Online Map Building for Terrain Scanning Robots using a Hybrid Neurofuzzy Kalman Filter

Homayoun Najjaran
Institute for Research in Construction
National Research Council Canada (NRC)
Ottawa, ON K1A 0R6
Canada
homayoun.najjaran@nrc-cnrc.gc.ca

William W. Melek
AI Division, Alpha Global IT Inc.
North York, ON, M3B2W7
Canada
william@alpha-it.com

Mina Hoorfar
Dept. of Mechanical &Industrial Eng.
University of Toronto
Toronto, ON, M5S3G8
Canada
hoorfar@mie.utoronto.ca

Abstract – This paper presents a proposed online map building for a terrain-scanning robot that is essentially a mobile manipulator built for landmine detection. Map building of the robot is a generic dynamic modeling process to provide a terrain map in real time. It is expected that the proposed map building process be equally applicable to other autonomous mobile manipulators. Map building provides an *a priori* terrain model for obstacle free path planning in parallel to other tasks of the robot. Map filtering, as the core of map building, updates the model in real time using the range measurements obtained from laser and ultrasonic rangefinders.

Filtering takes place by means of a proposed adaptive neurofuzzy Kalman filter. The filter is capable of dealing with uncertainties associated with sensor data and modeling through the learning capability of artificial neural networks in incorporating real life experiments for more efficient terrain typing. Experimental trials provided in Section III validate the effectiveness of the proposed map building process.

Keywords – Mobile manipulators, map building, terrain mapping, adaptive Kalman filter, fuzzy modeling.

I. INTRODUCTION

A large group of mobile robots are categorized as mobile manipulators that consist of an articulated arm mounted on a mobile platform to manipulate sensors, apply tools, and transfer materials. The terrain-scanning robot in Fig.1 [1], and NASA's Mars exploration robots [2] are some examples of mobile manipulators. The mobility enhances the functionality of manipulators mainly because a mobile manipulator can perform remotely in an environment to which human operators have no or limited access. This implies that such robots have to perform autonomously or semi-autonomously, with partial human supervision. Thus, a mobile manipulator needs to calculate an obstacle free path for its joints while moving its end effector, usually attached to the last joint of the arm, towards desired destinations in the task space. Unlike their stationary predecessors, mobile manipulators must deal with an infinite and unstructured task space, which can be as vast as a minefield or the entire surface of a planet, and as unidentified as the bottom of the ocean or the surface of the Mars.

The computation of the obstacle free path is referred to as path planning that is carried out using a model of the task space [3,4]. For terrestrial mobile manipulators, the model is a 3D map that determines the coordinates of every point of the task space with respect to a reference coordinate frame. The coordinates of the points are derived from the distance between the points and a sensor, measuring the distance, given the position and orientation of the sensor in the base coordinate frame. The distance may be measured using active or passive range sensing methods [5]. Active range sensing is carried out using rangefinders. It may be preferred over passive sensing in mobile robots for two reasons: first, it directly provides range data without image processing and triangulation, and second, it is more robust against the change of illumination in indoor and outdoor environments. Typically, rangefinders are either laser or ultrasonic based sensors mounted on a mechanical scanning system with sufficient degrees of freedom to deflect the direction of the sensor for scanning.

Fig. 1 Terrain-scanning robot (courtesy of Defence R&D Canada at Suffield, DRES)

Image registration is a process in which the range measurements are fused to the position and orientation of the rangefinder to produce a 3D raster image (i.e., range image). The range image is expressed in the rangefinder (sensor) coordinate frame and may be transformed into the base coordinate. The combination of producing and transforming the image forms the map building process. Map building may be either offline or online. Typically, mobile manipulators are influenced by extremely dynamic situations because not only their platforms move but also the environment may change over time. Therefore, it is required to adopt online map building that is capable of sensing the environment and updating the model in real time.

The map may be a global map, containing the model of the entire task space, or consist of a series of local models, obtained from partial information of local observations. Integrating the local maps into the terrain map, map building provides a dynamic model of the task space under the assumption that the environment can slightly change between two observations. Map building requires the design of efficient procedures that take place in parallel to other tasks of the robot including joint control, communication, and robot safety procedures. Also, map building and path planning must be developed on a real-time software platform that has a controllable latency, i.e., the maximum delay between the time when a task is invoked and when it actually starts.

Another important problem in map building is with regard to the uncertainties associated with the range measurements and modeling processes. Najjaran [6] proposed a real-time sensor fusion architecture that not only takes into account redundant range measurements of a laser rangefinder but also incorporates the measurements of two ultrasonic rangefinders to improve the reliability of the range measurements.

Fig. 2 portrays the architecture of the map building process. Image registration provides local images, fusing the laser range measurements and the coordinates of the scanning system. An estimate of the range and longitudinal slope of the terrain are obtained from the local images using a batch-processing filter (i.e., the weighted least squares method). The estimated range is fused to the ultrasonic range measurements using a static filter. The result is a better estimate of the range that is incorporated into a Kalman filter to update the final estimate of the range and lateral slope of the terrain in real time. The estimated range, longitudinal slope, and lateral slope of the terrain are sufficient in terms of the degrees of freedom for path planning of the terrain-scanning robot.

Map building features an outlier rejection scheme based on the Mahalanobis distance to determine the discrepancy between the observed and estimated range values. The Mahalanobis distance is to increase the robustness of outlier rejection scheme with respect to different terrains ranging from steep to smooth slopes. The Kalman filter incorporates the observations that have a Mahalanobis distance less than a certain threshold.

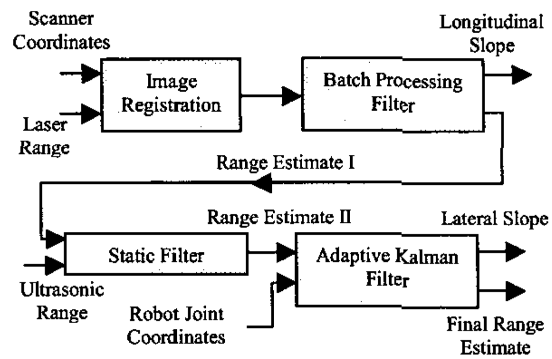

Fig. 2 Architecture of the map building process

II. NEUROFUZZY KALMAN FILTER

The parameter identification and structure of the filters have been explained in detail in [6]. The Kalman filter uses a linear state vector as the terrain model that encompasses the range and slope of a point. The slope disturbances and measurements noise are both modeled by white noise and can be identified by their standard deviation. Furthermore, it has been discussed that the gains of the Kalman filter, and hence, the outputs of the filter (i.e., estimated range and terrain lateral slope), are largely influenced by slope standard deviation, s_{slope}, which varies from one type of terrain to another. Thus, the use of an appropriate s_{slope} can adapt the filter and enhance the performance of the system, accordingly. The determination of s_{slope} involves terrain typing that requires a dynamic modeling method capable of quantifying the terrain characteristics based on real-time observations.

Fuzzy logic modeling was used to develop an a priori fuzzy knowledge base based on the experimental and simulation data. However, the knowledge base of the fuzzy modeling was limited to the cases defined by the operator. In the current approach, the hybrid neurofuzzy knowledge base is used to obtain s_{slope}. The inputs into the terrain typing procedure are the statistics of the observations (i.e., the mean and standard deviation of the range and slope) and the output is the slope standard deviation as shown in Fig. 3.

After the initial development and during the system experimental evaluation phase, supervised learning [7] is implemented through a human "teacher", i.e., the mobile robot operator, to monitor the output of the fuzzy knowledge base being computed through parametric inference engine [8] from the set of fuzzy rules build in the initial design phase. When the supervised learning phase concludes in every experimental trial, the input-output records are classified as training features and stored into database stored in the onboard memory of the robot as shown in Fig. 4. Over a sufficient period of time, enough training records became available in the robot database that provides the training set of a supervised neural network batch learning, which can incorporate the knowledge into the proposed modular neurofuzzy knowledge base.

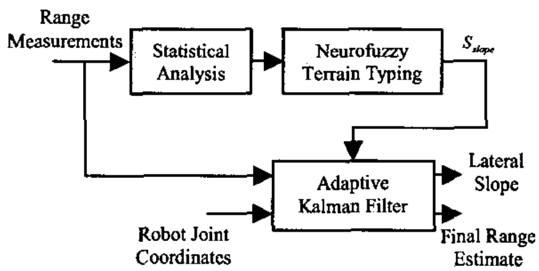

Fig. 3 Adaptive neurofuzzy Kalman filter

Supervised batch learning [7] is implemented through the use of multilayer perceptron (MLP) feedforward neural networks with error back propagation. Moreover, through random records selection, some of those features are used periodically to fine-tune the parameters of the initial fuzzy logic knowledge base. Fine-tuning is achieved through Sugeno-Yasukawa parameter perturbation technique to minimize the entropy prediction error of the fuzzy knowledge base according to a gradient-descent learning algorithm [9].

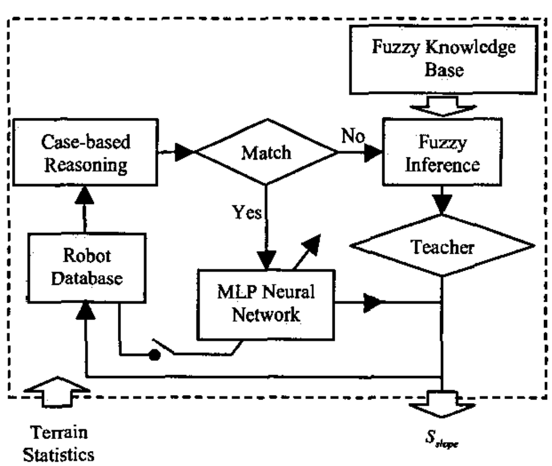

Fig. 4 Supervised neural network batch learning

Hence, as shown in Fig. 4, for future implementation the expert system can use "case-based" reasoning to select the most suitable input-output mapping for S_{slope} inference when presented with a "new" premise in the form of real life range measurements.

In other words, if the set of inputs is not "relatively" close to any of the records available into the system database, we use the fuzzy expert system to infer the initial output and then we verify such output through supervised "teacher" learning prior to final registry. Otherwise, if the premise is close to one or more of the records in the database, we use the trained feedforward neural network to predict the system output. For the incoming input to be similar to one or more of the existing records in the mobile robot training database the following condition based on the Fornobius norm J of the difference between the record input(s) and the system input(s) should be satisfied:

$$J_i = \left\| X_{DB} - X_i \right\|_A \le \vartheta_g \quad i = 1, \dots, q \tag{1}$$

where $X_i = \{x_{i1}, x_{i2}, \dots, x_{ir}\}$ is the vector of inputs to the adaptive neurofuzzy Kalman filter, XDB is the $q \times r$ matrix of existing training records in the mobile manipulator database, q is the current number of training records, and ϑ_g is the positive threshold that depends on the manipulator working environment.

A. Fuzzy Logic Terrain Typing

For the case of the Kalman filtering in Fig.4, the terrain typing model is a linear stochastic process that relates two independent states of the system: the estimated range and the lateral slope of a point that is the change of the range with respect to the change of the turret angle at the point. The terrain undulations are modeled using a linear state vector whose state transition and noise input matrices are invariant. In [6], it was determined that the optimality of the model largely depends on two parameters of the system: 1) the standard deviation of the probability distribution of the lateral slope disturbance; and 2) the variance of the joint probability distribution associated with the measurements of the sensing system. Thus, to complete the parameter identification of the modeling procedure it is required to specify two parameters: slope standard deviation S_{slope}, and the measurement variance. The latter is determined based on the statistical analysis of the sensors, batch processing filter, and the static filter. However, the former depends on the characteristics of the terrain that may vary significantly for different types of terrain. Thus, the performance of the system may be optimized if S_{slope} is updated online based on the terrain type. The online determination of S_{slope} involves terrain typing that requires a dynamic modeling method capable of quantifying the nature of the surrounding terrain based on real-time observations. One approach may be the use a qualitative modeling method known as black box approach. Black box modeling is mainly used in two circumstances: 1) when an exact mathematical model is not available; and 2) when a mathematical model is available but the desired outputs differ from the actual outputs of the system. Terrain typing for the determination of the slope standard deviation involves both cases. First, due to the uncertainties associated with the sensing system, there is no precise mathematical model available to relate the slope standard deviation to the measurements. Second, the actual characteristics of the terrain are not necessarily the same as the desired outputs of the filters used in the map building procedure. The inputs into the terrain typing procedure may be the statistics of the observations, and the output is the slope standard deviation. A qualitative model may represent the behavior of a system by linguistic IF-THEN statements called rules. The rules may be attributed to flat, rugged, and protuberant terrain, respectively.

Fuzzy logic modeling is a special case of qualitative modeling that can deal with quantitative data using the fuzzy sets. Being modeled by fuzzy logic, the terrain characteristics are directly quantified by the model output. Terrain typing is carried out based on a priori knowledge base developed offline according to simulation and experimental data. The knowledge base is used by the fuzzy inference mechanism to determine the slope standard deviation that significantly influences the Kalman gains, and hence, the outputs of the filter (i.e., estimated range and lateral slope of the terrain). Thus, fuzzy modeling can adapt the Kalman filter based on the observations acquired in real time.

In general, fuzzy systems are based on the concept of fuzzy partitioning of information. The decision-making ability of the fuzzy model depends on the existence of a set of rules and a fuzzy reasoning mechanism. Our methodology for fuzzy modeling of terrain typing considers a parameterized formulation of reasoning [8]. The inference engine to compute the fuzzy output of the model, E, is a linear combination of two extreme reasoning approaches, Mamdani's and Logical, with adjustable parameters. The crisp output S_{slope} is obtained using the BADD method as follows [8]

$$S^{*}_{slope} = \frac{\int_{S_0}^{S_1} S_{slope}[E(S_{slope})]^{\alpha} ds}{\int_{S_0}^{S_1} [E(S_{slope})]^{\alpha} ds} \qquad 0 \le \alpha \le \infty \tag{2}$$

where $E(\bullet)$ is the fuzzy output, α is the defuzzification parameter, and S^{*}_{slope} is the terrain typing model output after defuzzification.

The fuzzy knowledge base is developed based on the optimum number of rules, input and output membership functions, and the appropriate overlap between the membership functions. Rule generation is carried using out clustering the input-output data available a priori from simulation and experiments. In this methodology, the output space is clustered first, and then, the input space fuzzy partitions are derived by projecting the output space partitions onto each input space, separately. The output fuzzy clustering is carried out using the Fuzzy C-Means (FCM) algorithm [8]. The idea of fuzzy clustering is to divide the output data into fuzzy partitions that overlap each other. Therefore, the containment of each datum $S_{slope}(k)$ to each cluster i with a center v_i is defined by a membership grade u_{ik} in $[0,1]$. The membership grades and the cluster centers are obtained through an iterative procedure as follows:

$$u_{ik,l} = \left[\sum_{j=1}^{c} \left(\frac{\sqrt{S_{slope(k)} - v_{i,l-1}}}{\sqrt{S_{slope(k)} - v_{j,l-1}}} \right)^{\frac{2}{m-1}} \right] \tag{3}$$

where

$$v_{i,l} = \frac{\sum_{k=1}^{N} (u_{ik,l})^m S_{slope(k)}}{\sum_{k=1}^{N} (u_{ik,l})^m} \tag{4}$$

and N is the number of data initially available from simulations and experiments integrated together, $u_{ik,l}$ and $v_{i,l}$ are the membership grade of the output S_{slope} in the cluster i and the center of the cluster i, respectively. The number of output clusters, and hence the number of rules in the fuzzy model, is determined using the cluster index validity given by:

$$s_{cs} = \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik})^m \left((S_{slope(k)} - v_i)^2 - (v_i - \bar{v})^2 \right) \tag{5}$$

where $\bar{v}$ is the weighted mean of data, and m is the order of fuzziness of the model. The optimal number of clusters c corresponds to minimum S_{cs}.

B. MLP Neural Networks and Learning

The neural networks used for learning the terrain typing in Fig. 5 consists of one input layer together with one output layer that represent the system inputs & outputs, respectively, and one hidden layers that provide the learning capability for the network. The output of the MLP network, therefore, can be represented as follows:

$$\hat{S}_{slope}(x_1, \ldots, x_p) = \sum_{i=1}^{M} \omega_i \cdot g\left(\sum_{j=1}^{p} w_{ij} x_j - \theta_i \right) \tag{6}$$

where $\hat{S}_{slope}(.)$ is the network output, $[x_1, x_2, \ldots, x_p]$ is the input vector having p inputs from range measurements of the robot environment, M denotes the number of hidden neurons, w represents the hidden layer connection weights, θ is the threshold value associated with hidden neurons, and ω represents the output layer connection weights which in effect serves as coefficients to the linear output function.

The back propagation algorithm [7] is used to train the MLP network to predict the terrain slope standard deviation from the range measurements using the information accumulated in the mobile robot database in Fig 4. The back propagation algorithm works by propagating errors backwards from the output layer to the input layer. In other words, assume that w_{ij} denotes the connection weight from i^{th} neuron in layer $L-1$ to j^{th} in layer L, x_j signifies the input to j^{th} neuron, y_j represents the corresponding output, d_j is the desired output, then the input to unit j is:

$$x_j = \sum_i y_i w_{ji} \tag{7}$$

And the output of unit j is:

$$y_j = \frac{1}{1+e^{-x_j}} \tag{8}$$

The back-propagation algorithm attempts to minimize the global error which, for a given set of weights, is the squared difference between the actual y and desired d outputs of a unit, i.e.,

$$E = \frac{1}{2}\sum_c \sum_j \left(y_{j,c} - d_{j,c}\right)^2 \tag{9}$$

where E denotes the global error, and c is a training case available in the mobile manipulator database. The difference equation required to update the connection weights in (6) for back-propagation training is

$$\Delta w_{ji} = \eta \cdot \delta_{j,c} \cdot y_{i,c} \tag{10}$$

where η is the learning rate, $\delta_{j,c}$ is the error at neuron j in layer L, and $y_{i,c}$ is the output of neuron i in layer $L-1$.

III. EXPERIMENTS

The experiments of this research were carried out on a demining robot, MR2. The robot has been manufactured by Engineering Services Inc. (ESI) Toronto for the Defence R&D Canada at Defence Research Establishment Suffield (DRES). The MR2 is a dual-arm mobile manipulator capable of traveling on off-road terrain and scanning the terrain using a metal detector in a manner similar to a human operator. The upper arm carries a scanning laser rangefinder. It has three degrees of freedom that is sufficient for image registration and map building. The scanning laser rangefinder has a rotating mirror that adds another degree of freedom to the range sensing system. The upper arm also carries four ultrasonic rangefinders. The autonomous motion is synthesized based on a terrain map that is updated at each 50 ms, which is the interval required for the acquisition of one laser scan. Path planning computes the desired joint angles using the map. Path planning is executed five times faster than map building to provide a smooth motion from one position to the next. The desired joint angles are input to PID controllers to control the joints independently.

A simulation program has been developed using MATLAB. The program simulates the motion of the end effector of the terrain-scanning robot either from the side or front view of the end effector. The software has been successfully implemented into MR2. The performance of the robot was examined on different types of terrain and the results of the path planning procedure were acquired for several experiments conducted on artificial and natural terrains.

A fuzzy model was developed as the initial means for terrain typing methodology shown in Fig. 3. Fuzzy modeling input parameters were generated by the simulation program for a variety of theoretical profiles (step, ramp, curved, etc.). The inputs include the mean and standard deviation of obstacle height H, change of height S, and the second order change of height δS, that are denoted by M_H, σ_H, M_S, σ_S, and $M_{\delta S}, \sigma_{\delta S}$, respectively. The output parameter that is the appropriate slope standard deviation for each profile was carefully crafted by an operator, i.e., expert knowledge, considering the motion of robot end effector using the simulation program. Several fuzzy knowledge bases were developed using the input-output data. Fig. 5 shows a knowledge base developed based on six input parameters and five rules.

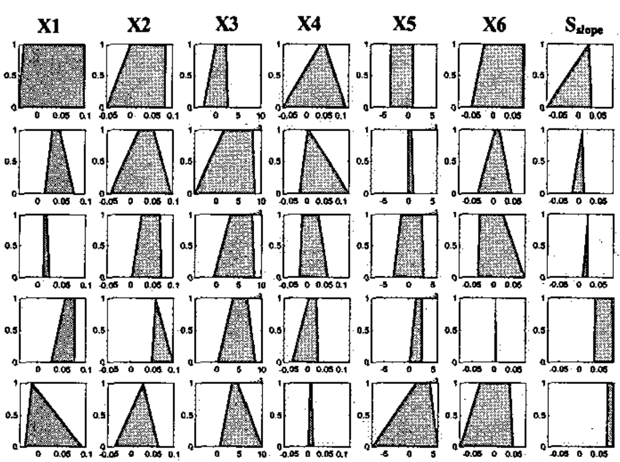

Fig. 5 Terrain typing fuzzy knowledge base

The robot was then examined in real experiments and 28 trials were performed. Range measurements were acquired for each trial and the fuzzy knowledge base in Fig. 5 was used to predict S_{slope} in every case. The operator, i.e., teacher monitored the process during every experimental trial. A total of 22 slopes predicted by the fuzzy knowledge base were accepted by the critic and 5 had to be modified. The results were stored in the robot database for future learning. The MLP neural network in Fig. 4 was then trained by back propagation learning to incorporate the 27 experimental trials in the database. The MLP neural network has 6 input neurons, 12 neurons in the hidden layer, and one output neuron to predict S_{slope}. Fig. 6 shows the comparison between the MLP network output and the output of the 27 training records in the database after 10 epochs of back propagation learning. The results validate the capability of the proposed adaptive neurofuzzy Kalman filter to estimate the terrain slope standard deviation from incomplete information using the fuzzy knowledge base, as well as to learn from real range measurements through reinforcement learning.

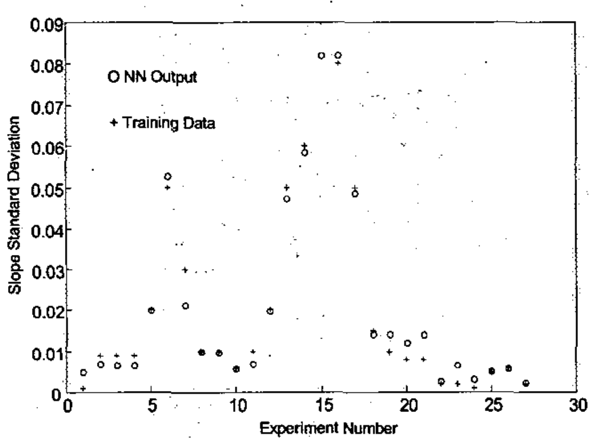

Fig. 6 Comparison between training record output and the output of the trained MLP network

For future implementations, more experimental training records will accumulate in the robot database and hence the neural network can incorporate more information about various terrain features resulting in less and less need for human intervention to correct some "unsatisfactory" S_{slope} values predicted by the knowledge base.

Furthermore, from the experiments in [6], it was concluded that among the input parameters the standard deviation of height, σH, the mean of the change of height, Ms, and the standard deviation of the change of height, σs, are the more significant parameters for terrain typing. This observation can help generate simpler (3-input 1 output) structure fuzzy knowledge base for terrain slope prediction as well as the MLP for learning from real-life range measurements.

IV. CONCLUSIONS

The use of a Kalman filter in the map building procedure of terrain scanning robots allows for maintaining a local map of the terrain and updating it online. The Kalman filter requires a dynamic model of the process. The model is a linear stochastic model that represents the terrain undulations. An important parameter of the stochastic model is the standard deviation of the probability distribution of the process disturbances. Depending on the terrain type, the slope standard deviation influences the output of the Kalman filter significantly. In this work, the online identification of the slope standard deviation was carried out using a hybrid neurofuzzy adaptive Kalman filter that can be used in many signal and data processing applications. The systematic modeling uses only input-output records with little a priori knowledge about the type of the terrain. Therefore, this method may be used for characterization of the terrain in applications such as planetary mobile robots. The significance of the proposed neurofuzzy terrain-typing module was validated experimentally from experiments on the mine detection robot (MR2).

REFERENCES

[1] Y. Das, K. Russell, N. Kircanski, A. Goldenberg, "An articulated robotic scanner for mine detection-a novel approach to vehicle mounted systems," Proceedings of the SPIE Conference, Orlando, Florida, pp. 5-9 April 1999.

[2] "Mars Exploration Rover Launches", NASA Press Kit, pp. 38-43, June 2003.

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," International Journal of Robotics Research, vol. 5, no. 1, 1986.

[4] E. Krotkov, R. Hoffman, "Terrain mapping for a walking planetary rover," IEEE Trans. Robotics and Automation. Vol. 10, Dec 1994, pp. 728-739.

[5] M. Hebert, "Active and Passive Range Sensing for Robotics,", IEEE International Conference on Robotics and Automation, April 2000.

[6] Najjaran H., "Uncertainty Management Method for a Terrain Scanning Robot", Ph.D. Dissertation, University of Toronto, 2002.

[7] S. Haykin, Neural Networks: A Comprehensive Foundation. Second Edition, Prentice Hall, 1999.

[8] Emami M. R, Turksen I. B., Goldenberg A. A.; "Development of a systematic methodology of fuzzy logic modeling"; IEEE Transactions on Fuzzy Systems, Vol.6, No.3, pp. 346-361, 1998.

[9] Sugeno M., Yasukawa T.; "A fuzzy-logic-based approach to qualitative modeling"; IEEE Transactions on Fuzzy Systems; Vol.1, pp. 1-31, 1993.