

A Web-Based Mobile Robotic System for Control and Sensor Fusion Studies

Christopher Paolini¹, Gerold Huber², Quentin Collier³ and Gordon K. Lee¹

¹Dept of Electrical & Com Engr.
5500 Campanile Drive
San Diego State University
San Diego, CA 92182

²Management Center Innsbruck
University of Applied Sciences
Universitätsstraße 15
6020 Innsbruck, Austria

³IUT de Bethune
Networks & Telecomm Dept.
62408 Bethune, France

Abstract

This paper presents a hardware system architecture for a mobile robotic system that employs multiple sensors (an ultrasonic array, a thermal sensor, and a video streaming system) to obtain information about the environment, a virtual field strategy for obstacle avoidance and path planning, and an ANFIS controller for path tracking. The particular focus is on the ANFIS controller and its hardware implementation, the graphics-user interface and the overall system architecture. Experimental results and preliminary evaluation show that the proposed architecture is a feasible one for autonomous mobile robotic systems.

Keywords: adaptive neural fuzzy inference control, multi-sensor integration

I. INTRODUCTION

The design of autonomous mobile robotic systems is a complex problem involving many tasks including (a) the development of a control strategy that can compensate for unknown system dynamics and which can change depending upon required tasks and environmental uncertainties; (b) the assembly of several sensors that can provide information to the controller about the robot's current location and surrounding environment; (c) a path planning algorithm to avoid obstacles which achieving a goal and (d) the integration of the associated software and hardware support for a complete system platform.

Several authors have addressed the issues associated with building and designing mobile robotic systems for a multitude of applications. For example, in [1], the authors provide a simple robotic design for educational purposes: a Lego Mindstorm[®] platform with a simple camera and sonar sensors is used to generate experiments to track objects. The platform includes a Handy Board microcontroller board, the CMU camera and a SRF04 sonar sensor.

Further, many researchers have developed control techniques for mobile robotic systems. For example, in [2] the authors develop a fuzzy velocity controller for the Nomad 200 mobile robot. The controller uses a wall-following strategy with both angular and linear velocity.

Li [3] and others designed a real-time fuzzy target tracking control method for autonomous mobile robots by using infrared sensors to obtain target information and then a fuzzy controller for path tracking. In [4], Liu and others used genetic algorithms and fuzzy logic for path planning with obstacle avoidance.

In this paper we employ an ultrasonic array and a thermal sensor to generate data about a mobile robot's environment, including the potential for animate objects such as humans and animals, a virtual field theory to define objects based upon the sensor data and a path planning method using a grid structure, an initial measurement unit (IMU) and an adaptive fuzzy inference neural network controller for path tracking, and a novel video streaming technique to send images of the robot's environment back to a user at a remote site. Further, we have developed the software and hardware to implement these tasks into overall robot architecture. In [5], we reported on some simulation results using the ANFIS controller. The main contributions of this paper include the development of a new internet driven user-interface, hardware implementation of the multi-input, multi-output (MIMO) ANFIS controller, the integration of this controller into the overall robot platform, and the evaluation of the architecture under several goal scenarios.

This paper is organized as follows: for completeness, Section 2 presents the sensors that have been developed for this robot platform as well as the virtual field method and grid generation tasks for path planning. Section 3 presents the ANFIS controller, that is, the control strategy and hardware implementation. In Section 4 the system integration process is discussed while experimental results are provided in Section 5. Finally a summary of the activities to date and future plans are given in Section 6.

II. THE MOBILE ROBOT TESTBED AND SENSOR CONFIGURATION

In order to test the ANFIS controller as well as other learning systems, a mobile robot testbed has been constructed at San Diego State University. The platform for the robot is the iRobot Create[®] system, which is a version of the vacuum cleaning robot Roomba[®]. The platform is designed in such a way that sensors, robotic

arms, and other instruments can be integrated with little effort. Further, the iRobot Create® platform contains a variety of sensors that can be controlled with the iRobot Create's® Open Interface® (OI).

The mobile robot platform contains several sensors [6] as well as an adaptive AJAX-based streaming video system [7]. Video is captured from an onboard camera connected to an embedded single board computer (SBC) through an IEEE 1394 (Firewire) serial interface, all set within the robot's payload bay. Unlike other video streaming techniques, through a novel software architecture, the data rate incurred by the video stream adapts according to the data rate available to a wireless Ethernet interface. The iRobot captures, encodes, and transmits the source stream to a gateway video server over a wireless network. The streaming video server then relays the stream to all Web browsers that are presently browsing the server. This architecture allows multiple people to remotely view the iRobot without increasing the data that is transmitted from the iRobot to the streaming video server.

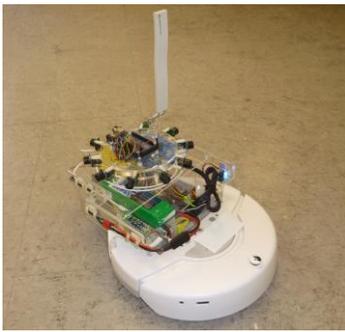


Figure 1: iRobot Create® with Ultrasonic Sensor Array

Figure 1 shows the customized iRobot Create® mobile robot developed at San Diego State University. Mounted within the cargo bay is a Migrus C787 DCF-P single board computer (SBC) with a 1.2GHz Eden ULV Processor. This SBC is designed for digital media devices and home entertainment centers and has a Mini-ITX form factor of 17 cm-sq. The Migrus C7 has one 16-bit (PCMCIA2.1) card socket and one Compact Flash (CF) socket. A Debian derived distribution of the Linux operating system called Voyage Linux, designed to run on embedded x86 platforms, is used for the SBC's operating system.

The acrylic case also holds in place an external 12V 4200mAh Ni-MH rechargeable battery which supplies power to the Migrus C7, the FB-4652, a Unibrain Fire-i™ Digital Camera, and a custom circuit that allows the iRobot to dock and recharge the external 12V Ni-MH battery through the charging base. The Unibrain Fire-i™ is connected to the Migrus C7 through a 400Mbps IEEE1394 interface and supports a video resolution of

640x480 pixels with a maximum frame rate of 30 frames per second (fps). WLAN connectivity is provided through a Linksys Wireless-G notebook PC adapter card, model WPC54G v1.2, which is plugged into the PCMCIA port on the Migrus C7. The Linksys Wireless-G interface supports data rates of up to 54Mbps.

Besides the video system, the robot platform has a thermal sensor and an ultrasonic array sensor system. The thermal sensor is used to detect animate objects which must be avoided as the robot traverses an area [9]. For thermal sensing, a TPA81 sensor is used which can detect infrared data in the 2µm-22µm range. The sensor has eight thermopiles arranged in a row and can measure the absolute temperature of 8 adjacent points simultaneously. The TPA81 module also has a servo that can be used to pan the thermal sensor and construct a thermal image. This sensor has an I2C interface. As a calibration check, both animate and inanimate objects were tested; the thermal sensor is able to detect a candle flame at a range of 6.5 feet and a human body at a range of 3 feet. For the mobile robot, the TPA81 sensor is used in conjunction with a servo to pan a space. A thermal image of heat can be built from 8 adjacent points simultaneously. The resulting image is a 32x8 pixel bitmap.

Object detection is measured through an array of 10 MaxBotix LV-EZ1 ultrasonic range finder sensors (Figure 2). The LV-EZ1 is a 42 KHz transducer that provides three forms of communication: serial, analog, and Pulse Width Modulation (PWM). These sensors provide sonar range information from $D_{MIN}=6$ inches to $D_{MAX}=254$ inches with 1 inch resolution. Each sensor can be connected to either a 3.3 V or 5.0 V microcontroller supply and consumes less than 2 mA of current. Our system requires a total of 10 ultrasonic sensors to cover a 360 degree Field of View (FOV). When firing two or more sensors at the same time there can be crosstalk between sensors. Crosstalk causes unstable readings that can be interpreted as an obstacle that does not exist. To overcome crosstalk, the LV-EZ1 sensors provide a *Daisy Chaining* mode that allows multiple sensors to work together without interfering with each other. The total time for the chain to update 10 sensors is 2 Hz.

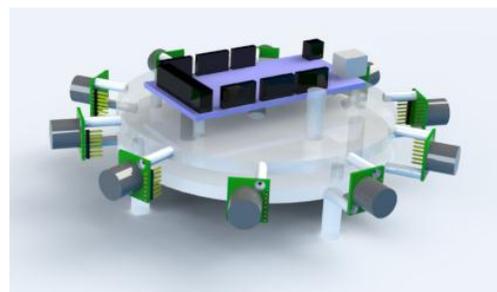


Figure 2: The Ultrasonic Sensor Array and ANFIS Microcontroller

The objective of the mobile robot presented here is to traverse an unknown area while moving towards a goal while avoiding obstacles. For this project, the Vector Field Histogram (VFH) and Virtual Force Field (VFF) methods were implemented to achieve real-time obstacle avoidance using ultrasonic sensor data. The VFH allows the detection of animate and inanimate objects in an unknown environment. The VFF method uses a Certainty Grid (CG) to store confidence values derived from ultrasonic sensor readings. The CG forms a localized map of a robot's environment and is used to determine if an obstacle exists at a particular distance from the robot at a given time.

The method focuses on an obstacle representation by using a two-dimensional Cartesian histogram grid. Each cell (i,j) in the grid holds a Certainty Value (CV) which represents the likelihood of an obstacle lying within cell (i,j). Similar to the Histogramic In Motion Mapping (HIMM) method [8], only the cell (i,j) that corresponds to the sonar distance is updated by a constant. The corresponding cell (i,j) is the one that lies at the acoustic axis of an ultrasonic sensor's FOV. A different variable updates the remaining cells. The corresponding cells are the ones that lie on the acoustic axis. In this way, subsequent readings will clear high CVs if the obstacle has vanished from the FOV of an ultrasonic sensor [9].

The histogram grid is a virtual window that follows the iRobot Create® platform at all times. A 2D histogram grid is generated and mapped to a Cartesian coordinate plane, based upon sensor information.

A nine degree-of-freedom IMU, developed by sparkfun [10] is used for heading information; the device contains a triple-axis accelerometer ADXL345, a single-axis gyro LY530ALH (yaw), a dual-axis gyro LPR530ALH (pitch and roll) and a triple-axis digital magnetometer HMC5843. With the Atmega328 and a modified version of open source code [11], the sensor board becomes a Direction Cosine Matrix (DCM) based Attitude Heading Reference System (AHRS) with gyro drift correction using an accelerometer (gravity) vector and magnetometer (compass) vector.

III. THE ANFIS CONTROLLER

The classical adaptive neural-network based fuzzy inference system (ANFIS) approach [12] requires selection of the consequent and premise parameters which can be done in two ways using input-output datasets: off-line training (supervised learning) and on-line learning (unsupervised learning). The second method can also be used as a fine-tuning of the off-line training parameters.

Both types of learning require a forward pass and a backwards pass to find estimates of the premise and consequent parameter sets. Typically, in the forward pass, the consequent parameters are identified by a recursive least-squares method while in the backwards pass, the premise parameters are identified using a

gradient method. In this effort, we use the Kalman Filter formulation for the least-squares estimation:

$$\left. \begin{aligned} X_{i+1} &= X_i + S_{i+1} a_{i+1} (b_{i+1}^T - a_{i+1}^T X_i) \\ S_{i+1} &= S_i - \frac{S_i a_{i+1} a_{i+1}^T S_i}{1 + a_{i+1}^T S_i a_{i+1}} \end{aligned} \right\} \quad i=0, 1, \dots, P-1 \quad (1)$$

where a_i and b_i are values obtained from the datasets [13], S_i is a covariance matrix, and X_i is the current least squares estimate of the consequent parameters. P is the number of datasets. In the backwards pass, the gradient method minimizes the error measure defined by:

$$E = \sum_{i=1}^P E_i = \sum_{i=1}^P (y_i - y_{di})^2 = \sum_{i=1}^P e_i^2 \quad (2)$$

where y_d is some desired or target output, y is the actual output, and $e_i = y_i - y_{di}$ is the error of the network. This gives the premise parameters.

To revise the filter equations for on-line (unsupervised) learning and to reduce the effects of old measurements, an appropriate approach is to discount the measurement exponentially. Thus, an observation that is τ samples old carries a weight that is λ^τ of the weight of the most recent observations, where $0 \leq \lambda \leq 1$ is the forgetting factor. Then equation (2) can be rewritten as:

$$\begin{aligned} X_{i+1} &= X_i + S_{i+1} a_{i+1} (b_{i+1}^T - a_{i+1}^T X_i) \\ S_{i+1} &= \frac{1}{\lambda} \left(S_i - \frac{S_i a_{i+1} a_{i+1}^T S_i}{1 + a_{i+1}^T S_i a_{i+1}} \right) \end{aligned} \quad (3)$$

resulting in estimates of the consequent parameters. Also

$$E = \sum \lambda^{\tau-i} e_i^2 \quad (4)$$

where E is the error criterion in the on-line learning process. The measurements that are older than τ (the memory horizon) carry a weight less than 0.3 in the error function (4). The criterion (4) can be minimized by a gradient method as in the off-line training case resulting in the premise parameters. Other tuning methods can be found for the ANFIS controller such as in [14]. In this paper, a multi-input, multi-output version of the single-input, single-output ANFIS [13] is used, since there are several inputs to the controller (from the error associated with the sensors) and many outputs need to be generated for the actuation of the robot. Figure 3 illustrates how the ANFIS controller is integrated into the hardware architecture.

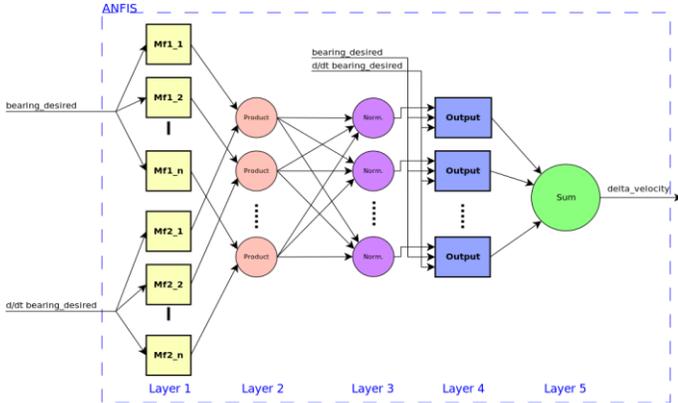


Figure 3: ANFIS Control Structure

In [5], a MISO ANFIS controller was simulated. Figure 4 shows a plot of the desired path with which the ANFIS controller was not trained. The blue line represents the collection of points (x_d, y_d) that identify the desired path the iRobot is to take. The discrete red points show the actual position the iRobot would be located due to actuator error. The blue vector field shows the direction and a *scaled* magnitude of the $(\Delta x, \Delta y)$ 2-tuple at fifty selected positions. To simulate how our iRobot would perform using an ANFIS controller, we used the Player/Stage robot programming and simulation framework [15]. Figure 5 shows a screenshot of the developed simulation [5]. The small circles in the simulation floor plan are imposed cylindrical obstacles with which the iRobot is to avoid a collision. During the simulation it was observed that at no time did an intersection occur between the virtual iRobot and a cylindrical obstacle. The 2-norm of the vector of Δx corrections was 3.81mm and 1.66mm for Δy corrections.

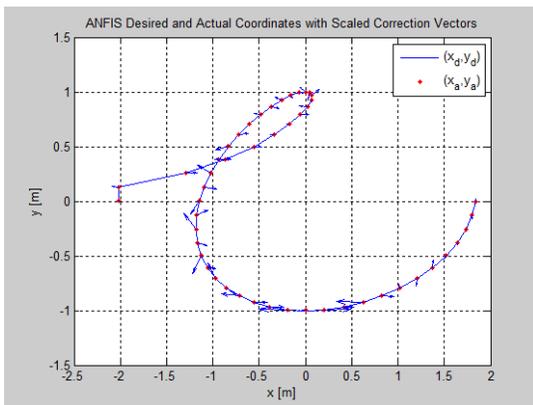


Figure 4: Desired Path and Corresponding Correction 2-tuples.

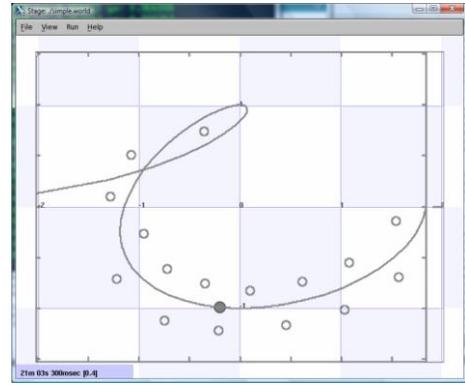


Figure 5: A Player/Stage Simulation of the Path Shown in Figure 4. Note: at 50 discrete coordinates, the robot's position is corrected using the 2-tuples determined by an ANFIS control scheme.

In this paper, results are presented for the actual experimentation. An Atmega1280 based microcontroller board by Arduino is used for implementing the ANFIS controller in hardware; the device has 54 digital i/o pins, 16 analog inputs, 4 serial ports and a 16 MHz crystal oscillator. This board processes all measurements of the inertial measurement Unit (IMU), the ultra-sonic sensors (USS) and the thermal sensor (TS). Furthermore, it runs the ANFIS-controller whenever the command line interpreter on the controller registers a 'pcb2arduino-protocol'-request from the single board computer (SBC). The desired bearing is either given by the graphical user interface (telerobotic mode), or calculated from the Cartesian coordinates of the path planning.

In the single input version only the desired bearing angle is used as the ANFIS input, whereas in the two input version, the time differential of the desired bearing is also used to compensate for the acceleration ramps of the two motors (Figure 3).

The ANFIS System is trained in two steps. In a first step the system is trained off-line to reach a minimal root mean square error based upon training patterns. For this step, two different sets of training data are used to experimentally generate time trajectories. The first dataset is an ideal function of the robot motion while, the second dataset is created from a sensor log while driving the robot in telerobotic mode. The latter dataset provides a better representation of the robot characteristics. Once the ANFIS parameter set is generated using the training patterns, we export the parameter set onto the on-line version of the ANFIS controller, which runs on the robot microcontroller. There the controller is fine-tuned while the robot is running and is capable of compensating changes in the robot characteristics.

IV. SYSTEM INTEGRATION

Each of the individual components of the mobile robot platform (thermal sensor, ultrasonic sensor array, video system, and hardware support systems) have been tested and integrated onto the system architecture. Software interface modules as well as the communication infrastructure between the robot and a streaming video server over a wireless network has also been developed, tested, and integrated. A block diagram of the overall system architecture is provided in Figure 6.

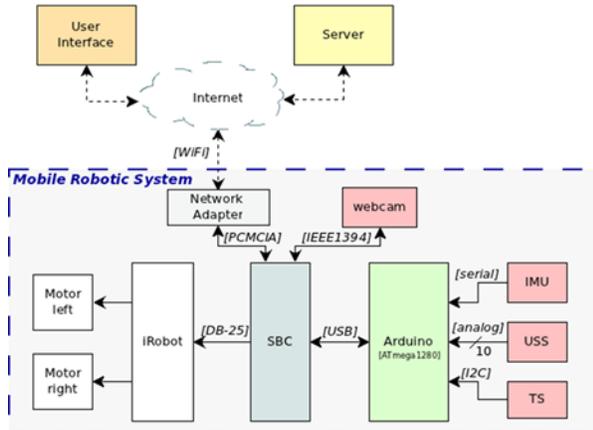


Figure 6: Overall System Architecture

In order to use tele-operated control, a user interface has been created. This interface is programmed with the help of AJAX. The GUI is shown in Figure 7. The upper left block provides information on the heading. A mouse cursor is used to move the robot towards a particular heading angle with a given velocity. For example moving the cursor closer to the center increases the speed of the robot. The two indicators below the target directional indicates the speed of the robot and the heading (in degrees) for the movements of the robot.

The upper right block provides actual video streaming from a small camera. This information is streaming with the Maxwell server.

In order to provide additional data to the user, we have developed a web interface which provides the status of the robot, for example the IP address of our robot, battery indicator, or information of the magnetic bearing.

Finally, the lower right block is the cell presence array, which gives a view of obstacles in space at a particular time, from ultra-sonic sensors data. The black squares are the identified obstacles and the blue square represents the robot.

V. EXPERIMENTAL RESULTS

The ANFIS-based controller using the sensors described above was trained and then used on-line for several different scenarios. Typical sensor time trajectories are shown in Figure 8. Note that for these time trajectories, after training, when on-line

implementation of the ANFIS controller was initiated, the mobile robot performed according to the desired trajectories.

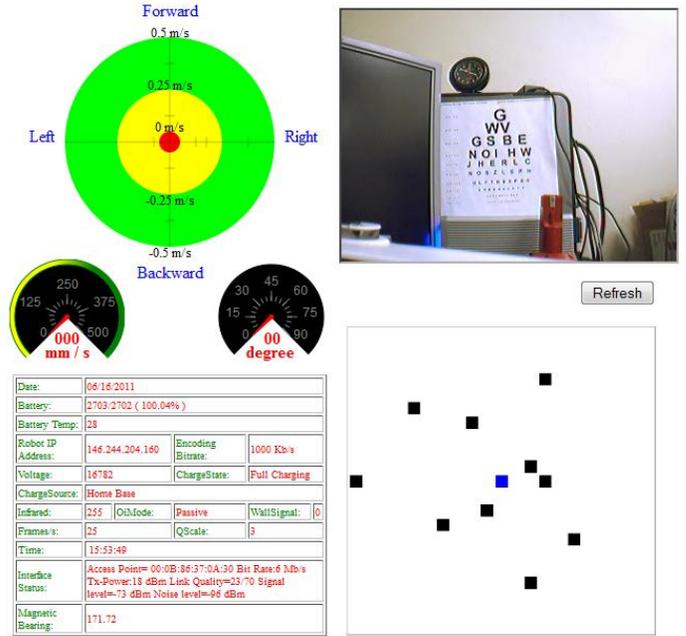


Figure 7: GUI for the Mobile Robotic System

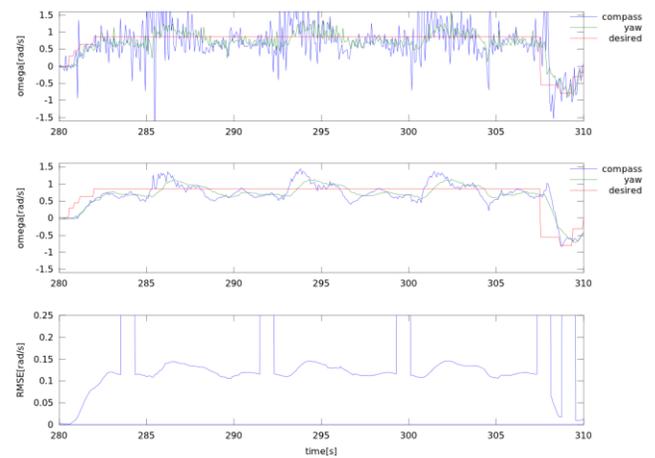


Figure 8a: Time Trajectories for a Constant Turn Scenario

In order to quantify the controller's performance, time measurements were generated for different number of membership functions. Table 1 shows that the respond of the arduino board takes about 14ms. The additional computation time required for the ANFIS calculation is minimal; further, if the IMU is employed (which sends new data with a frequency of 10Hz), the response time does not change too much, as the serial requests of the SBC are prioritized.

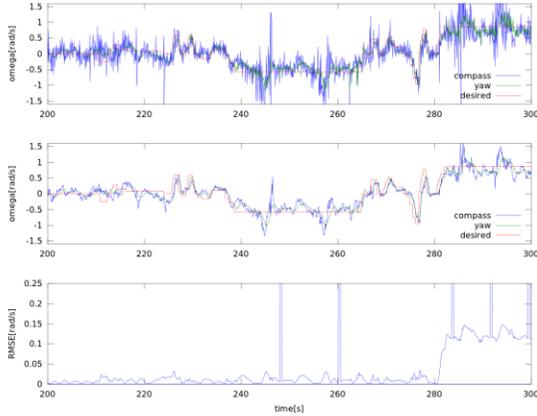


Figure 8b: Time Trajectories for a Desired Bearing Scenario

Table 1: Computational Time for ANFIS Implementation

Inputs	Membership-functions	Computation Time	Response Time incl. ANFIS computation
1	2	3-4ms	16-18ms
	3	7-8ms	20-23ms
	4	9-11ms	23-26ms
	5	13-15ms	26-29ms
	6	17-19ms	30-32ms
	7	22-23ms	36-38ms
2	2	12-14ms	26-28ms
	3	30-31ms	44-46ms
	4	80-82ms	94-96ms

VI. CONCLUSIONS AND FUTURE WORK

This paper presented the implementation of an ANFIS-based controller and associated software and hardware, along with a user-interface for a mobile robotic system. The robot can be run in either tele-operated or autonomous mode and information about the status of the robot is presented to the user during real-time operation. The system has been tested on-line, remotely whereby, for example, a user in Europe can run the robot in San Diego. Real-time video streaming is provided so that the user can see what the robot sees while mobbing along a trajectory.

Future research is planned to make the sensor fusion more efficient as well as in testing the mobile robot platform for different applications. It is expected that the use of this mobile robot testbed will provide additional opportunities for algorithm development as well as human-machine interface advances.

VII. REFERENCES

- [1] H. Dinh and T. Inanc, "Low Cost Mobile Robotics Experiment with Camera and Sonar Sensors", Proc. of the American Control Conference, pp. 3793-3798, 2009.
- [2] M. Mucientes, R. Iglesias, C.V. Regueiro, A. Bugarín and S. Barro, "A Fuzzy Temporal Rule-Based Velocity Controller for Mobile Robotics" *Fuzzy Sets and Systems*, vol. 134, no. 1, pp. 83-99, Feb. 2003.
- [3] T.H. Li, S. J. Chang and W. Tong, "Fuzzy Target Tracking Control of Autonomous Mobile Robots by using Infrared Sensors", *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, pp. 491-501, August 2004.
- [4] Q. Liu, Y.G. Lu and C.X. Xie, "Optimal Genetic Fuzzy Obstacle Avoidance Controller of Autonomous Mobile Robot Based on Ultrasonic Sensors", Proc. of the *IEEE International Conference on Robotics and Biomimetics*, pp. 125-129, 2006.
- [5] A. Gallardo, J. Taylor, C. Paolini, H.K. Lee, and G. Lee, "An ANFIS-Based Multi-Sensor Structure for a Mobile Robotic System" Proc. of the IEEE Symposium on Computational Intelligence, Paris, 2011.
- [6] M. Schaffer, "Sensor Development for a Mobile Robot: Autonomous Human Avoidance", MESA/SDSU Technical Report, 2009.
- [7] C. Paolini and S. Natarajan, "Adaptive AJAX-based streaming video for the iRobot Create Platform for Use in Buildings with Infrastructure Mode 802.11 Networks", Proc. of the ISSCI, World Automation Congress, Kobe, 2010.
- [8] J. Borenstein and Y. Koren, "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, pp. 278 - 288, 1991.
- [9] A. Gallardo, "Implementation of the Vector Field Histogram and Virtual Force Field Methods for Mobile Robots", MESA/SDSU Technical Report, 2010.
- [10] <http://www.sparkfun.com/products/9623>
- [11] <http://code.google.com/p/sf9domahrs/>
- [12] J.-S.R. Jang, "ANFIS: Adaptive-Network Based Fuzzy Inference System", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 3, pp. 665-685, 1993.
- [13] Y. G. Yang, "A Modified Adaptive Networks-Based Fuzzy Inference Controller", Ph.D. Dissertation, North Carolina State University, 1998.
- [14] M. Bishr, Y. G. Yang, and G. K. Lee, "Self-Tuning PID Control Using an Adaptive Neural-Network-Based Fuzzy Inference System", *Journal on Intelligent Automation and Soft Computing*, Vol. 6, No. 4, pp. 271-280, 2000.
- [15] J.-S.R. Jang, "Functional Equivalence between Radial Basis Function Networks and Fuzzy Inference Systems," *IEEE Trans. on Neural Networks*, Vol. 4, No. 1, pp 156-159, Jan. 1993.