Self-Organizing Fuzzy Haptic Teleoperation of Mobile Robot Using Sparse Sonar Data

Ondrej Linda, Milos Manic, *Senior Member, IEEE*

***Abstract*— Mobile robot teleoperation has been used in many areas of industrial automation, such as explosives disposal, nuclear waste manipulation, freight handling or transportation. Here, the commonly provided audio-visual feedback often resulted in an inadequate perception of the remote environment. Haptic augmentation was shown to improve and positively enhance the control of the mobile robot. This paper presents a novel Self-Organizing Fuzzy Adaptive Mapping algorithm (SOFAMap) for a haptic teleoperation of mobile robots. The SOFAMap algorithm was specifically developed for a mobile robot with a rotational sonar sensory system, constituting an alternative to a traditionally used multi-sonar array. The main contributions of this work are: 1) development of a specific self-organizing environment mapping structure inspired by the Growing Neural Gas algorithm; 2) incorporating a fuzzy controller into the algorithm to adapt to robot's motion; 3) resolving typical issues such as sensor noise, communication time delay and low sampling rate. The experimental testing was performed in both virtual environment and on a real robotic platform, consisting of a Lego NXT mobile robot and a Novint Falcon 3-DOF haptic interface. The results showed that a high-fidelity haptic feedback can be successfully generated using a simpler and more affordable rotational sonar sensory system, as opposed to the typical multi-sonar array. Further, it was demonstrated that the SOFAMap algorithm improves the operator's awareness of unstructured environments, making it applicable to wide range of mobile robot teleoperation systems.**

***Index Terms*— Fuzzy Control, Mobile Robot Teleoperation, Self-Organization, Sonar Sensor, Haptic Interface.**

I. Introduction

Robot teleoperation as an active area of research significantly contributes to an increasing use of robots in industrial automation as well as in operations under hazardous conditions [1]-[4]. It is widely used in applications, where the presence of the human operator in the unstructured and possibly contaminated environments is not desirable.

However, the teleoperation of mobile robots constitutes a difficult task. The low quality of the information delivered to the operator has a negative impact on the perception of the remote environment and often leads to incorrect decisions [5], [6]. For instance, relying exclusively on the video feedback commonly leads to disorientation, incorrect depth estimation or failure to detect obstacles in unstructured environments. These negative effects of the separation of the operator from the point of action become even more significant in applications where precise maneuvering is required.

Recently, haptic interfaces have been used to enhance the perception of the operator [7]-[11]. Haptic devices can significantly increase the notion of telepresence and provide additional sensory feel that can improve the depth judgment and obstacle awareness [8], [12]. Further, the force-feedback applied against the control motion, prevents the operator from imposing additional danger to the robot by performing an incorrect maneuver.

A typical problem associated with haptic teleoperation is variable communication time delay [13]-[16]. The desired teleoperation system must maintain stability and transparency despite strong variations of time delays and packet losses in the communication network. In addition, the delayed haptic feedback can cause significant disturbances in the operator's perception [17].

Multimodal control interfaces, such as systems combining audio-visual and haptic feedback, substantially increase the safety of mobile robot teleoperation applications. For instance, safe control is essential for robotic operations in hazardous environments, search and rescue response, disposal of explosives and nuclear power plant maintenance or radioactive waste manipulation [18]-[20]. Further, safe driving systems are important in many industrial applications such as freight handling or transportation [21].

The mobile robot teleoperation system augmented by the haptic feedback requires a high fidelity haptic display as well as an accurate sensory system to map the robot's environment. Multi-sonar array constitutes a traditionally used input sensory system for multi-directional obstacle detection [2], [9]. Deficiencies of such input system are its large physical dimensions, high communication bandwidth requirements and high price. This paper considers an alternative of a rotational sonar sensory system, preferred for its small size, low-cost and suitability even for small-sized, low-cost robotic platforms.

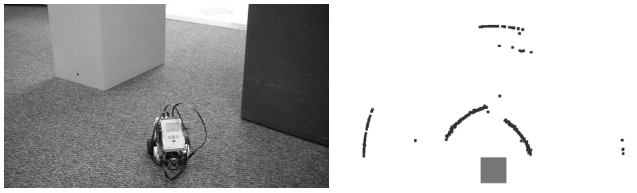


Fig. 1. Robot in an unstructured environment and the corresponding image obtained from the rotational sonar sensory system .

The presented algorithm alleviates the tradeoffs for the low cost, such as considerable measurement errors and low sampling rate. Fig. 1 shows an example of the environment mapping with a rotational sonar sensor. Measurement errors due sensor imprecision are apparent when comparing the sampled points with the real environment.

This paper presents the Self-Organizing Fuzzy Adaptive Mapping algorithm (SOFAMap) for haptic teleoperation of mobile robots. The introduced algorithm was specifically developed to cope with the weak points of the rotational sonar sensory system. The core of the algorithm is a self-organizing neural network structure inspired by the Growing Neural Gas algorithm (GNG) [22], [23]. Firstly, by minimizing the approximation error with respect to the input data, the SOFAMap algorithm locally optimizes the resolution of the modeled environment. Secondly, it contains an inherent mechanism for adjusting to a variable time delay. Thirdly, a built-in intelligent fuzzy controller is used to adapt the constructed model to robot's motion. These improvements, embedded in the SOFAMap algorithm, together result in a high-fidelity haptic teleoperation system for a mobile robot.

Significant amount of work has been already done in the area of robotic environment mapping. For instance, refer to the work of Thrun or Thorpe [24]-[26]. However, unlike in the referred work, the presented algorithm comprises a specific approach for modeling of the robot's surrounding with emphases on the consequent generation of haptic feedback.

The rest of the paper is organized as follows. Section II introduces the SOFAMap algorithm. Section III describes the generation of the haptic force-feedback using the constructed SOFAMap structure. Section IV experimentally evaluates the performance of the implemented system. Finally, conclusion is given in section V.

II. The Self-Organizing Fuzzy Adaptive Mapping Algorithm (SOFAMap)

This section describes the SOFAMap algorithm, a fuzzy-neural approach for robot's environment mapping. The algorithm is inspired by the Growing Neural Gas algorithm (GNG). An adaptive environment model is being constructed online, incrementally processing the incoming inputs from the rotational sonar sensory system. The self-organizing neural structure is coupled with a fuzzy controller to further enhance the accuracy of the model considering the limitations of the system such as the communication time delay.

A. Related Background

The GNG algorithm was proposed by Fritzke [22]. It was originally inspired by the neural gas algorithm presented by Martinetz and Schulten [27]. As a clustering and vector quantization technique, the GNG algorithm is capable of overcoming some of the limitations of the standard self-organizing maps. It is a combination of an incrementally growing structure and a winner take all approach. The GNG algorithm is able to adapt itself to the local dimensionality and density of the input data. Inherently, it is robust against noise.

Fig. 2 demonstrates the ability of the GNG algorithm to expand and accurately model an arbitrary data distribution. It can be observed that the resolution of the GNG network resembles the density of the input data distribution.

The rotational sonar sensory system used in this paper consists of sonic range finder mounted on a continuous servo motor. The servo motor periodically swings around the reference forward direction. The servo motor is equipped with an internal rotation sensor measuring the current angular displacement relatively to the pre-set reference direction.

Sonar sensor has typically a limited range of sight and the distance and angle measurements contain a considerable error. The sampling rate is constrained by the actual communication time delay. Multiple scans are necessary for obtaining a complete image of the environment. Further, the rotational sonar sensory system is attached to a mobile robot. Coupled with the robot's motion, the angular and distance error of past measurements significantly increase.

Fig. 3 shows a schematic representation of the rotational sonar sensory system. The *xy*-coordinate system is defined relatively to the position and direction of the robot. The maximum range of sight is denoted by the symbol R (for the actual sonic sensor user $R = 2m$). Measurement M_t consists of angle $\hat{\alpha}_t$ and distance $\hat{d}_t$ from the sensor to the nearest obstacle in the given direction.

B. The SOFAMap Algorithm

The boundary of the robot's view is approximated with a polyline structure. The polyline is calculated relatively to the robot's location and direction of movement. Line segments are connected via neurons. However, the segments between neurons only define topological neighbors for particular neurons, rather than propagating input signals as it is the case with classical artificial neural networks. Every neuron n_i

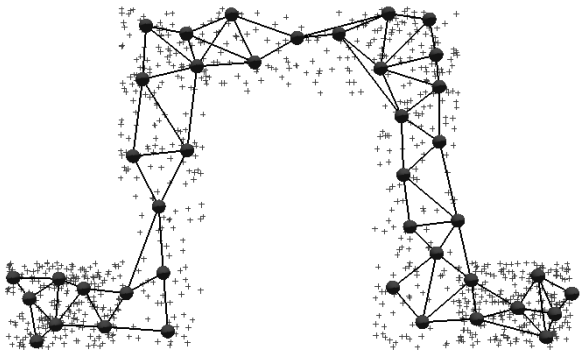


Fig. 2. The GNG network learns the topology of the input dataset.

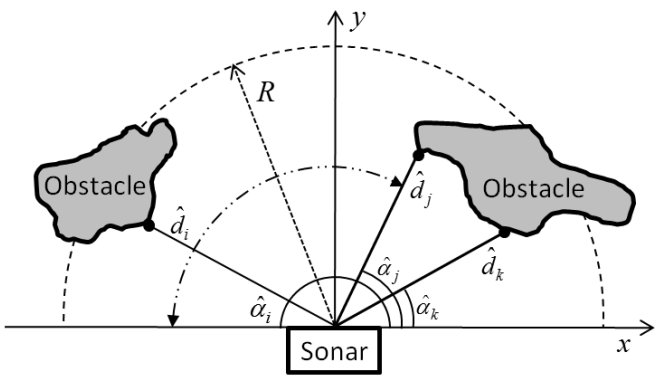


Fig. 3. Schematic representation of the rotational sonar sensory system.

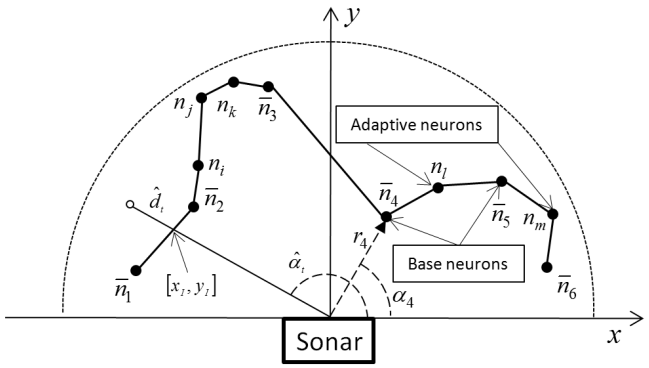


Fig. 4. Example of the constructed SOFAMap structure consisting of both base and adaptive neurons.

maintains distance r_i and angle α_i with respect to the sonar sensor and its local error variable $error_i$ and weight w_i. In the rest of the paper this polyline will be referred to as the SOFAMap structure.

Unlike the original GNG network, the introduced SOFAMap structure is a heterogeneous architecture with two types of neurons: base neurons and adaptive neurons. The base neurons hold a regular, stable but coarse boundary of the whole view. The base neurons are never removed from the structure. They constitute the lowest level of detail of the constructed environment model. The adaptive neurons are dynamically inserted into the structure to locally increase the resolution in the unstructured segments of the robot's view. Additionally, the adaptive neurons can be dynamically removed from the structure should the resolution be decreased. Fig. 4 illustrates the SOFAMap structure consisting of both base and adaptive neurons.

The SOFAMap algorithm goes as follows:

Init Step: Initialize the SOFAMap structure using only the base neurons. They are uniformly distributed around the view and positioned at the maximum sight distance from the sonar sensor. The error variables are initialized to zero.

Step 1: At time t a new measurement M_t, consisting of the measured distance from the sensor to the obstacle $\hat{d}_t$ and the angle $\hat{\alpha}_t$, is obtained by the rotational sonar sensory system:

$$M_t = \{\hat{d}_t, \hat{\alpha}_t\} \tag{1}$$

Step 2: Find neuron n_a with the smallest angular distance from the measurement $\hat{\alpha}_t$.

$$n_a = \{n_i : \min_i |\hat{\alpha}_t - \alpha_i|, i = 1,2,...,N\} \tag{2}$$

Here N denotes the number of neurons currently in the structure.

Step 3: Compute the new distance $r_{a,t}$ of neuron n_a from the sensor according to:

$$r_{a,t} = \frac{\Phi\ r_{a,t-1} + \hat{d}_t}{\Phi + 1} \tag{3}$$

Here Φ denotes a weight that discounts the previous value of distance $r_{a,t-1}$. The value of Φ is determined by a fuzzy controller based on the speed and the turning rate of the mobile robot. The computation of Φ will be explained in details after the description of the algorithm. Upon updating the distance $r_{a,t}$, the computed value of Φ is stored as the weight w_a of neuron n_a.

Step 4: Update the local error variable $error_a$ of neuron n_a:

$$error_{a,t} = error_{a,t-1} + \left[\hat{d}_t - \sqrt{x_I^2 + y_I^2}\right]^2 \tag{4}$$

Here x_I and y_I denote the x and y coordinates of the intersection of a line at angle $\hat{\alpha}_t$ from the sonar sensor and the SOFAMap structure as shown in Fig. 4.

Step 5: Discount the locally accumulated error $error_i$ and the weight w_i of every neuron in the SOFAMap structure:

$$error_{i,t} = error_{i,t-1}\ \delta_{err}\ ,\ \ i = 1,2,...,N \tag{5}$$

$$w_{i,t} = w_{i,t-1}\left(\delta_w\right)^{|\hat{\alpha}_t - \hat{\alpha}_{t-1}|}\ ,\ \ i = 1,2,...,N \tag{6}$$

Here δ_{err} and δ_w are heuristically determined discount rates for the local error and the weight, respectively. In the original GNG algorithm, parameter δ_{err} was set to 0.995, which was used here as well [22]. In order to adapt to variable time delay, (6) was introduced. The coefficient δ_w is raised to the power of the relative angular difference between the current sample $\hat{\alpha}_t$ and its previous value $\hat{\alpha}_{t-1}$. Higher time delay results in lower sampling rate. Consequently, the exponent in (6) will become greater and the past measurements will have lower weight. The value of parameter δ_w has to be chosen with respect to the performance of the actual hardware used. Too low values would result in unstable behavior, while too high values cause slow adaptability to the changes in the environment. Value 0.99 was used here.

Step 6: Remove all adaptive neurons with weight w_i less than the established minimum weight threshold w_{min}. Threshold w_{min} has to be experimentally determined in order to obtain stable solution.

Step 7: If the number of sampled measurements is an integer multiple of heuristically determined parameter λ, insert a new neuron as follows (here $\lambda = 20$):

7.1 Find neuron n_q with the highest accumulated local error:

$$n_q = \{n_i : \max_i(error_i), i = 1,2,...,N\} \tag{7}$$

7.2 Insert new neuron n_r between neuron n_q and its neighbor n_f with the largest accumulated error. Set the distance d_r and angle α_r of the new neuron n_r as follows:

$$\alpha_r = \frac{\alpha_q + \alpha_f}{2} \tag{8}$$

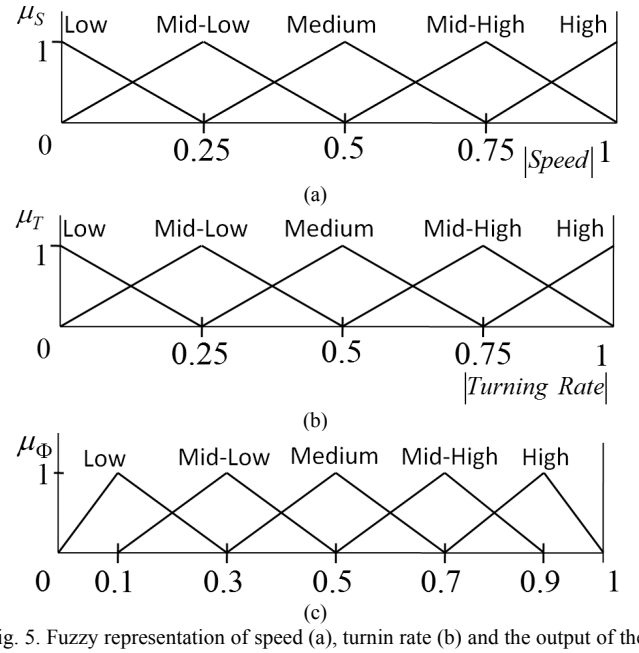


Fig. 5. Fuzzy representation of speed (a), turnin rate (b) and the output of the fuzzy controller (c).

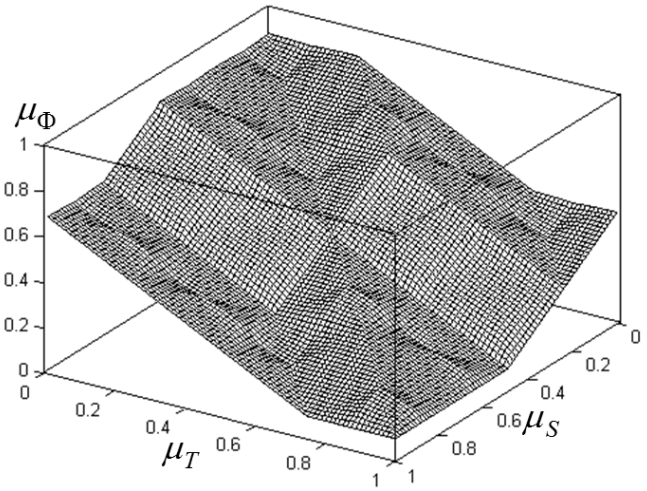


Fig. 6. Fuzzy control surface.

$$d_r = \sqrt{x_I^2 + y_I^2} \tag{9}$$

Here x_I and y_I denote the x and y coordinates of the intersection point of a line at angle α_r from the sonar sensor and the SOFAMap structure.

7.3 Multiply the local error variable $error_q$ of neuron n_q with parameter β to reduce the accumulated error. Initialize the local error $error_r$ of neuron n_r with the new value of error $error_q$.

$$error_{q,t} = error_{q,t-1}\ \beta \tag{10}$$

$$error_{r,t} = error_{q,t} \tag{11}$$

Here, parameter β was set to 0.5, following the original GNG implementation in [22].

7.4 Initialize the weight w_r of the new neuron n_r to the weight of neuron n_q.

$$w_r = w_q \tag{12}$$

Step 8: When new sonar measurement is acquired, go to step 1.

The value of Φ (3) follows the rate at which the surroundings of the robot changes. This rate is proportional to the speed and turning rate of the robot. The faster the surrounding of the robot changes, the less relevant the measurements obtained in the past are. On the other hand, if the robot remains stationary, past measurements remain relevant.

In the original GNG algorithm, this weight was fixed at the beginning of the algorithm and remained constant during the whole training process [22]. In the presented SOFAMap algorithm, the weight Φ (3) is calculated dynamically using a fuzzy controller [28]. The controller uses two input fuzzy representations of speed and turning rate to calculate the output out_Φ, as shown in Fig. 5. The implemented fuzzy controller is a Mamdani type controller with a multiplication product used as the t-norm operator. Fig. 6 shows the implemented fuzzy control surface. More weight is assigned to the turning rate as it affects the constructed view faster, compared to robot's speed. The output out_Φ is used to calculate the weight Φ (3):

$$\Phi = \left(out_\Phi\right)^k M \tag{13}$$

Here, parameter k controls the exponential trend of the weight Φ ($k > 1$) and M describes the relative importance of the previous measurements versus the new one. (13) was constructed in order to transform the output of the fuzzy controller into the appropriate adaptive learning rate (3). Here, parameters k and M were chosen as 2 and 4, respectively. Similarly to parameter δ_w, M should be chosen with respect to the actual hardware implementation, as improper values might result in unstable behavior.

High values of Φ (robot is stationary) result in only slow adjustment towards the new inputs. On the other hand, low values of Φ (robot is moving) cause a fast adaptation to the most recent measurements.

C. SOFAMap Algorithm Analysis

This section discusses the computational complexity of the SOFAMap algorithm and gives a proof of its convergence towards the boundary of the robot's surrounding.

Analysis of the computational complexity is fairly straightforward. Step 1 and 4 as well as the fuzzy controller in step 3 are computed in constant time. Searching for the nearest neuron in step 2, updating all neurons in step 5, and removing low weight neurons in step 6 can be performed in time *O(N)*, where *N* denotes the number of neurons in the structure. Step 7 is carried out once every λ iterations and it consist of finding the neuron with the highest error and then finding its neighbor with the highest error. This can be also performed in time *O(N)*. Hence, the computational complexity *T* can be expressed as:

$$T = \mathrm{O}(1)_1 + \mathrm{O}(N)_2 + \mathrm{O}(1)_3 + \mathrm{O}(1)_4 + \\ + \mathrm{O}(N)_5 + \mathrm{O}(N)_6 + \frac{1}{\lambda}\mathrm{O}(N)_7 = \mathrm{O}(N) \quad (14)$$

Therefore, a single update of the SOFAMap algorithm runs with a linear asymptotic complexity of *O(N)*, where *N* denotes the number of neurons in the structure.

During robot's motion the fuzzy controller adaptively reduces the weight of previous measurements in order to quickly adapt to the changing environment. As the robot becomes stationary, the SOFAMap structure starts adapting to the current environment. It can be shown that processing additional measurements reduces the error of the model, while the robot remains stationary.

Suppose that at time *t*, neuron *a* located at distance $r_{a,t}$ from the sensor, is being updated by measurement d_t. Its error $Err_{a,t}$ can be defined as the difference between distance $r_{a,t}$ and measurement d_t. Hence, the error $Err_{a,t}$ of neuron *a* at time *t* and the error $Err_{a,t-1}$ at time *t-1* can be defined as follows:

$$Err_{a,t} = \left| r_{a,t} - d_t \right| \qquad Err_{a,t-1} = \left| r_{a,t-1} - d_{t-1} \right| \quad (15)$$

It can be shown that $Err_{a,t} < Err_{a,t-1}$. By substituting (3) for $r_{a,t}$ in (15), the following inequality is formulated:

$$\left| \frac{\Phi\, r_{a,t-1} + d_t}{\Phi + 1} - d_t \right| < \left| r_{a,t-1} - d_{t-1} \right| \quad (16)$$

This can be simplified as follows:

$$\left| \frac{\Phi\, r_{a,t-1}}{\Phi + 1} + \frac{d_t}{\Phi + 1} - d_t \right| < \left| r_{a,t-1} - d_{t-1} \right| \quad (17)$$

$$\left| \frac{\Phi\, r_{a,t-1}}{\Phi + 1} - \frac{\Phi\, d_t}{\Phi + 1} \right| < \left| r_{a,t-1} - d_{t-1} \right| \quad (18)$$

$$\frac{\Phi}{\Phi + 1} \left| r_{a,t-1} - d_t \right| < \left| r_{a,t-1} - d_{t-1} \right| \quad (19)$$

Finally, since the robot is stationary, measurements d_t and d_{t-1} can be considered equal. Clearly, they might differ due to the fact, that single neuron *a* captures a whole interval of measurements at various angles around itself. However as more neurons are added into the stationary SOFAMap structure, this difference is minimized. Hence, (19), proves that the error of the SOFAMap structure is continuously minimized as the robot remains stationary.

III. Haptic Augmentation for Robot Teleoperation

For the mobile robot haptic teleoperation, two degrees of freedom of the control motion were considered: controlling the speed and the turning rate of the robot. By combining the speed and turning rate, a Cartesian planar control space was created. The control space is shown in Fig. 7. Here the point *(0, 0)* denotes the stationary state of the robot (zero speed and zero turning rate). This control space was restricted by the maximum speed and the maximum turning rate into a circular zone around the origin of the coordinate system. By moving the cursor in the control space, the operator can smoothly control the speed and the turning rate of the robot. As further shown in Fig. 7 the SOFAMap structure is projected onto the control space.

In the presented work, the amplitude F_M of the generated force feedback was modeled using a sigmoidal function. Sigmoidal function was chosen for its smooth shape and simple control mechanism. The force F_M is determined by the penetration depth Δ of the cursor position into the boundary of the view, which is defined by the SOFAMap algorithm:

$$F_M = \frac{1}{1 + e^{(-\kappa(\Delta - \varepsilon))}} \quad (20)$$

Here, constant κ determines the stiffness of the perceived penetration, constant ε sets the interval around the polyline, where the force is applied and the variable Δ denotes the penetration depth computed as:

$$\Delta = \sqrt{(x_C - x_I)^2 + (y_C - y_I)^2} \quad (21)$$

Here x_I and y_I denote the coordinates of the intersection point of a line connecting the cursor with the origin and the SOFAMap structure. Coordinates x_C and y_C define the position of the cursor. Fig. 8 illustrates the calculation of the penetration depth.

The computed force F_M is applied along the vector from the cursor to the origin of the control space as follows:

$$F_x = -\frac{F_M\ x_C}{\sqrt{x_C^2 + y_C^2}} \quad (22)$$

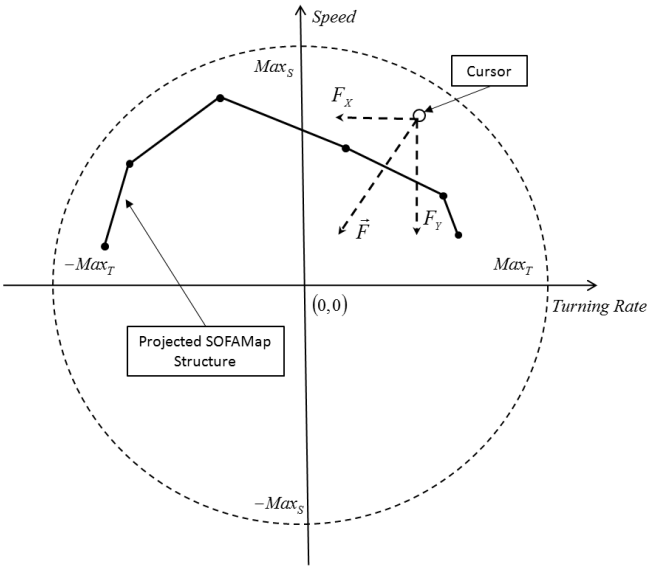


Fig. 7. The Cartesian control space with the projected SOFAMap structure.

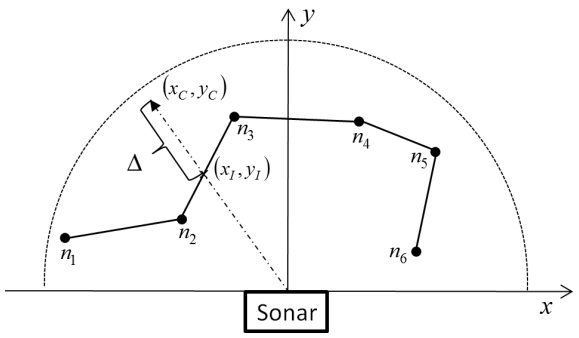


Fig. 8. Calculation of the penetration depth

$$F_y = -\frac{F_M \; y_C}{\sqrt{x_C^2 + y_C^2}} \tag{23}$$

In this way, the generated force-feedback provides a haptic sensation of obstacles in robot's view.

IV. Experimental Testing

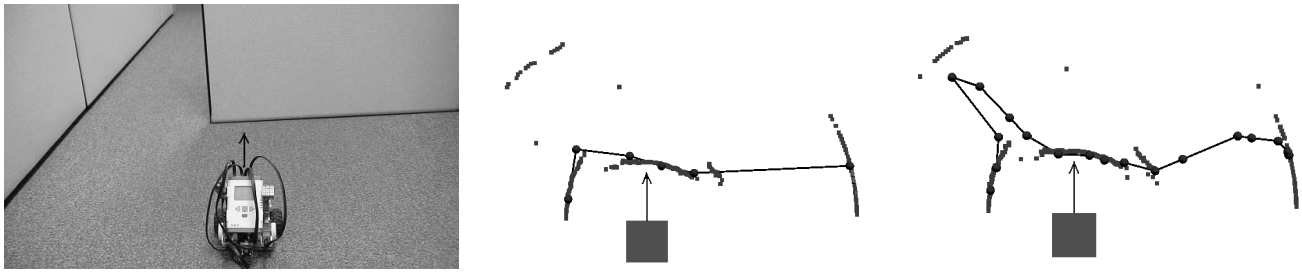
The SOFAMap algorithm was implemented in both virtual environment and on a real robotic platform. A multiple tests were conducted in order to demonstrate the performance and the behavior of the presented method.

A. Experimental Test Bed

The Lego NXT robotic platform was used as a scaled-down prototype version of industrially exploited mobile robots [29]. In addition, a virtual environment was implemented in order to simulate the mobile robot and the rotational sonar sensory system. In both cases, the system was interfaced via the Novint Falcon force-feedback haptic device [30], [31]. This control device comprises a force-controlled parallel delta robot configuration, which has three force-feedback and tactile sensation enabled degrees of freedom. Working frequency of 1kHz and position resolution of 400 dpi within the 4" x 4" x 4" workspace enables smooth control of the multi-robot system as well as fluent perception of the generated haptic force. The experimental test bed is shown in Fig. 9. The Lego NXT robot was connected via Bluetooth to an Intel Core 2 laptop at 2 GHz with 2GB RAM.

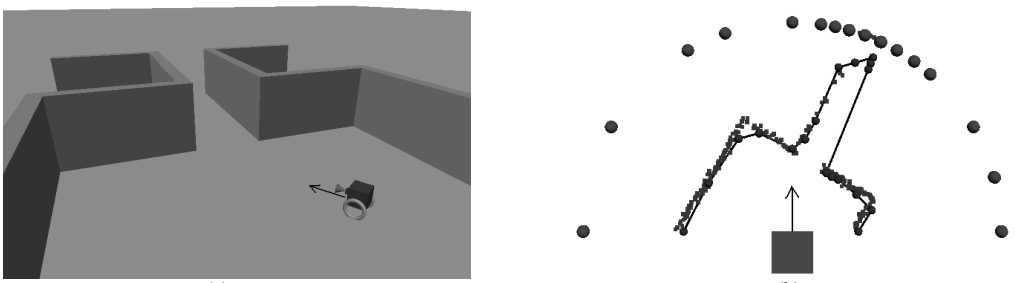
Fig. 9. The LegoNXT mobile robot with rotational sonar sensory system and the Novint Falcon force-feedback haptic interface.

The SOFAMap algorithm was developed first in the virtual environment using a simulated mobile robot and an approximate model of the rotational sonar sensory system. Consequently, the developed algorithm was implemented on the Lego NXT robotic platform to validate the proposed solution under the real world conditions. The advantages of the initial development in the virtual environment were as follows: 1) the ability to model an ideal sonar sensor, free of range and angular noise; 2) the ability to arbitrarily modify the simulated communication time delay and sampling rate of the



(a) (b) (c)

Fig. 10. The SOFAMap structure modeling the robot's environment (a) using only base neurons (b) and using adaptive neurons (c).



(a) (b)

Fig. 11. The simulated robot in the virtual environment (a) and the projected angular density of neurons for particular SOFAMap structure (b).

sensor; 3) the ease of setting and dynamically modifying system conditions and observing the behavior of the system.

B. Adaptive Environment Mapping

Firstly, the level of detail of the environment model constructed with the SOFAMap algorithm was investigated. The accurate model of robot's view is essential for providing the operator with a correct and high-fidelity haptic feedback.

Fig. 10(a)-10(b) show the robot in a cluttered environment and the appropriate constructed SOFAMap structure containing only the base neurons. The whole 180° view is covered with six evenly distributed base neurons. As demonstrated, the low-resolution structure gives the operator a basic notion about the environment. However, features such as the narrow corridor were not captured in the model.

Fig. 10(c) demonstrates the adaptation of the SOFAMap algorithm to the sampled sonar data. The structure adaptively self-organizes and a substantially more accurate model of the environment is constructed. It can be observed from Fig. 10(c) that the constructed SOFAMap structure now accurately reflects the narrow corridor.

The SOFAMap algorithm locally adapts the resolution of the model to optimize the level of detail with respect to the number of neurons in the structure. For example, simple elements, such as straight walls, require only low resolution of the constructed model, while more irregular elements such as corners or corridors require more neurons.

Fig. 11(a)–11(b) illustrates the simulated robot in the virtual environment and the adapted resolution of the SOFAMap structure, respectively. To visualize the angular density of neurons in the structure, the positions of particular neurons were projected onto a semi-circle around the sampled view. It is apparent that the resolution is adaptively increased around corners and corridors, whereas long and straight segments of walls do not require mapping of such a high fidelity.

C. Adaptation to Robot's Movement

The SOFAMap algorithm minimizes the error of the environment model by inserting new neurons into the structure and by removing old ones. Through this mechanism, the resolution of the model is being increased in areas where more neurons are needed for more precise description of the robot's surrounding.

New neurons are inserted into the structure at a constant rate of one neuron per λ iterations. However, the rate at which old neurons are removed is adaptively adjusted. Removing neurons at a faster rate than they are inserted, results in resolution decrease. In a similar manner, the resolution of the model will be increased when the frequency of removing old neurons is lower than the rate at which neurons are inserted.

The implemented fuzzy controller measures the speed and the turning rate of the robot. Its output value directly influences the rate of removing neurons from the structure by computing particular value of weight Φ (3). Fig. 12 plots the output value of the intelligent fuzzy controller versus the speed and the turning rate of the robot. Fig. 13 demonstrates the behavior of the resolution of the SOFAMap structure as the speed and turning rate of the robot dynamically changes. It can be observed that for low amplitude of robot's speed and

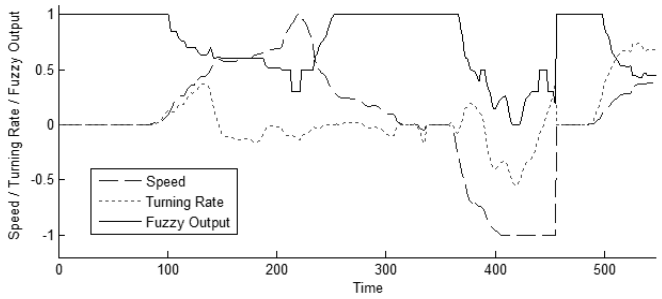


Fig. 12. The response of the fuzzy controller to the speed and turning rate of the robot.

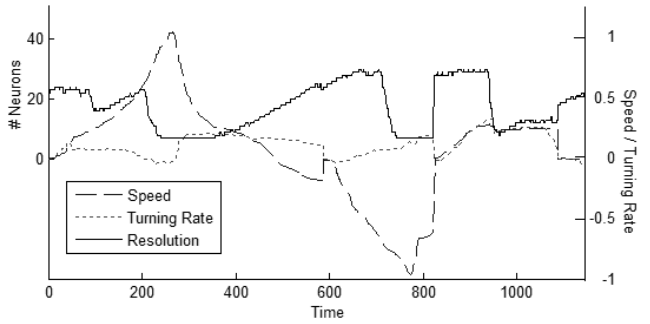


Fig. 13. The resolution of the constructed structure as a function of the speed and the turning rate of the robot.

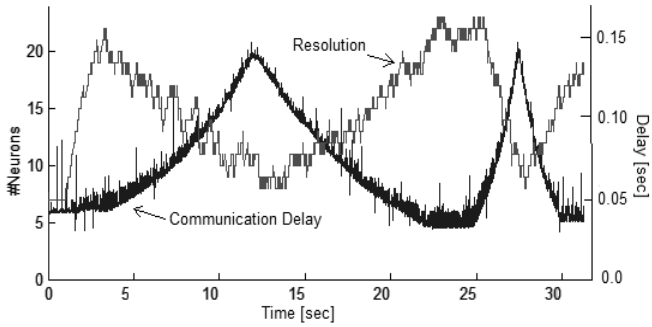


Fig. 14. The behavior of the resolution of the SOFAMap structure as a function of the communication time delay.

turning rate, the resolution of the SOFAMap structure is increasing until it converges to an optimal value, given the additional constraints of the system (e.g. actual time delay or sampling rate). At this point the rate of inserting new neurons becomes equal to the rate of removing old neurons from the structure. As the amplitude of the speed and the turning rate increase, the fuzzy controller starts calculating lower weights Φ for the adaptive neurons. Consequently, the rate of removing old neurons increases and overtakes the rate of adding new neurons. The resolution of the structure decreases until only base neurons are left in the structure.

D. Adaptation to Communication Time Delay

The communication time delay plays an important role in the SOFAMap algorithm. It directly influences the rate at which data can be sampled from the rotational sensory system. Especially during robot's motion, the communication delay limits the resolution of the sampled information. Consequently, the SOFAMap algorithm has to consider the rate of obtaining new sonar measurements and adjust its resolution accordingly. This is necessary in order to build the

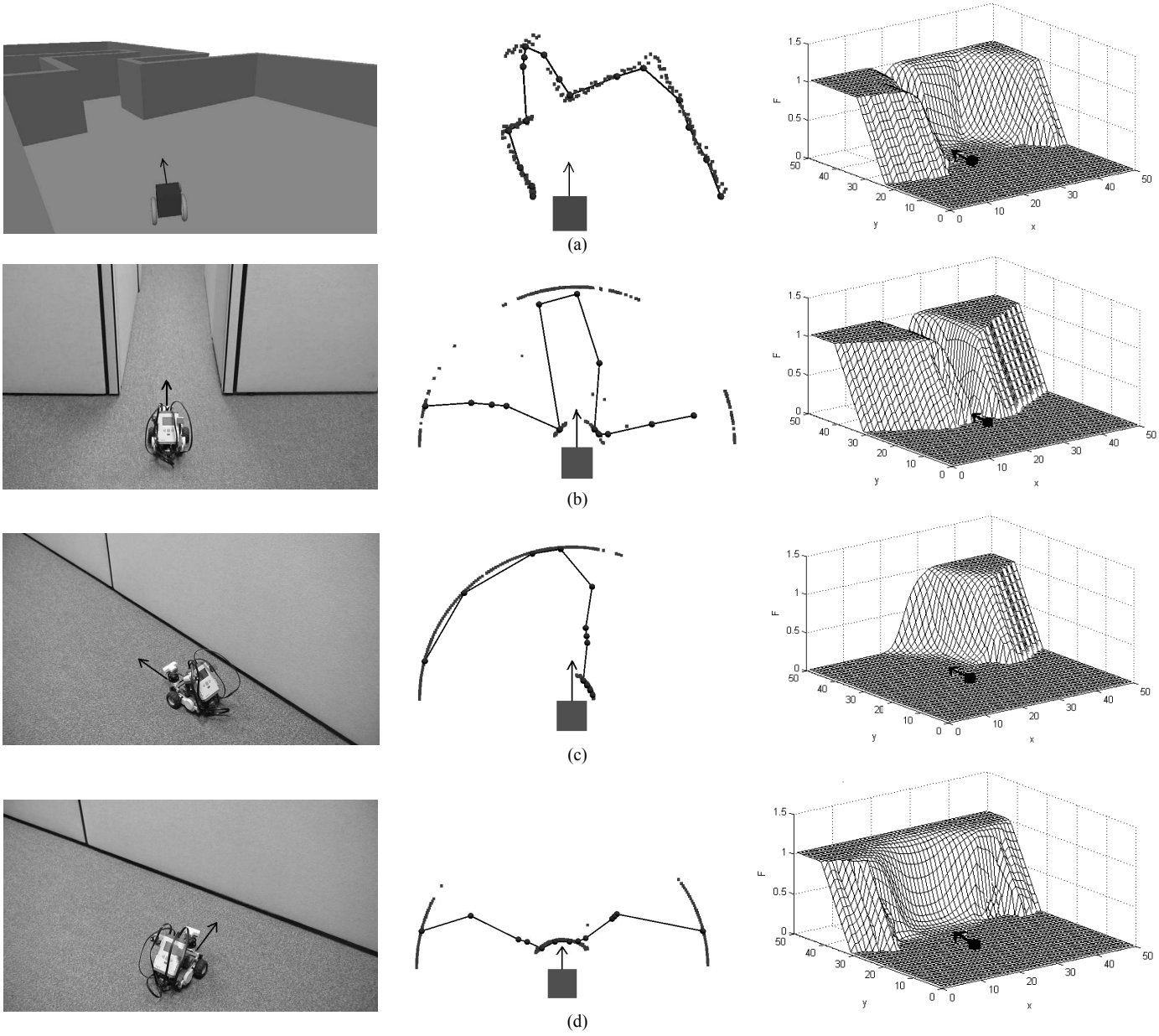


Fig. 15. The generated haptic force field for the virtual environment (a) and a real robot (b-d).

most accurate environment model with the highest level of detail possible.

Fig. 14 illustrates the behavior of the resolution of the SOFAMap structure as a function of the communication time delay. The variable time delay was simulated by adding a *pause* command to the program code. As the communication time delay increases, the resolution decreases and eventually only base neurons are left in the structure. When the communication delay is low and hence a lot of samples are obtained, the algorithm adaptively increases the resolution of the constructed environment model. This is a result of the relation between the sampling rate and the weight assigned to particular neurons established in (6).

E. Haptic Force-Feedback

A force field is calculated in order to demonstrate the haptic feedback delivered to the operator. The force field is a 3-dimensional plot, capturing the amplitude of the force for the whole planar control space. Fig. 15 displays the computed force fields along with the robot's environment and the constructed SOFAMap structure. From the examples in Fig. 15 it can be observed that the sigmoidal function provides a smooth increase of force as the operator is pushing further against the constructed boundary. Also, it is apparent that the generated force-feedback restricts the movement of the operator and protects hazardous maneuvering in unstructured environments. For instance, in Fig. 15(d) the speed of the robot is decreased when driving against a wall. Similarly, in Fig. 15(c) the operator is aware of the wall on the right side of the robot by sensing the haptic input. The generated force-feedback prevents turning against the obstacle.

V. CONCLUSION

This paper presented a novel self-organizing fuzzy based haptic teleoperation approach for mobile robots. The introduced SOFAMap algorithm was specifically developed

for a rotational sonar sensory system to alleviate some of the deficiencies of the traditionally used multi-sonar array. Despite utilizing only sparse sonar measurements, the operator is provided with a high fidelity haptic feedback. The SOFAMap algorithm also addresses the typical problems of teleoperation systems: sensor noise, variable communication time delay and low sampling rate. The resolution of the constructed model adapts to the irregularity of the environment. In addition, the resolution of the model also reflects the variable communication time delay and the actual sonar sampling rate. Furthermore, a built-in fuzzy controller is used to compensate for robot's motion.

The performance of the system was evaluated in a virtual environment as well as on a real robotic platform. The model of robot's view, constructed online by the SOFAMap algorithm, was shown to locally optimize its resolution with respect to the geometry of the robot's surroundings. Further, the visualization of the SOFAMap generated haptic force demonstrated the ability of the algorithm to accurately capture the unstructured environment. The ability of the system to adapt to a variable communication time delay and compensate for robot's motion was also illustrated.

ACKNOWLEDGMENT

The authors would like to thank the University of Idaho Nuclear Engineering Program for providing support for this project.

REFERENCES

[1] I. Petrovic, J. Babic, M. Budisic, "Teleoperation of Collaborative Mobile Robots with Force Feedback Over Internet," *Proc. of Intl' Conf. on Informatics in Control, Automation and Robotics, Robotics and Automation 2*, France, 2007.

[2] K. Sim, K. Byun, F. Harashima, "Internet-Baased Teleoperation on an Intelligent Robot With Optimal Two-Layer Fuzzy Controller," *IEEE Trans. Ind.Electron.*, vol. 53, no. 4, pp. 1362–1372, Aug. 2006.

[3] J. Kofman, X. Wu, T. Luu, S. Verma, "Teleoperation of a Remote Manipulator Using a Vision-Based Human-Robot Interface," *IEEE Trans. Ind.Electron.*, vol. 52, no. 5, pp. 1206–1219, Oct. 2005.

[4] T. Fong, Ch. Thorpe, Ch. Baur, "Multi-Robot Remote Driving With Collaborative Control." *IEEE Trans. Ind.Electron.*, vol. 50, no. 4, pp. 699–704, Aug. 2003.

[5] T. W. Fong, C. Thorpe, C. Baur, "Advanced interfaces for vehicle teleoperation: collaborative control, sensor fusion displays, and web-based tools," *Autonomous Robots*, July 2001.

[6] T. W. Fong,, F. Conti, S. Grange, C. Baur, "Novel Interfaces for Remote Driving: Gesture, Haptic, and PDA," *SPIE 4195-33, SPIE Telemanipulator and Telepresence Technologies VII*, November, 2000.

[7] T. Shimono, S. Katsura, K. Ohnishi, "Abstraction and Reproduction of Force Sensation From Real Environment by Bilateral Control," *IEEE Trans. Ind.Electron.*, vol. 54, no. 2, pp. 907–918, Apr. 2007.

[8] S. Lee, G. S. Sukhatme, G. J. Kim, C. M. Park, "Haptic control of a mobile robot: A user study," *Proc. of IEEE Intl' Conf. on Intelligent Robots and Systems*, pp. 2867-2874, 2002.

[9] N. Diolaiti and C. Melchiorri, "Teleoperation of a mobile robot through haptic feedback," in *Proc. of IEEE Intl' Workshop on Haptic Virtual Environments and Their Applications*, pp. 67–72, November 2002.

[10] H. Hashimoto, M. Saito, A. Sasaki, C. Ishii, M. Niitsuma, H. Hashimoto, "Hand Force Feedback System to Recognize Surrounding Objects for Safe Walking, " *Proc. Of IEEE/ASME Intl' Conf. on Advanced Intelligent Mechatronics*, pp. 307-312, July 2005.

[11] R. Oboe, "Web-Interfaced, Force-Reflecting Teleoperation System," *IEEE Trans. Ind.Electron.*, vol. 48, no. 6, pp. 1257–1265, Dec. 2001.

[12] L. Williams, R. Loftin, H. Aldridge, E. Leiss, W. Bluethmann, "Kinesthetic and Visual Force Display for Telerobotics," *Proc. IEEE Intl' Conf. on Robotics and Automation*, pp. 1249-1254 ,2002.

[13] L. Huijin, S.Aiguo, "Virtual-Environment Modeling and Correction for Force-Reflecting Teleoperation With Time Delay," *IEEE Trans. Ind.Electron.*, vol. 54, no. 2, pp. 1227–1233, Apr. 2007.

[14] G. Sankaranarayanan, B. Hannaford, "Experimental Comparison of Internet Haptic Collaboration with Time-Delay Compensation Techniques," *Proc. IEEE Intl' Conf. on Robotics and Automation*, pp. 206-211, 2008.

[15] C. Tzafestas, S. Velanas, G. Fakiridis, "Adaptive Impedance Control in Haptic Teleoperation to Improve Transparency under Time-Delay," *Proc. IEEE Intl' Conf. on Robotics and Automation*, pp. 212-219, 2008.

[16] I. Elhajj, N. Xi, W. K. Fung, Y. H. Liu, W. J. Li, T. Kaga, T. Fukuda, "Haptic Information in Internet-Based Teleoperation," *Proc. IEEE/ASME Trans. On Mechatronics*, vol. 6, no. 3, Sept. 2001.

[17] S. Okamoto, M. Konyo, S. Saga, S. Tadakoro, "Identification of Cutaneous Detection Thresholds against Time-delay Stimuli for Tactile Displays," *Proc. IEEE Intl' Conf. on Robotics and Automation*, pp. 220-225, May 2008.

[18] A. Kron, G. Schmidt, B. Petzold, M. I. Zah, E. Steinbach, P. Hinterseer, "Disposal of Explosive Ordnances by Use of a Bimanual Haptic Telepresence System," *Proc. of IEEE Intl' Conf. on Robotics and Automation*, pp. 1968-1973, April 2004.

[19] J. Casper, R. R. Murphy, "Human-Robot Interactions During the Robot-Assisted Urban Search and Rescue Response at the World Trade Center" *IEEE Trans. On Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 33, No. 3, pp. 367-385, June 2003.

[20] K. Kim, H. Lee, J. Park, M. Yang, "Robotic Contamination Cleaning System," *Proc. IEEE/RSJ Intl' Conf. on Intelligent Robots and Systems*, pp. 1874-1879, Oct. 2002.

[21] W. Echelmeyer, A. Kirchheim, E. Wellbrock, "Robotics-Logistics: Challenges for Automation of Logistic Processes," *Proc. of IEEE Intl' Conf. on Automation and Logistics*, pp. 2099-2103, Sept. 2008.

[22] B. Fritzke, "A growing neural gas network learns topologies," In G. Tesauro, D. D. Touretzky. & T. K. Leen (Eds.), *Advances in neural information processing systems*, MIT Press, Cambridge, pp. 625-632,1995.

[23] B. M. Wilamowski, N. J. Cotton, O. Kaynak, G. Dundar, "Computing Gradient Vector and Jacobian Matrix in Arbitrarily Connected Neural Networks," *IEEE Trans. on Industrial Electronics,* vol. 55, no. 10, pp. 3784-3790, Oct 2008.

[24] S. Thrun, "Robotic mapping: A survey," in G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

[25] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hähnel, M. Montemerlo, A. Morris, Z. Omohundro, C. Reverte, and W. Whittaker, "Autonomous exploration and mapping of abandoned mines," *IEEE Robotics and Automation*, 11(4), 2005.

[26] Ch.-Ch. Wang, Ch. Thorpe, M. Hebert, S. Thrun, H. Durrant-Whyte, "Simultaneous Localization, Mapping and Moving Object Tracking," in *International Journal of Robotics Research*, vol. 26, no. 6, June, 2007.

[27] T. Martinetz, K. Schulten, "A neural gas network learns topologies," in T. Kohonen, K. Makisara, O. Simula, J. Kangas, (Eds.), *Artificial Neural Networks*, North-Hollands, Amsterdam, pp. 397-402, 1991.

[28] E. Cox, *Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration*, Morgan Kaufmann Publishers, 2005

[29] LEGO MindStorms Webpage [URL], Available: http://mindstorms.lego.com/, from December 2008.

[30] Novint Falcon Webpage [URL], Available: http://home.novint.com/, from December 2008.

[31] C. I. Nichol, M. Manic, "Video Game Device Haptic Interface for Robotic Arc Welding," 2nd IEEE Conference on Human System Interaction, Catania, Italy, 21-23 May 2009.