered set of range readings spanning the forward field of the robot in its current position, and $r_{i,\mu}$, $r_{i,var}$ and $r_{i,max}$ are the *i*th elements of r_{μ}, r_{var} and r_{max}, respectively, and v an appropriately defined threshold value. Elements of scans prior to current time *k* outside the robot's current forward facing field of view (after transformation) are discarded in the above calculations.

B. Position Detection

The robot pose $\mathbf{p}(k)$ at time *k* is represented as a vector of three elements:

$$\mathbf{p}_{pose}(k) = [x_{pose}(k) \quad y_{pose}(k) \quad \theta_{pose}(k)]$$

x_{pose} and y_{pose} are rectangular coordinates given in latitude and longitude in the world coordinate frame, and θ_{pose} is the robot's angular heading with East defined as θ=0 and positive angles measured counter clockwise; k and k-1 represent current and immediately previous time steps.

The pose detection system relies primarily on GPS readings for both location and orientation calculation, using odometer readings for smoothing.

The GPS position $\mathbf{p}_{gps}(k)$ rectangular coordinates (x_{gps} and y_{gps}) are obtained directly from the most recent GPS sensor readings (using the Motorola proprietary filter supplied with the hardware unit), while the heading is obtained differentially using the current coordinate and a previous coordinate. The current robot position based on odometer readings relative to the last known real world position is represented by $\mathbf{p}_{odo}(k)$.

A heuristic algorithm is derived that uses the difference (or error) between $\mathbf{p}_{gps}$ and $\mathbf{p}_{odo}$ when selecting the next position value. This error term is:

$$p_{error} = \left|\mathbf{p}_{gps} - \mathbf{p}_{odo}\right|$$

The algorithm considers an additional aggregate error term e_{agg} when selecting the next position value corresponding to the cumulative worst case aggregate odometer error that may have accumulated over a sequence of controller time steps. The rationale of the algorithm is that if GPS and odometer readings agree within a nominal error threshold, the robot pose at time k is based on GPS; otherwise pose is based incrementally on odometer readings relative to the last known accurate pose. However, if the potential worst case incremental error emax is exceeded, the system is forced to revert to GPS for a time step to maintain localization within the world coordinate frame. This smoothes the GPS noise, but keeps the odometer based path from deviating by more than emax from the "assumed to be" position of the robot.

The algorithm is implemented as follows:

```
If   e_agg > e_max
         p_pose(k) =  p_gps(k)
         e_agg = 0
Else if  p_error < Δe_agg
         p_pose(k) =  p_gps(k)
         e_agg = 0
Else
         p_pose(k) =  p_odo(k)
         e_agg =  + Δe_agg
```

IV. Fuzzy Logic Controller

Given the clarifications for sensor data processing and position detection, the overall structure of the multi sensor control system is shown in Figure 4. It consists of four modules: the laser range filter, position detection, heading error calculation and the actual fuzzy logic robot controller. The control system receives as inputs laser, odometer and GPS data, as well as a control reference input (next waypoint or goal point). It outputs actuator commands in terms of robot rotational and translational velocity.

The fuzzy logic controller is implemented as a Mamdani-type controller similar to previous work [5], [7]. The fuzzy controller input from the filtered laser range block consists of a three value vector with components related to the distance of the closest object in the left sector of the scan, in the center sector and in the right sector, respectively. This information is used to calculate three collision possibilities *left*, *center*, *right* reflecting potential static / dynamic obstacles in the robot field of view, similar to the approach followed in [5], [7], but for outdoor environments. The fourth input to the fuzzy logic controller is the robot's heading error calculated from the robot's current heading and the desired heading.

Implementation wise, each of the three aggregate range inputs includes three trapezoidal membership functions namely, *close*, *medium* and *far*. The input linguistic variables are denoted as *left distance*, *right distance* and *center distance* corresponding the left area, right area and center area sectors. The *heading error* input uses four trapezoidal membership functions and one triangular membership function. Chosen membership functions for the input variables are empirically derived based on extensive tests and experiments.

The value of each distance input variable d_i (corresponding to left area, center area, right area) is fuzzified and expressed by the fuzzy sets C_i, MD_i, A_i referring to *close*, *medium*, and *far* as shown in Figure 2. The range of the membership functions for each d_i is between 0-8 meters. The value of the input variable *heading error, he,* is fuzzified and expressed by the fuzzy sets *FL, L, AH, R, FR*, referring to *far left*, *left*, *ahead*, *right*, and *far right,* respectively. The range of the membership functions for the *heading error,* is between -180 and 180 degrees.

The fuzzy logic controller has two output variables, *translational velocity* (*tr*) implemented with two trapezoidal and one triangular membership functions, and *rotational velocity* (*rv*) implemented with four trapezoidal membership functions and one triangular membership function The value of the output variable *tr* is expressed by the fuzzy sets *ST, SL, F* referring to *stop*, *slow*, and *fast*. The value of the output variable *rv* is expressed by the fuzzy sets *HRR, RR, AHR, LR, HLR* referring to *hard right*, *right*, *ahead*, *left*, *hard left*.

The output commands are normalized in a scale from 0 to 1 for the translational velocity, where 0 corresponds to complete stop and 1 to maximum speed. Rotational velocity output commands are normalized from -1 to 1, where -1 corresponds to a right turn with maximum angular velocity and 1 to a left turn with maximum angular velocity.

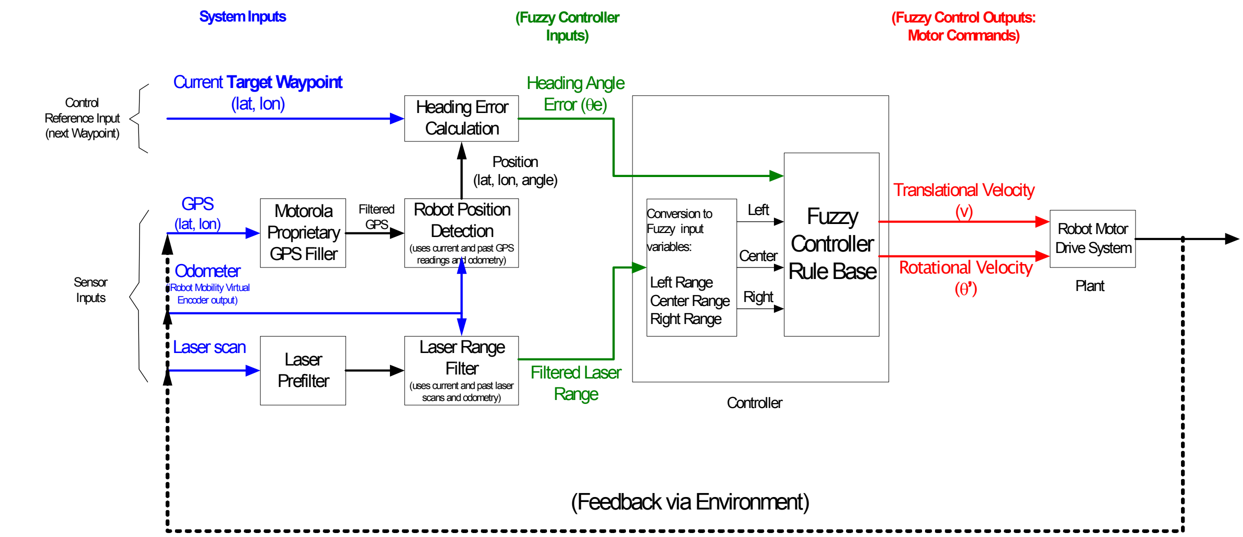

Fig. 3. Controller Block Diagram

Each fuzzy rule j is expressed as:

IF d_1 is D_{j1} AND d_2 is D_{j2} AND d_3 is D_{j3} AND he is HE_j THEN tr is TR_j AND rv is RV_j ;

for j=1,…, number of rules. D_{ji}, is the fuzzy set for d_i in the jth rule which takes the linguistic value of C_i, MD_i, A. HE_j is the fuzzy set for the he which takes the linguistic values FL, L, AH, R, FR. TR_j and RV_j are the fuzzy sets for tr and rv respectively.

The generic mathematical expression of the jth navigation rule is given by:

$$\mu_{R^{(j)}}(d_i, he, tr, r) = \min[\mu_{D_i^j}(d_i), \mu_{HE^{(j)}}(he), \mu_{TR^{(j)}}(tr), \mu_{RV^{(j)}}(rv)]$$

The overall navigation output is given by the max-min composition and in particular :

$$\mu_N^*(tr, vr) = \max_{d_i, he} \min[\mu_{AND}^*(d_i, he), \mu_R(d_i, he, tr, rv)]$$

where $\mu_R(d_i, he, tr, rv) = \bigcup_{j=1}^{J} \mu_{R^{(i)}}(d_i, he, tr, rv)$.

The navigation action dictates change in robot speed and/or steering correction and it results from the deffuzification formula, which calculates the center of the area covered by the membership function computed from the last equation.

I. Experimental Results

Testing and validation of the proposed *MATLAB* based fuzzy logic controller within the overall *distributed-SFX* architecture has been performed using single and multiple robots. Experiments have been performed in an outdoor environment (somewhat uneven terrain) with dirt, grass, trees and some vegetation. The first set of experiments required that the robots travel from an initial position to a distant goal position while avoiding many obstacles in unknown locations (navigation in an area with trees). The second set of experiments required that robots follow sets of waypoints generating search patterns. Patterns include raster scans with two robots starting from different initial positions, avoiding each other as well as other obstacles found in their paths. All pattern and goal points are specified in GPS coordinates

A. Experiment 1

The first experiment demonstrates ability to move in an environment with many unknown obstacles (trees). The robot is given an initial position on one side of a large group of trees and a final goal point location on the other side of the group of trees.

Fig. 4 shows one full path traveled through the tree covered area from the initial point to the final point; periodic laser scans are shown over the course of the robot's path. This experiment has been repeated several times, and Fig. 5 shows the different paths followed by the robot for three repetitions of the same experiment.

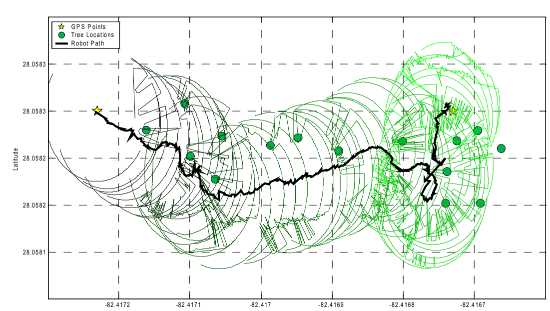
Fig. 4. Robot path – laser scans are shown

Path differences are caused by variations in the input readings produced by differences in GPS reception, and by subtle and cumulative differences in laser reading and odometer readings. The outdoor environment is effectively continuous; hence a small change in movement or initial conditions may result in a different path followed every time. However, in all cases, the robot moves toward its final goal GPS point.

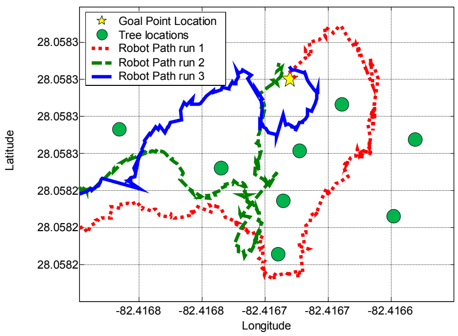

Fig. 5: Three different paths for the same scenario

B. Experiment 2

Experiments have been performed with two robots operating simultaneously. Robots travel through a set of goal points delineating a box shape. The robots start from different initial positions and travel to each of the goal points defining the box. One robot moves clockwise, the other counterclockwise. The paths followed are shown in Fig. 6; similar results have been obtained repeating the same experiment. The robots negotiate a static object (a tree) and a dynamic object (the other robot) as they traverse the set of goal GPS points. While the robots were moving they crossed each other's path and avoided one another as shown in Fig. 7.

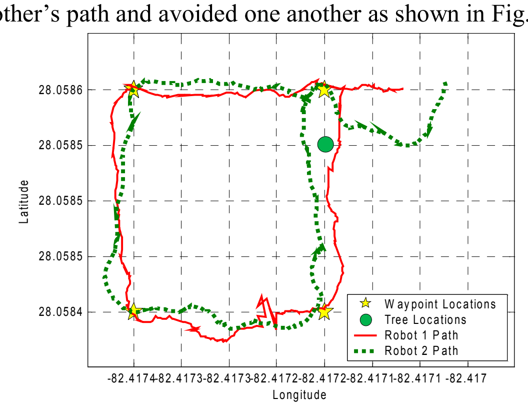

Fig. 6. Path followed by the two robots in the second experiment

C. Experiment 3

A raster search is performed using two robots. Robots start from different positions in the field and move in opposite directions while performing the raster search. The trajectories followed by the two robots are presented in Fig. 8. The experiment has been repeated several times with similar results.

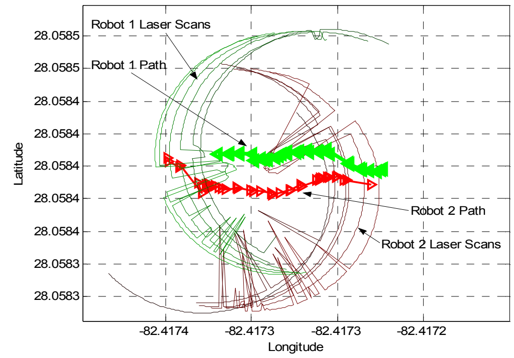

Fig. 7. Robots avoiding each other

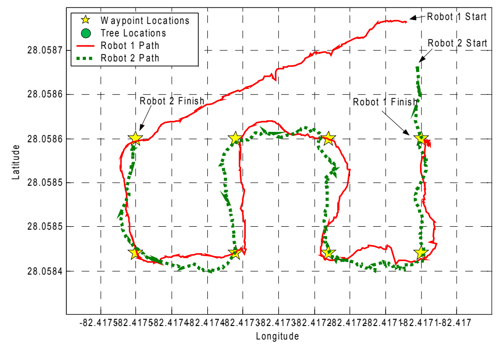

Fig. 8. Raster search performed by the two robots

II. Conclusions and Discussions

This paper has presented a *MATLAB* based approach using fuzzy logic controllers to robot navigation in outdoor environments using fuzzy logic controllers. The multi sensor fuzzy logic robot controllers have been derived and implemented using laser, GPS and odometer data as inputs, also considering fusing of such sensor data (laser, GPS and odometer) filtering out noise. Extensive outdoor environment experiments with one and two mobile robots have quantified sensor inaccuracies affecting navigation and have demonstrated single and multiple robot waypoint navigation, goal point navigation, random and raster scan search patterns along with collision avoidance.

A major advantage of this approach is that it may stand alone as a control architecture, or it may be part of an overall multi layer distributed system architecture where data exchange, commands, recruitment, interpretation and processing of sensor fusion data is accomplished at different levels (obviously with communication delays). The control system may continue functioning under communication loss without affecting robot motion.

Future research will include controller enhancement involving additional sensors like IMU, IR and standard video; derivation of remote supervisory controllers for coordinated movement and variable robot autonomy;

inter robot communication and multi robot distributed control.

REFERENCES

1. Saffiotti, A., "Handling uncertainty in control of autonomous robots," in Uncertainty in Information Systems, edited by A. Hunter and S. Parsons, Springer, LNAI 1455, pp: 198-224, 1998.
2. Murphy, R. R., Introduction to AI Robotics, MIT Press, 2000.
3. Arkin, R. C., Behavior-Based Robotics, MIT Press, 1998.
4. Balch, T., Parker L. E., (Editors), Autonomous Robots, A. K. Press, 2002.
5. Doitsidis, L., Valavanis, K. P., Tsourveloudis, N. C., "Fuzzy Logic Based Autonomous Skid Steering Vehicle Navigation", In *Proceedings of the IEEE International Conference on Robotics and Automation,* pp: 2171-2177, Washington DC, 2002.
6. Valavanis, K. P., Hebert, T., Kolluru, R., Tsourveloudis, N. C., "Mobile Robot Navigation in 2-Dynamic Environments Using Electrostatic Potential Fields," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 30, No. 2, Part A, pp: 187-196, 2000.
7. Tsourveloudis, N. C., Valavanis, K. P., Hebert, T., "Autonomous Vehicle Navigation Utilizing Electrostatic Potential Fields and Fuzzy Logic," *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 4, pp: 490-497, 2001.
8. Abdessemed, F., Benmahammed, K., Monacelli, E., "A fuzzy-based reactive controller for a non-holonomic mobile robot," In *Robotics and Autonomous Systems,* article in press.
9. Lee, T. L., Wu, C. J., "Fuzzy Motion Planning of Mobile Robots in Unknown Environments," *Journal of Intelligent Robotic Systems,* Vol. 37, pp: 177-191, 2003.
10. Aguire, E., Gonzalez, A., "Fuzzy Behaviors for Mobile Robot Navigation: Design, Coordination and Fusion," *International Journal of Approximate reasoning,* 25: 225-289, 2000.
11. Goodridge, S. C., Luo, R. C., "Fuzzy Behavior Fusion for Reactive Control of an Autonomous Mobile Robot: MARGE," In *Proceedings of the IEEE International Conference on Robotics and Automation,* pp: 1622-1627, San Diego, CA, 1997.
12. Ishikawa, S., "A Method of Indoor Mobile Robot Navigation by Using Fuzzy Control," In *Proceeding of the IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS,* pp: 1013-1018, Osaka, Japan, 1991.
13. Li, W., "Fuzzy Logic-based 'Perception-Action' Behavior Control of a Mobile Robot in Uncertain Environment," In *Proceedings of the 3rd IEEE Conference on Fuzzy Systems,* Vol. 3, pp: 1626-1631, 1994.
14. Pin, F. G., Watanabe, Y., "Navigation of Mobile Robots Using Fuzzy Logic Behaviorist Approach and Custom Design Fuzzy Inference Boards," *Robotica,* Vol. 12, 1994.
15. Saffioti, A., "Fuzzy Logic in Autonomous Robotics: Behavior Coordination," In *Proceedings of the IEEE International Conference on Fuzzy Systems,* pp: 573-578, 1997.
16. Saffioti, A., Konolige, K., Ruspini, H. E., "A Multivariable Logic Approach to Integrating and Planning and Control," *Artificial Intelligence,* 76: 481-526, 1995.
17. Tunstel, E., "Mobile Robots Autonomy Via Hierarchical Fuzzy Behavior," In *Proceedings of the 6th International Symposium on Robotics and Manufacturing,* pp: 837-842, 1996.
18. Seraji, H., Howard, A., "Behavior-Based Robot Navigation on Challenging Terrain: A Fuzzy Logic Approach," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 3, pp: 308-321, 2002.
19. Hagras, H., Callaghan, V., Colley, M., "Outdoor Mobile Robot Learning and Adaptation," *IEEE Robotics and Automation Magazine*, pp: 53-69, 2001.
20. Mali, A. D., "On the Behavior-Based Architectures of Autonomous Agency," *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, Vol. 32, No. 3, pp: 231-242, August 2002.
21. Borenstein, J. and Feng. L., "Measurement and Correction of Systematic Odometry Errors in Mobile Robots," *IEEE Journal of Robotics and Automation*, Vol. 12, No. 6, , pp: 869-880, December 1996
22. Ojeda, L. and Borenstein, J., 2003, "Reduction of Odometry Errors in Over-constrained Mobile Robots," In *Proceedings of the UGV Technology Conference at the 2003 SPIE AeroSense Symposium*, Orlando, FL, April 21-25, 2003.
23. Panzieri, S., Pascucci, F., Ulivi, G.,"An Outdoor Navigation System Using GPS and Inertial Platform," *IEEE Transactions on Mechatronics*, Vol. 7, No. 2, pp:134-142, June 2002.
24. Ohno, K., Tsubouchi, T., Shigematsu, B., "Outdoor Navigation of a Mobile Robot between Buildings based on DGPS and Odometry Data Fusion", In *Proceedings of International Conference in Robotics and Automation*, pp: 1978-1984, Taipei, 2003.
25. Thrapp, R., Westbrook, C., Subramanian, D., "Robust Localization algorithms for an autonomous campus tour guide," In *Proceedings of the IEEE International Conference on Robotics and Automation,* Vol. 2, pp: 2065-2071, 2001.
26. Vaneck, T.W., "Fuzzy Guidance Controller for an Autonomous Boat," *IEEE Control Systems Magazine*, pp: 43-51, April, 1997
27. Bruch, M.H., Gilbreath, G.A., Muelhauser, J.W.,and J.Q. Lum, "Accurate Waypoint Navigation Using Non-differential GPS," *AUVSI Unmanned Systems 2002*, Lake Buena Vista, FL, July 9-11, 2002.
28. Long, M. T., Murphy, R. R., Parker, L. E., "Distributed multi-agent diagnosis and recovery from sensor failures," in *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems,* Vol. 3, pp: 2506-2513, 2003.
29. Nelson, A. L., Doitsidis, L., Long, M. T., Valavanis, K. P., Murphy, R. R., "Incorporation of Matlab into a Distributed Behavioral Robotics Architecture", I*n proc. of the 2004 IROS/RSJ conference*, pp: 2028 – 2035, 2004.
30. S. Müller, H. Waller, "Efficient Integration of Real-Time Hardware and Web Based Services Into MATLAB," *ESS'99 11th European Simulation Symposium and Exhibition*, Erlangen-Nuremberg, 1999, ESS'99, Oct. 26-28, 1999. JMatLink download site: http://www.held-mueller.de/JMatLink/
31. L. E. Parker, "ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation," *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 2, pp: 220-240, 1998.
32. A.L. Nelson, E. Grant, T.C. Henderson, "Evolution of neural controllers for competitive game playing with teams of mobile robots," *Journal of Robotics and Autonomous Systems*, Vol. 46, No. 3, pp: 135-150, Mar 2004.
33. G. J. Barlow, T. C. Henderson, A L. Nelson, E. Grant, "Dynamic Leadership Protocol for S-nets," *Proceedings*, *2004 IEEE International Conference on Robotics and Automation* (ICRA), New Orleans, LA, April 2004.
34. R. R. Murphy, R. C. Arkin, "Sfx: An Architecture For Action-oriented Sensor Fusion", *Proceedings of IEEE/RSJ Intelligent Robots and Systems*, Vol. 2, July 7-10, pp: 079-1086, 1992.
35. N. N. Okello, D. Tang, D. W. McMichael, "TRACKER: a sensor fusion simulator for generalized tracking," *Proceedings of Information, Decision and Control, IDC,* pp: 359-364, February 1999.
36. R. Willgoss, V. Rosenfeld, J. Billingsley, "High precision GPS guidance of mobile robots," *Proceedings of the Australasian Conference in Robotics and Automation*, 2003.
37. M. T. Long, Creating A Distributed Field Robot Architecture for Multiple Robots, M.Sc. Thesis, USF, Summer 2004.
38. J. Contreras, A. Losi, M. Russo, "JAVA/MATLAB simulator for power exchange markets", *Power Industry Computer Applications*, PICA 2001. Innovative Computing for Power - Electric Energy Meets the Market. *22nd International Conference on IEEE Power Engineering Society,* pp: 106-111. May 20-24, 2001.