Modeling of Ultrasonic Range Sensors for Localization of Autonomous Mobile Robots

Ricardo Gutierrez-Osuna, Jason A. Janet, *Student Member, IEEE,* and Ren C. Luo, *Fellow, IEEE*

***Abstract*— This paper presents a probabilistic model of ultrasonic range sensors using backpropagation neural networks trained on experimental data. The sensor model provides the probability of detecting mapped obstacles in the environment, given their position and orientation relative to the transducer. The detection probability can be used to compute the location of an autonomous vehicle from those obstacles that are more likely to be detected. The neural network model is more accurate than other existing approaches, since it captures the typical multilobal detection pattern of ultrasonic transducers. Since the network size is kept small, implementation of the model on a mobile robot can be efficient for real-time navigation. An example that demonstrates how the credence could be incorporated into the extended Kalman filter (EKF) and the numerical values of the final neural network weights are provided in the Appendixes.**

***Index Terms*— Kalman filtering, mobile robots, motion planning, neural networks, sonar navigation.**

I. Introduction

ULTRASONIC range sensors have gained extensive use in the mobile robot community. One particular transducer, the pulse-type ultrasonic rangefinder from Polaroid Corporation[1] [17], [18], has been adopted as a *de facto* standard for most mobile robot platforms. There are several reasons for its popularity: robustness, ease of use, range accuracy, light weight, and low cost [1]–[4], [11]–[16], [19]. However, this sensor presents some drawbacks. First, it has limited angular resolution. A detected target can reside anywhere inside a conical region with half angle of approximately ±15°. Second, when the target surface is oriented at unfavorable angles, the bulk of acoustic energy is reflected away from the transducer, and the target is undetected. Third, if the target is not isolated enough, energy can return from multiple surface reflections, creating the illusion of targets at erroneous distances. As a result of these issues, a probabilistic model is required to capture the behavior of the sensor.

Although the pressure distribution of the acoustic beam is best modeled with Bessel functions [2], most of the actual implementations on mobile robots assume simpler distributions (i.e., Gaussian [1], [11]) and ignore secondary lobes of detection. We propose a modeling technique based on a backpropagation neural network trained on experimental data. This approach has the advantage that it does not make any assumptions about the propagation characteristics of the acoustic signal. Instead, it relies on the nonlinear mapping capabilities of the backpropagation neural network to model the multilobal detection pattern.

II. Problem Formulation

A. Mobile Robot Localization Framework

While navigating through its environment, an autonomous mobile robot has access to two sources of information for localization purposes: dead reckoning and external sensors. Dead reckoning is the most straightforward method to infer the position and orientation of the vehicle. However, wheel drift, floor irregularities, and wheel encoder inaccuracies can accrue large errors over time. Therefore, like many of the methods of relative inertial localization, dead reckoning requires periodic updates and cannot be employed solely [3], [16]. Some kind of feedback from the environment is necessary. Traditionally, this has been accomplished with the use of dedicated active beacons (infrared, cameras, laser, etc.) or man-made landmarks (guide wires, reflective tape, barcodes, etc.). However, as proposed in [6], [7], [14], and [15], any mapped objects in the environment can be used as geometric beacons. These geometric beacons need to be detectable by the sensors, accurately mapped and within range frequently enough. Also, the sensors must be modeled, which means that it is possible to predict sensor readings from a map of the environment. Under these assumptions, the location of the vehicle can be updated by matching sensor observations against model predictions. As shown in Fig. 1, each observation-prediction pair is weighted by the confidence in the prediction and combined with the dead-reckoning information into a filter or control strategy, like the extended Kalman filter (EKF) used in [13]–[15].

B. Sensor Model

The key feature in the localization framework we have adopted for this research is the sensor model, which provides the ability to predict and understand sensor information. With the sensor model, the robot cannot only determine whether a geometric beacon is detectable or not, but also obtain probabilistic information about the quality of the beacon, allowing it to make intelligent decisions. We will focus our presentation

Manuscript received February 12, 1996; revised November 15, 1997. Abstract published on the Internet May 1, 1998.

The authors are with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695-7911 USA.

Publisher Item Identifier S 0278-0046(98)05688-3.

[1] Ultrasonic Ranging System, Polaroid Corporation, Cambridge, MA, 1982.

0278–0046/98$10.00 © 1998 IEEE

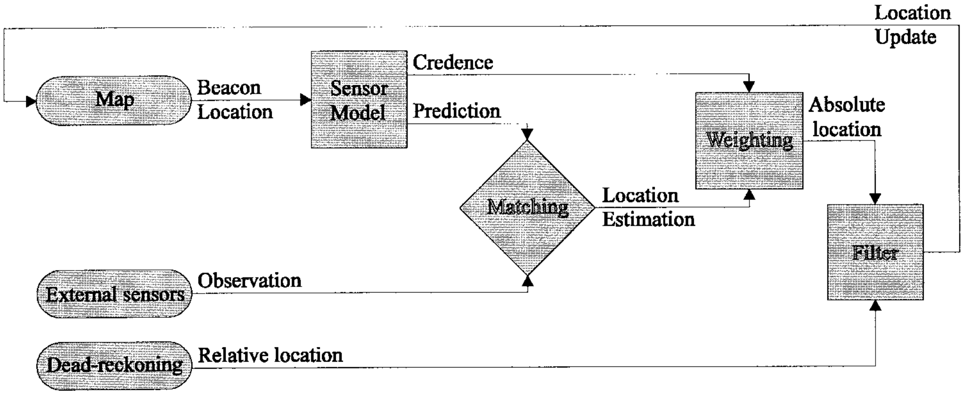


Fig. 1. Localization framework.

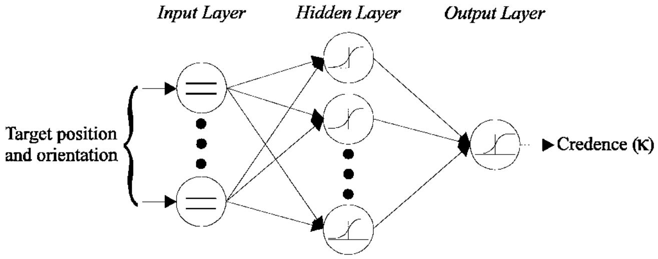


Fig. 2. Neural network model.

on the Polaroid ultrasonic rangefinder, due to its popularity in the mobile robot community. For this transducer, the pressure amplitude of the acoustic beam has the shape of a Bessel function,[2] but most of the actual implementations assume Gaussian distributions. Rather than making any assumptions about the acoustic beam pattern, we decide to train a neural network on experimental data obtained from an engineered environment. As shown in Fig. 2, the neural network used in this paper is the standard feedforward architecture with a single hidden layer, training with the backpropagation algorithm and adaptive learning rates [17]. Backpropagation neural networks are known for their nonlinear modeling capabilities in multidimensional spaces, which make them an excellent choice to model the detection probability distribution.

C. Neural Network Modeling

In order to train the neural network, the following steps must be taken.

1) Define the number of inputs and outputs of the network. We decide to rely exclusively on geometric information, such as the relative position and orientation of transducer and beacon.[3] These variables define a multidimensional input space. The output space in our case has only one dimension: detection probability of the beacon, or credence κ.
2) Obtain an experimental input–output mapping. This is accomplished by collecting a data set at discrete points in the input space and calculating their associated credence values.
3) Select a robust training strategy that reduces the effects of noise in the experimental data. We utilize a two-pass training process described as follows. In a first pass, the data set is split and the network is trained on one subset and tested on the other until the mean-square errors stabilize. The epoch with minimum test set error indicates the point where the network started to overfit (memorize the noise in the training set). The training set error at that epoch, properly scaled,[4] is used as a stopping point in a second pass in which the neural network is trained on the entire data set.

Fig. 3 illustrates an example of this two-pass training strategy. In the remaining sections of this paper, we apply these techniques to model the behavior of the transducer for two geometric beacons: 1) a bisecting flat surface and 2) a small flat surface.

III. Sonar Model for A Bisecting Surface

The situation in which the surface of the beacon bisects the acoustic beam is interesting, because it models the behavior of large walls, which can often be found during navigation. It also reduces the number of inputs to the network, since the relative position and orientation of the beacon can be defined with two variables. We choose the distance measured along the normal to the beacon surface (R) and angle between normals (θ), as depicted in Fig. 4.

A. Data Collection

Data collection was performed in our laboratory by marking the floor with a grid pattern similar to the one in Fig. 4. The minimum range for R was determined from the transducer's

[2] This is true for single-frequency excitation of the transducer. If more frequencies are added to the excitation signal, the distribution becomes more Gaussian shaped [2].

[3] The effects of other variables, such as temperature, or texture, are not considered here. These variables could be incorporated into the model as extra inputs, but rarely does a mobile robot have such information about its environment.

[4] The scaling factor is $(N_{\rm train} + N_{\rm test})/N_{\rm train}$, where $N_{\rm train}$ and $N_{\rm test}$ denote the number of examples in the training and test sets, respectively.

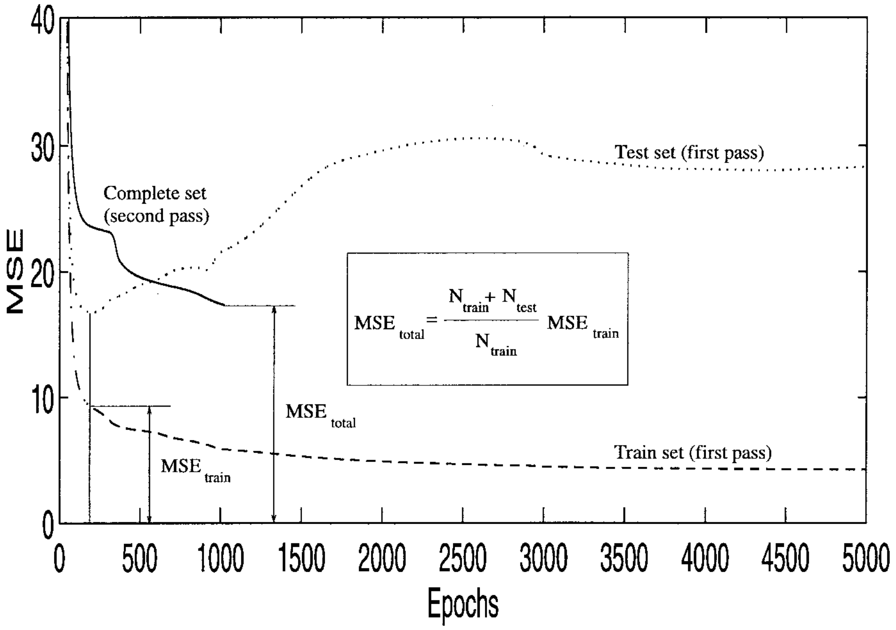
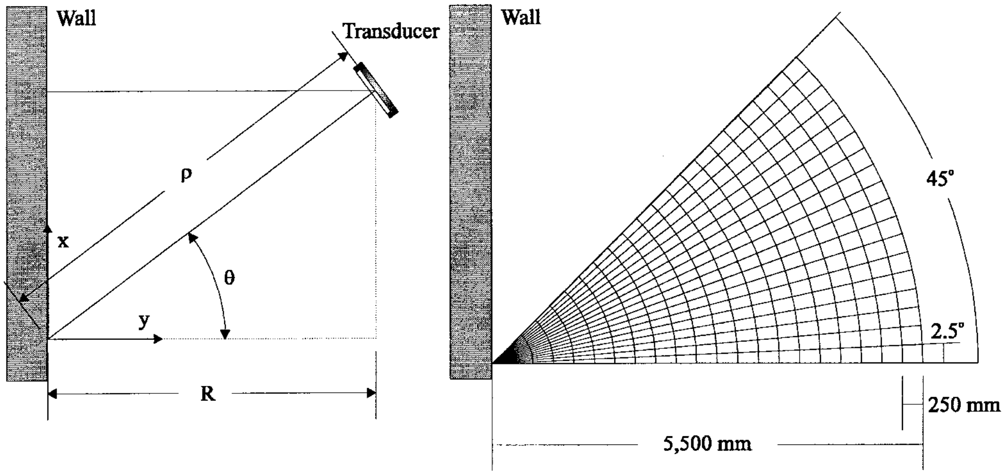


Fig. 3. Mean-square error evolution.



(a)

(b)

Fig. 4. (a) Reference frame and (b) data collection grid.

normal mode of operation,[5] while its maximum value was limited by the dimensions of the room. Only one half cone was collected, since the detection pattern can be assumed to be symmetric with respect to the transducer's normal. The maximum value for θ was 45°, well beyond the angular width of the transducer, to guarantee that all significant lobes of detection can be observed. Two data sets were collected: 1) a training set at the points defined in the grid of Fig. 4 and 2) a test set in an interleaved grid[6] (not shown in Fig. 4). At each position 100 readings were collected. Range and increments for the input variables are specified in Table I.

TABLE I
DATA COLLECTION RANGE AND INCREMENTS FOR A BISECTING SURFACE

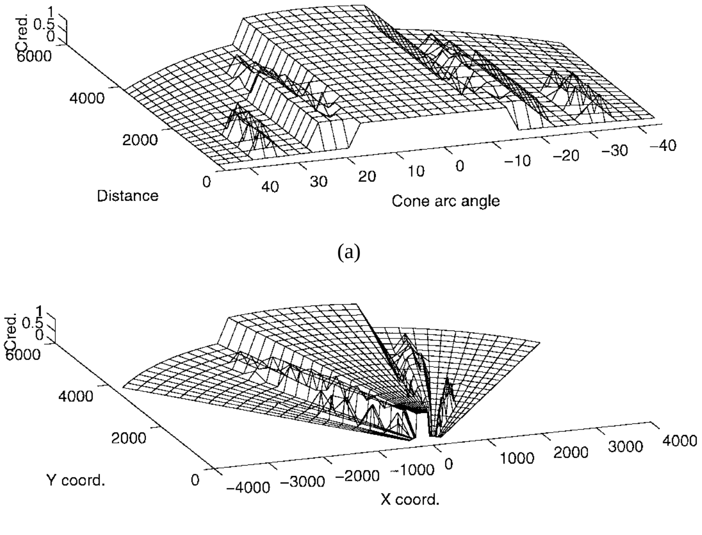
Data	# of	Distance ρ (mm)			Angle θ (°)		
Sets	points	Min	Max	Inc	Min	Max	Inc
Train	399	500	5500	250	0	45	2.5
Test	360	625	5375	250	1.5	43.75	2.5

B. Experimental Credence

The experimental credence at each location (R, θ) is obtained as the fraction of correct range readings over the total number of readings. Given that the relative position of transducer and wall is known, we can determine the correct range reading at each location. For this experiment, the correct reading corresponds to the closest point in the wall $(R = \rho \cos \theta)$, since an echo from that region will first return to the transducer. To account for inaccuracies in the grid and position of the transducer, we allow a 10% tolerance on both

[5] The minimum range could be further reduced by controlling the blanking time on the transducer's ranging board.

[6] The training set actually consists of four complete data sets from the grid in Fig. 4, collected in different days. The test set is made up of one data set on the interleaved grid. Room temperature was held constant at 23 °C for all sets.



(a)

(b)

Fig. 5. Experimental credence surface for the training set in (a) polar coordinates and (b) Cartesian coordinates.

sides of the assumed correct range. Only those readings that fall in this interval are labeled as correct. The experimental credence is shown in Fig. 5 as a surface in three-dimensional space (R, θ, κ). Three lobes of detection can be observed, as reported by the manufacturer, which clearly proves the multilobal detection pattern of the transducer.

C. Input Representation

The learning performance of a neural network is greatly influenced by the way input variables are represented. This fact led us to experiment with two types of input representations. First, we trained a network where each input variable was fed through a single input neuron. We called this method Representation 1 (REP1). Then we trained a second network, where each input variable was linearly decomposed into a number of input neurons (REP2). This representation was used to help the network discriminate the different detection lobes [17]. An example of both representations is shown in Table II.

We turn back to the experimental credence surface of Fig. 5 to determine the number of hidden neurons for the network with REP1. By looking at a half-cone pattern, we can see that the network will need at least five hidden neurons, one for each transition in detection along the θ dimension. Since overfitting is controlled with the two-pass training strategy, we can afford to double the number of hidden neurons[7] and guarantee that the network will be able to model the three detection lobes. The number of hidden units for REP2 is chosen to match the total number of weights on both networks, so that a fair comparison can be made between them. If we (somewhat arbitrarily) choose REP2 to map each input variable into ten input neurons, then the REP2 network must have two hidden neurons.[8]

TABLE II
INPUT REPRESENTATION EXAMPLE

Input[6]	**REP1**	**REP2 (5 input neurons)**				
	1 neuron	**#1**	**#2**	**#3**	**#4**	**#5**
0.15625	0.15625	0.375	0.625	0	0	0
0.5	0.5	0	0	1	0	0
0.875	0.875	0	0	0	0.5	0.5

D. Comparison of Results

The comparison of the two networks is based on two metrics: 1) speed of training and 2) generalization capabilities. REP1 trains more than three times faster than REP2. It also trains to a lower error. Both networks perform well and extract the three detection lobes. However, REP1 presents smoother surface contours, as shown in Fig. 6, which can be interpreted as better generalization capabilities. REP1 also executes faster when used in real-time navigation, since it can take raw input data, avoiding the linear decomposition step of REP2. For these reasons, we conclude that REP1 is more appropriate and will be the only one used in the second model presented in this paper.

IV. SONAR MODEL FOR A SMALL SURFACE

During navigation, a robot does not always encounter large flat surfaces for referencing. In many cases, smaller mapped obstacles can be used as geometric beacons, so the previous model (two inputs ρ and θ) cannot be applied. For this reason, we model the detection pattern for a small flat surface. A second angular variable, denoted by δ, appears in the relative orientation between transducer and beacon, as shown in Fig. 7. The size of the beacon must also be taken into account. Rather than including different sizes and shapes as extra input variables, which would complicate data collection and increase the size of the neural network, we have presented in [6] and [7] an algorithm that efficiently searches for geometric beacons in the map and decomposes their front surfaces into fixed-size subsurfaces.[9]

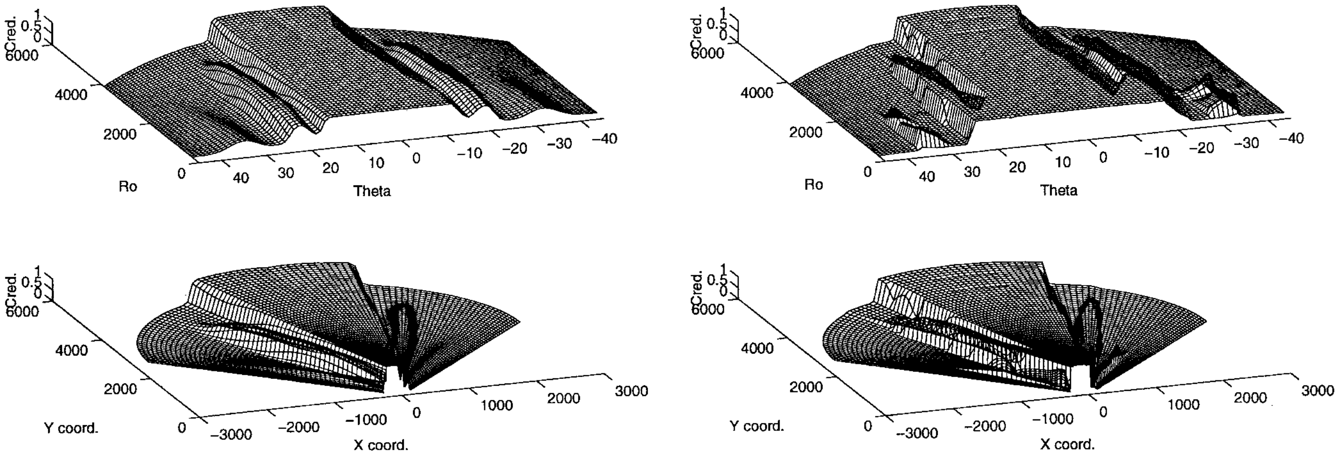
A. Data Collection

In order to capture the detection pattern of a small flat reflector, it must be isolated from its environment. In particular, two different sources of echo must be eliminated: 1) supporting structure and 2) edges of the reflector. We utilize two inexpensive stands for transducer and reflector, as shown in Fig. 7. Both transducer and reflector are mounted on protractors, to allow for the two angular degrees of freedom. The reflector is large enough to occlude the protractor and the vertical section of the stand. Echoes from the vertical section of the stand can be identified, since the nominal distance between transducer and reflector is known during data collection, and the stand is 0.5 m behind the reflector. The reflector is circular

[7] The risk of overfitting will increase with the number of hidden neurons, because the network will have extra degrees of freedom to make more complex mappings [17], [20].

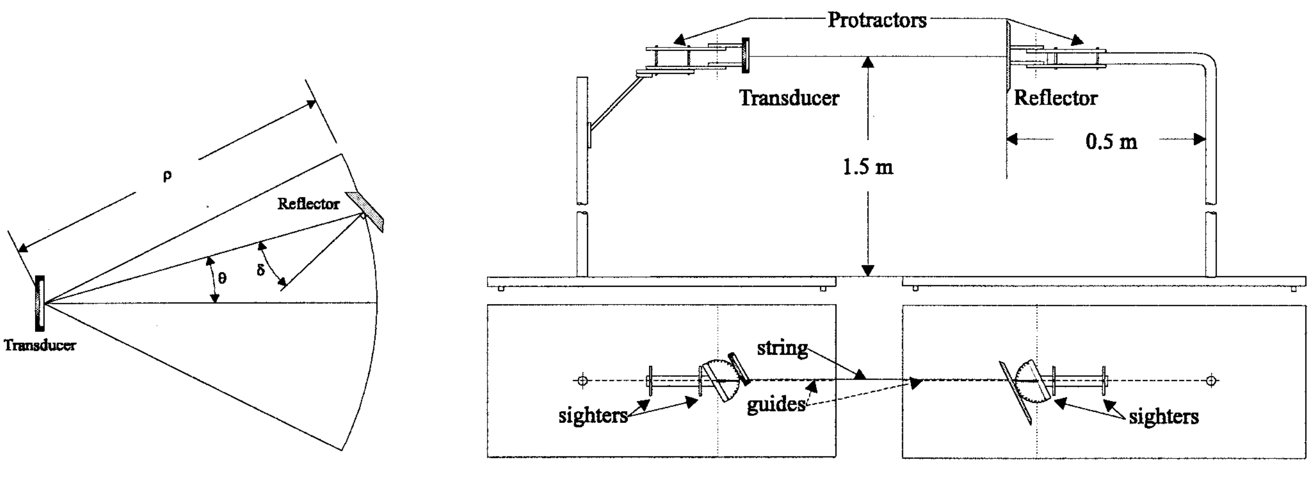
[8] The REP1 network has 2-10-1 neurons (41 weights, including biases). The REP2 network has 20-2-1 neurons (45 weights).

[9] The implicit assumption made here is that the detectability of a surface can be determined by decomposing it into fixed-size subsurfaces and calculating the detection probability of the individual subsurfaces.



(a) (b)

Fig. 6. Final network mappings for (a) REP1 and (b) REP2.



(a) (b)

Fig. 7. (a) Reference frame. (b) Mounting stands side and top views.

in shape, 10 cm in diameter, with edges beveled at 45° to attenuate the echoes diffracted from the edges.[10]

Range and increments for the three input variables are shown in Table III. Since we have an extra variable, the number of position and orientation combinations is much larger than in the first experiment, and only one data set is collected.[11] Room temperature is held constant at 23 °C. Unlike in the first experiment, data collection covers positive and negative ranges of the angular variables. The reason for this is that alignment of the stands has an accuracy of ±3°. By collecting data from the four quadrants in the (θ, δ) plane (see Fig. 10) we can find the centroid of the detection pattern, shift the pattern to the origin of coordinates, and average data from the four quadrants.[12]

TABLE III
Input Variable Range and Increments for Small Beacon

	Min.	Max.	Incr.
Cone penetration (θ °)	-40	40	2
Cone arc deviation (δ °)	-40	40	2
Distance (ρ mm)	500	5,500	250

B. Experimental Credence

The experimental credence is obtained as in the first experiment. A range reading is labeled as correct if it falls within ±10% of the nominal distance between transducer and reflector. The credence for this model can be viewed as a temperature in three-dimensional input space (ρ, θ, δ). Since

[10] The circular shape of the reflector serves two purposes. First, since it does not have vertical edges, it reduces the possibility of detecting edges from the edges of the reflector. Second, it makes the reflector invariant to rotations with respect to the axis normal to the reflector.

[11] The data set is divided into a train set and a test set. Data collected at distances (500, 1000, $\cdots$, 5500) is used for the train set, and data collected at distances (750, 1250, $\cdots$, 5250) is used for the test set.

[12] We assume that the detection pattern is symmetric with respect to $\theta = 0°$ and $\delta = 0°$, so we are modeling just one quadrant.

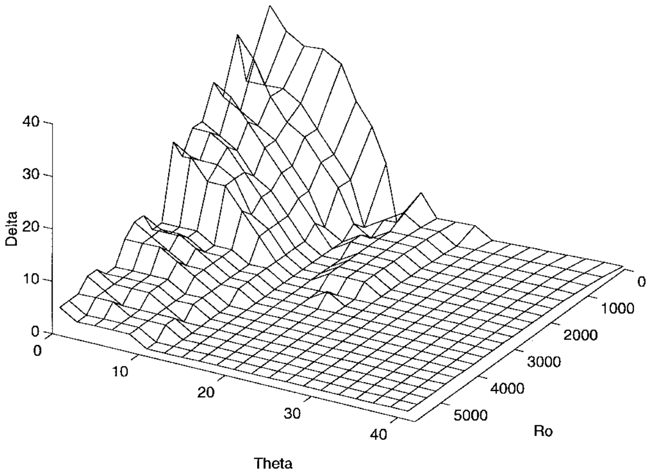


Fig. 8. Experimental credence for finite surface with $\kappa = 0.9$.

we have three dimensions plus the credence, one of these variables must be fixed to plot the experimental credence.

Fig. 8 presents the experimental results for a credence value fixed at $\kappa = 0.9$. The surface represents the boundary of 90% detection probability. From the plot, we can observe that the third lobe has disappeared,[13] and that there is more noise, especially in the distance axis.

C. Neural Network Training Results

We concluded in the first experiment that REP2 was more sensitive to noise in the data set. Based on this fact, together with the noisy data collection results shown in Fig. 8, we decide to focus on REP1 for this model. Two networks, with five and ten hidden units, were trained. The training results using the format of Fig. 8 with $\kappa = 0.1$ and $\kappa = 0.9$ are presented in Fig. 9. The network with five hidden units performed better noise reduction, but was too small to isolate the secondary lobe from the main lobe. So, we conclude that the network with ten hidden units is more appropriate. An alternative plot of this model is shown in Fig. 10, where the credence volume has been sliced at $\rho = 1500$ mm, and the first quadrant in the (θ, δ) plane has been mirrored.

V. Conclusions

We have presented a new modeling methodology for ultrasonic sensors, where the variable of interest is the probability of detecting an obstacle in the environment, given its position and orientation relative to the transducer. This model can help a robot choose the most reliable geometric beacons for localization. Two experiments were performed, which proved the multilobal detection pattern of the transducer. By modeling the different lobes, our model is more accurate than frequently used Gaussian distributions. Also, since the size of the neural network is kept small, the model can be implemented on a robot navigating in real time. The effect of *texture* has not been considered in this paper. It may be incorporated as an extra input to the neural network, but rarely does a mobile robot have information about the different textures of the objects in its environment, not to mention the effort that would be required in order to collect data for different textures. Therefore, our models cannot be generalized to all textures. However, we chose specular surfaces for both wall and reflector to present a worst case scenario. Any other surface with more texture will have a higher probability of being detected, so our model is a conservative estimate of the credence. The *size* of the reflector is factored into the search procedure we have proposed in [6] and [7], which decomposes the environment (larger size beacons) into subsurfaces or patches that have the size of the reflector used by the model.

Appendix I
Final Neural Network Weights

We include in Table IV the final weights for the architecture that provides the best results at modeling the credence κ. The reader may construct a 3:10:1 neural network, so as to duplicate the results and/or use this model for self localization. The inputs are ordered as (ρ, θ, δ) and should be normalized between 0 and 1 according to the following limits: ρ between 0–6000 mm, θ between 0°–50°, and δ between 0°–50°. Hidden and output neurons use the sigmoidal activation function $1/1 + \exp(\Sigma_{i=1}^{N} w_i x_i + w_{\text{bias}})$.

Appendix II
Incorporating Credence in the Kalman Filter

This Appendix demonstrates how the credence factor could be incorporated into a tracking algorithm like the Kalman filter. Specifically, we use the EKF to perform low-level self localization as the mobile robot traverses a known, geometrically described environment. Collecting geometric beacons along the way and decomposing them into visible segmented surfaces [6], [7], we can predict a range $\rho_{\text{calc},i}$ for each sonar $i = 1, 2, \cdots, N$. Based on previous state estimates and the position/orientation of each sonar with respect to the mobile robot, we can also predict surface orientation and which region of the sonar cone each surface segment is located. Hence, we can *predict* the variables $\rho, \theta,$ and δ. Consequently, we can also predict the credence κ. Of course, the typical EKF model, as applied to mobile robots, compares range sensor readings solely on the basis of the calculated range ρ_{calc} and the observed range ρ_{obs}. In the following paragraphs, we will put this all in the context of the EKF model, so that the reader can factor κ into the EKF innovation.

Borrowing notation from [8], we intend to estimate the state of an autonomous mobile robot from noisy sensor data (like that of sonars). To find the minimum mean-squared error (MMSE) for the vehicle's state $x(k)$, we use data from the dynamic and observation models for the vehicle and its sonars. The vehicle's state vector $x(k)$ changes according to the discrete-time nonlinear transition

$$x(k+1) = f[x(k), u(k+1), k+1] + v(k) \qquad (1)$$

[13] The beacon is made of Plexiglass, which is smoother than a regular wall. Also, the returning echoes are weake, r since the surface is smaller. Together, they explain why the third lobe has vanished.

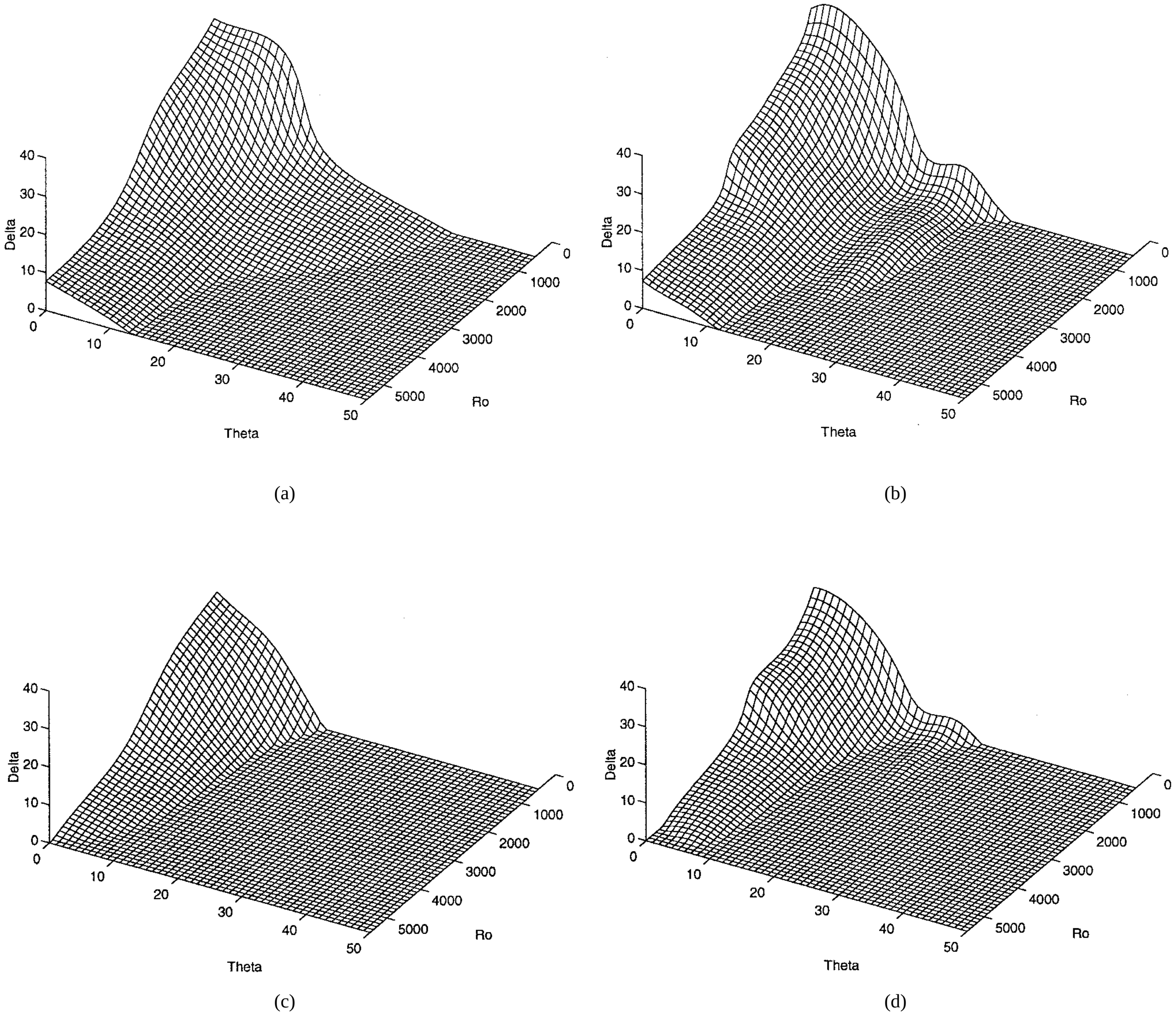


Fig. 9. Credence volume boundary. (a) Five hidden units with $\kappa = 0.1$. (b) Ten hidden units with $\kappa = 0.1$. (c) Five hidden units with $\kappa = 0.9$. (d) Ten hidden units with $\kappa = 0.9$.

TABLE IV
FINAL WEIGHTS OF THE 3:10:1 NEURAL NETWORK

hidden neuron #	*input-hidden weight*			*hidden bias*	*hidden-output weight*
	ρ	θ	δ		
1	-24.035341	-0.177473	1.909205	0.449708	15.596516
2	-3.367232	0.315339	17.663235	1.097636	10.005672
3	-16.772976	15.253121	5.930555	0.089570	-7.889393
4	-0.160200	-8.976890	31.707226	-11.428481	11.137950
5	-2.843503	17.543863	15.631051	-1.140790	-10.959534
6	5.257276	4.748250	9.610018	-6.720143	-12.753185
7	1.437722	1.532246	1.345657	-1.450179	0.623121
8	12.616241	-3.659353	11.396241	-4.321978	-14.463370
9	3.591183	3.271416	-1.302033	-7.067067	-2.778177
10	0.373195	9.242279	12.127007	-5.400525	-9.808138
output bias: 16.351728					

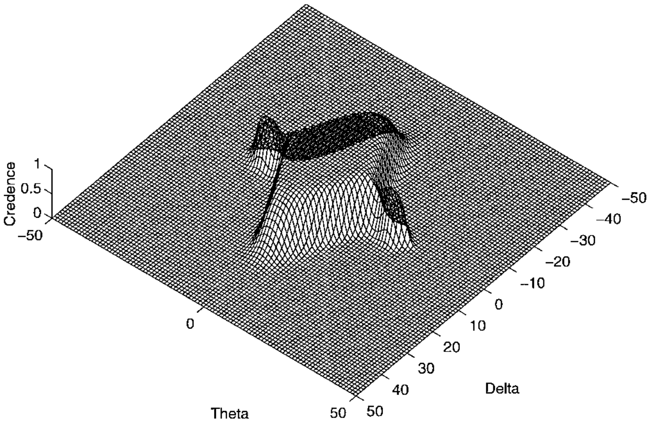


Fig. 10. Slice of credence volume at $\rho = 1500$ mm.

where $f[x(k), u(k+1), k+1]$ is the process model, $x(k)$ is the vehicle's state at time step $k, u(k+1)$ is the input vector, and $v(k)$ is random additive noise. The set of noisy sonar observations $z(k+1) = \{\rho_{1\text{obs}}, \rho_{2\text{obs}}, \cdots, \rho_{N\text{obs}}\}$ are related to $x(k)$ by the nonlinear equation

$$z(k+1) = h[x(k), u(k+1), k+1] + w(k) \tag{2}$$

where $z(k+1)$ is the observation vector, $h[x(k), u(k+1), k+1]$ is a nonlinear transformation function that converts from state space to observation space, and $w(k)$ is random additive noise. We must assume that $v(k)$ and $w(k)$ are Gaussian (with zero mean) and uncorrelated for all k. Hence, their respective covariances are[14]

$$E[v(i)v^T(j)] = \delta_{ij}Q(i) \tag{3}$$

$$E[w(i)w^T(j)] = \delta_{ij}R(i) \tag{4}$$

with cross covariance

$$E[v(i)w^T(j)] = 0 \quad \forall i, j. \tag{5}$$

Letting $\hat{x}(i|j)$ be the conditional mean at time i conditioned on all observations up to time j,

$$x(i|j) = E[x(i)|Z^j] \tag{6}$$

where $Z^j = \{z(1), z(2), \cdots, z(j)\}^T$. The estimated covariance is

$$P(i|j) = E[\{x(i) - \hat{x}(i|j)\}\{x(i) - \hat{x}(i|j)\}^T|Z^j]. \tag{7}$$

There must be a prediction of the observation $z(k+1|k)$ at time $k+1$ conditioned on previous observations and state estimates. The conditional mean of the predicted observation is $\hat{z}(k+1|k)$ with covariance $P_{zz}(k+1|k)$. The linear update equations are

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k)v(k) \tag{8}$$

$$P(k+1|k+1) = P(k+1|k) - W(k+1)P_{vv}(k+1|k) \cdot W^T(k+1). \tag{9}$$

$W(k+1)$ is the *Kalman gain.*

The vector $v(k+1)$ is the *innovation*. Simply stated, the innovation is the difference between the calculated ranges and observed ranges. Typically, the innovation is a vector comprised of the element-by-element calculated and observed range differences. However, we consider $v(k+1)$ to be dependent on how reliable each surface segment is to its respective sonar:

$$v(k+1) = [z(k+1) - \hat{z}(k+1|k)] \circ \overline{\kappa}(k+1). \tag{10}$$

The operand $\circ$ denotes the element-by-element product of two vectors. This guarantees that we maintain proper dimensionality for (8) and (9). Here is where we include the credence factor and, thus, append the standard Kalman filter model. Inclusion of credence κ also impacts the second-order statistics. Specifically, the innovation covariance is

$$P_{vv}(k+1|k) = P_{zz}(k+1|k) + R(k+1). \tag{11}$$

Now, we can compute the Kalman gain with

$$W(k+1) = P_{xz}(k+1|k)P_{vv}^{-1}(k+1|k) \tag{12}$$

where $P_{xz}(k+1|k)$ is the predicted cross-correlation matrix between $\hat{x}(k+1|k)$ and $\hat{z}(k+1|k)$.

[14] δ_{ij} in (3) and (4) refers to the function $\delta_{ij} = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{otherwise} \end{cases}$, not the cone arc deviation.

REFERENCES

[1] B. Barshan and R. Kuc, "Active sonar for obstacle localization using envelope shape information," in *Proc. 1991 Int. Conf. Acoustics, Speech, and Signal Processing*, 1991, pp. 1273–1276.

[2] O. Bozma and R, Kuc, "Building a sonar map in a specular environment using a single mobile sensor," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 1260–1269, Dec. 1991.

[3] I. J. Cox and J. J. Leonard, "Unsupervised learning for mobile robot navigation using probabilistic data association," in *Computational Learning Theory and Natural Learning Systems, Volume II: Intersections Between Theory and Experiment*, S. Hanson T. Petsche, M. Kearns, and R. Rivest, Eds. Cambridge, MA: MIT Press, 1994, pp. 297–319.

[4] M. Drumheller, "Mobile robot localization using sonar," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, pp. 325–332, Feb. 1987.

[5] A. Elfes, "Sonar-based real-world mapping and navigation," *J. Robot. Automat.*, vol. 3, no. 3, pp. 249–265, 1987.

[6] J. A. Janet, R. C. Luo, and M. G. Kay, "Autonomous mobile robot global motion planning and geometric beacon collection using traversability vectors," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 132–140, Feb. 1997.

[7] J. A. Janet, R. Gutierrez-Osuna, M. G. Kay, and R. C. Luo, "Autonomous mobile robot self-referencing with sensor windows and neural networks," in *Conf. Rec. IECON'95*, Orlando, FL, 1995, pp. 1124–1129.

[8] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," presented at the American Automatic Control Conf., Seattle, WA, June 1995.

[9] R. Kuc and M. W. Siegel, "Efficient representation of reflecting structures for a sonar navigation model," in *Conf. Rec. 1987 IEEE Int. Conf. Robotics and Automation*, 1987, pp. 1916–1923.

[10] ——, "Physically based simulation model for acoustic sensor robot navigation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, pp. 766–778, June 1987.

[11] R. Kuc, "A spatial sampling criterion for sonar obstacle detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 686–690, July 1990.

[12] J. Leonard and H. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*. Norwell, MA: Kluwer, 1992.

[13] J. J. Leonard and H. F. Durrant-White, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 376–382, June 1991.

[14] ——, "Simultaneous map building and localization for an autonomous mobile robot," in *Proc. IEEE/RSJ Int. Workshop Intelligent Robots and Systems (IROS'91)*, 1991, pp. 1442–1447.

[15] J. Manyika and H. Durrant-Whyte, *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*. New York: Ellis Horwood, 1994.

[16] L. Marce, "Absolute location of a robot in polyhedral and nonpolyhedral environments," in *Conf. Rec. Control '85*, 1985, pp. 325–332.

[17] M. Smith, *Neural Networks for Statistical Modeling*. New York: Van Nostrand Reinhold, 1993.

[18] D. Wilkes, G. Dudek, M. Jenkin, and E. Milios, E., "Ray-following model of sonar range sensing," in *Proc. Mobile Robots V*, 1990, pp. 536–542.

[19] A. Zelinski, "Mobile robot map making using sonar," *J. Robot. Syst.*, vol. 8, no. 5, pp. 557–577, 1991.

[20] J. M. Zurada, *Introduction to Artificial Neural Systems*. St. Paul, MN: West, 1992.

Ricardo Gutierrez-Osuna received the B.S. degree in industrial engineering in 1992 from the Polytechnic University of Madrid, Madrid, Spain, and the M.S. degree in computer engineering in 1995 from North Carolina State University, Raleigh, where he is currently working towards the Ph.D. degree in computer engineering.

His research in mobile robotics includes probabilistic navigation using Markov models, sonar mapping with certainty grids, self-organizing feature maps and feedforward neural networks, and obstacle avoidance with potential fields. He is currently working on the development of instrumentation hardware and pattern recognition architectures for a gas sensor array test bed, an "electronic nose" used for environmental and quality control applications. His research interests are mobile robotics, probabilistic reasoning, and pattern recognition.

Jason A. Janet (S'93) received the B.S. degree in mechanical engineering in 1990 from the University of Virginia, Charlottesville, and the M.S. degree in integrated manufacturing systems engineering and the Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, in 1994 and 1998, respectively.

He is currently with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, where he teaches courses in distributed real-time controls, space robotics, and rapid prototyping of modular robotic arms and legs, as well as conducting several projects involving neural networks, pattern analysis, tracking, autonomous mobile robots, electromechanical design, and controls. His research is primarily in the area of pattern analysis and controls for autonomous vehicles using neural networks.

Ren C. Luo (M'83–SM'88–F'92) received the Diplom-Ing. and Ph.D. degrees from Technische Universitaet Berlin, Berlin, Germany, in 1979 and 1982, respectively.

He is currently a Professor in the Department of Electrical and Computer Engineering and the Director of the Center for Robotics and Intelligent Machines, North Carolina State University, Raleigh. From 1992 to 1993, he was the Toshiba Endowed Chair Professor in the Institute of Industrial Science, University of Tokyo.