Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Neural network based target differentiation using sonar for robotics applications

Billur Barshan and Birsel Ayrulu
Department of Electrical Engineering
Bilkent University
Bilkent, 06533 Ankara, Turkey

Simukai W. Utete
Robotics Research Group
University of Oxford
Oxford, OX1 3PJ, U.K.

Abstract

This study investigates the processing of sonar signals using neural networks for robust differentiation of commonly encountered features in indoor environments. The neural network can differentiate more targets, and achieves high differentiation and localization accuracy, improving on previously reported methods. It achieves this by exploiting the identifying features in the differential amplitude and time-of-flight characteristics of these targets. An important observation follows from the robustness tests, which indicate that the amplitude information is more crucial than time-of-flight for reliable operation. The study suggests wider use of neural networks and amplitude information in sonar-based mobile robotics.

1 Introduction

Neural networks have been employed efficiently as pattern classifiers in numerous applications [5]. These classifiers make weaker assumptions on the shape of the underlying distributions of input data than traditional statistical classifiers and can prove more robust when the underlying statistics are unknown or the data is generated by a nonlinear system.

Sonar is a very useful and cost-effective mode of sensing for mobile robots. This paper investigates the use of neural networks to process sonar signals encountered in target differentiation and localization applications for indoor environments. The pattern recognition capability of neural networks allows differentiation of more targets with increased accuracy by exploiting the identifying features in the differential amplitude and time-of-flight characteristics of the reflected signals. The robustness of the network performance to partial removal of the input information has been investigated, demonstrating that the network is robust to different failure modes, and indicating that the amplitude information is more crucial than time-of-flight for reliable target differentiation and localization. A comparison with previously investigated approaches indicates improved performance.

The most common sonar ranging system is based on *time-of-flight* (TOF) which is the time elapsed between transmission and reception of a pulse. Differential TOF models of targets have been used by several researchers [8, 9]. Systems combining amplitude, energy, and duration of the echo signals along with TOF information [1, 2, 4], or exploiting the complete echo waveform [6] have also been considered.

Accurate target classification can be achieved by using sonar systems employing both amplitude and TOF information. In the present paper, neural networks are used to process amplitude and TOF information so as to reliably handle the target classification problem.

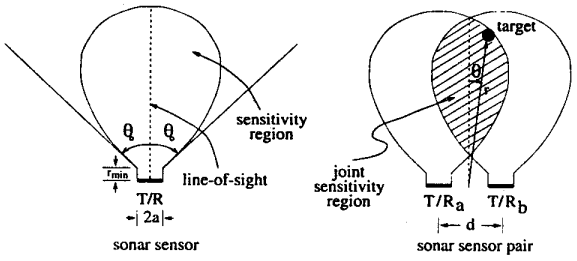
2 Background on Sonar Sensing

In the commonly used TOF systems, an echo is produced when the transmitted pulse encounters an object and a range value $r = ct_o/2$ is produced when the echo amplitude first exceeds a preset threshold level τ at time t_o. Here, t_o is the TOF and c is the speed of sound in air (at room temperature, $c = 343.3$ m/s.).

In general, it is observed that the echo amplitude decreases with increasing target range and azimuth θ (Fig.1(a)). The echo amplitude falls below τ when $|\theta| > \theta_o$, which is related to the aperture radius a and the resonance frequency f_o of the transducer by $\theta_o = \sin^{-1}\left(\frac{0.61c}{af_o}\right)$ [17].

With a single stationary transducer, it is not possible to estimate the azimuth of a target with better resolution than $2\theta_o$. In our system, two identical ultrasonic transducers a and b with center-to-center separation d are employed to improve the angular resolution. Each transducer can operate both as transmitter and

receiver and detect echo signals reflected from targets within its *sensitivity region* (Fig.1(a)). Both members of the sensor configuration can detect targets located within the *joint sensitivity region*, which is the overlap of the individual sensitivity regions (Fig.1(b)).



(a) (b)

Figure 1: (a) Sensitivity region of an ultrasonic transducer. (b) Joint sensitivity region of two transducers.

The target primitives modeled in this study are *plane*, *corner*, *acute corner*, *edge*, and *cylinder* (Fig.2). Since the wavelength ($\lambda \cong 8.6$ mm at $f_o = 40$ kHz) is much larger than the typical roughness of surfaces encountered in laboratory environments, targets in these environments reflect acoustic beams specularly, like a mirror. Detailed physical reflection models of these target primitives with corresponding echo signal models are provided in [1]. In the following, A_{aa}, A_{ab}, A_{ba}, and A_{bb} denote the maximum values of the sonar echo signals, and t_{aa}, t_{ab}, t_{ba}, and t_{bb} denote the TOF readings extracted from these signals. The first index in the subscript indicates the transmitter, the second index denotes the receiver.

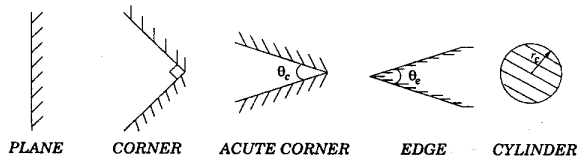


Figure 2: Target primitives used in this study.

3 Target Differentiation Algorithm

In this section, the target differentiation algorithm used in earlier work [1] is reviewed. This will not only be useful in motivating the structure of the inputs to be used in the neural network, but will also provide a basis for comparison of its performance. The algorithm is based on the idea of exploiting amplitude and TOF differentials in resolving target type and may be summarized in the form of rules:

if $[t_{aa}(\alpha) - t_{ab}(\alpha)] > k_t\sigma_t$ **and** $[t_{bb}(\alpha) - t_{ba}(\alpha)] > k_t\sigma_t$ then **acute corner** $\rightarrow$ exit

if $[A_{aa}(\alpha) - A_{ab}(\alpha)] > k_A\sigma_A$ **and** $[A_{bb}(\alpha) - A_{ba}(\alpha)] > k_A\sigma_A$ then **plane** $\rightarrow$ exit

if $[\max\{A_{aa}(\alpha)\} - \max\{A_{bb}(\alpha)\}] < k_A\sigma_A$ **and** $[\max\{A_{bb}(\alpha)\} - \max\{A_{ab}(\alpha)\}] < k_A\sigma_A$ then **corner** $\rightarrow$ exit

else **edge**, **cylinder** or **unknown** $\rightarrow$ exit

Here, $\sigma_A(\sigma_t)$ is the amplitude (TOF) noise standard deviation, and $k_A(k_t)$ is the number of $\sigma_A(\sigma_t)$ which is employed as a safety margin to achieve robustness in the differentiation process. Differentiation is achievable only in those cases where the differentials exceed $k_A\sigma_A(k_t\sigma_t)$. If this is not the case, a decision cannot be made and the target type remains unknown.

4 Target Classification and Localization with Neural Networks

The algorithm summarized above does not provide a distinctive rule to differentiate edges and cylinders. One way of differentiating edges and cylinders is to use the radius of curvature estimation method proposed in [3]. In this work, neural networks are employed to identify and resolve parameter relations embedded in the characteristics of experimentally-obtained sonar returns from all target primitives in a robust manner.

Panasonic transducers are used with aperture radius $a = 0.65$ cm, resonance frequency $f_o = 40$ kHz, and $\theta_o \cong 54°$ [11] (Fig.1). The center-to-center separation of the transducers used in the experiments is $d = 25$ cm. The entire sensing unit is mounted on a small stepper motor with step size 1.8°. The motion of the stepper motor is controlled through the parallel port of a PC 486 with the aid of a microswitch. Data acquisition from the sonars is through a DAS-50 A/D card with 12-bit resolution and 1 MHz sampling frequency. Echo signals are processed on a PC 486 in the C programming language. Starting at the transmit time, 10,000 samples of each echo signal are collected and thresholded. The amplitude information is extracted by finding the maximum value of the signal after the threshold is exceeded.

The targets employed in this study are: cylinders with radii 2.5, 5.0 and 7.5 cm, a planar target, a corner, an edge ($\theta_e = 90°$) and an acute corner ($\theta_c = 60°$). Amplitude and TOF data from these targets are collected at 25 different locations (r, θ) for each target, from $\theta = -20°$ to $\theta = 20°$ in 10° increments, and from $r = 35$ cm to $r = 55$ cm in 5 cm increments (Fig.3). The target primitive located at r and θ is scanned by the sensing unit for scan angle $-52° \leq \alpha \leq 52°$ with 1.8° increments. With the given scan range and motor step size, 58 ($= \frac{2\times52}{1.8}$) sets of amplitude and TOF

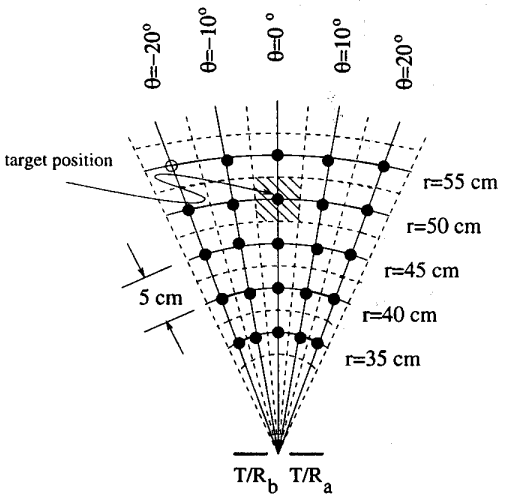


Figure 3: Discrete network training locations.

data (A_{aa}, A_{ab}, A_{ba}, A_{bb} ; t_{aa}, t_{ab}, t_{ba}, t_{bb}) are acquired for each target location. The structure of these amplitude and TOF characteristics is provided in [1].

The network employed has one hidden, one input and one output layer. The inputs to the neural network are 58 samples each of the difference signals $A_{aa}(\alpha)-A_{ab}(\alpha)$, $A_{bb}(\alpha)-A_{ba}(\alpha)$, $t_{aa}(\alpha)-t_{ab}(\alpha)$, and $t_{bb}(\alpha)-t_{ba}(\alpha)$, resulting in 232 input layer neurons.

The hidden layer comprises 100 neurons. This number was determined by a process known as *enlarging*, which starts with a relatively small number of neurons and increases the size of the hidden layer until learning occurs. The number of output layer neurons is 21. The first seven neurons encode the target type. The next seven represent the target range r which is binary coded with a resolution of 0.25 cm. The last seven neurons represent the target azimuth θ with respect to the line-of-sight of the sensing unit, which is also binary coded with resolution 0.5°.

Four sets of data are collected for each target location for each target primitive, resulting in 700 (= 4 data sets × 25 locations × 7 target types) sets of waveform data for training. The network is trained with these 700 sets of data, using the back-propagation algorithm [16] with a learning constant equal to 0.01, momentum constant equal to 0.9, and a sigmoid-type nonlinearity. With the software tool PlaNet, the weights are found in about one hour [12].

The network is tested as follows: Each target primitive is placed in turn in each of the 25 locations shown in Fig.3. Four sets of measurements are collected for each combination of target type and location, again resulting in 700 sets of experimentally obtained data. The neural network estimates the target type, range, and azimuth from this data.

target type	% of correct classif.	% of correct r estim. error tolerance ϵ_r			% of correct θ estim. error tolerance ϵ_θ		
		±1cm	±5cm	±10cm	±2°	±10°	±20°
P	100(90)	66(50)	78(69)	88(80)	96(68)	98(93)	99(92)
C	99(100)	90(72)	92(86)	95(91)	90(85)	92(90)	93(90)
E	99(96)	59(56)	82(78)	92(90)	77(69)	89(81)	97(95)
AC	98(99)	83(75)	91(83)	95(88)	87(80)	93(90)	97(92)
CY1	90(88)	80(60)	89(78)	94(87)	97(83)	97(95)	98(99)
CY2	89(70)	77(65)	82(78)	89(91)	98(78)	99(98)	99(96)
CY3	92(86)	86(60)	92(76)	98(86)	97(60)	99(94)	99(96)
AVG	95(90)	77(63)	87(78)	93(88)	92(75)	95(92)	97(94)

Table 1: The percentages of correct classification, range (r) and azimuth (θ) estimation.

In Table 1, the numbers outside the parentheses give the resulting percentages of correct target-type classification, correct range and correct azimuth estimation. A range or azimuth estimate is considered correct if it is within an error tolerance of ϵ_r or ϵ_θ of the actual range or azimuth respectively. The average percentages over all target types are also given in the last row of the table. The percentage of correct target type classification is high at 95%. The percentage of correct range estimation lies in the range 77% to 93%, and that for correct azimuth estimation lies in the range 92% to 97%, depending on the error tolerance level (ϵ_r or ϵ_θ). For comparison, the average correct target type classification obtained using the differentiation algorithm given in Section 3 on the same data set is 61% and the average correct range and azimuth estimation percentages are 72% and 59% respectively for $|\epsilon_r| = 1$ cm and $|\epsilon_\theta| = 2°$.

The network is also tested for targets situated arbitrarily in the continuous estimation space and not necessarily confined to the 25 locations of Fig.3. The results are given in Table 1 in parentheses. As expected, the percentages in this case are lower than those for the training positions; the network gives the best results when a target is situated exactly at one of the training sites. Noting that the network was trained only at 25 locations and at grid spacings of 5 cm and 10°, it can be concluded from the percentage of correct r and θ estimates obtained at error tolerances of $|\epsilon_r| = 1$ cm and $|\epsilon_\theta| = 2°$, that the network demonstrates the ability to interpolate between the training grid locations. Thus, the neural network maintains a certain spatial continuity between its input and output and does not haphazardly map positions which are

not drawn from the 25 locations of Fig.3. The correct target type percentages in Table 1 are quite high and the localization accuracy would be acceptable in many applications. If better estimates are required, this can be achieved by reducing the training grid spacing in Fig.3.

In addition to the above structure, multi-stage and modular network structures have also been implemented and tested. In the multi-stage network, training is performed in two or three stages by introducing the target type, range and azimuth information in sequence. In the modular structure, three separate networks for target type, range, and azimuth have been employed. However, these structures did not result in substantial improvement in the differentiation and localization results.

testing condition	type %	range %	azimuth %
$A_{aa} - A_{ab} = 0$	58(50)	20(13)	34(19)
$A_{bb} - A_{ba} = 0$	59(45)	22(16)	35(19)
$t_{aa} - t_{ab} = 0$	94(83)	71(58)	89(73)
$t_{bb} - t_{ba} = 0$	95(84)	71(58)	89(73)
$A_{aa} = t_{aa} = 0$	25(22)	13(10)	16(11)
$A_{bb} = t_{bb} = 0$	22(19)	17(14)	15(10)
$A_{ab} = t_{ab} = 0$	16(15)	9(7)	16(9)
$A_{ba} = t_{ba} = 0$	25(24)	13(9)	13(7)
$t_{aa} = 0$	94(82)	69(56)	89(73)
$t_{ab} = 0$	95(83)	74(61)	90(74)
$t_{ba} = 0$	95(82)	74(60)	90(74)
$t_{bb} = 0$	94(81)	67(55)	87(72)
$A_{aa} = 0$	25(21)	13(10)	22(14)
$A_{ab} = 0$	16(15)	9(7)	16(10)
$A_{ba} = 0$	24(23)	9(7)	13(8)
$A_{bb} = 0$	23(19)	17(13)	15(9)
25% of inputs zeroed	81(73)	38(32)	65(53)

Table 2: The percentages of correct classification, range, and azimuth estimation when the targets are tested in the training positions and at arbitrary positions (in parentheses).

The network is further tested to investigate its robustness in physically plausible failure or missing data situations. The same 700 sets of test data are used, but with some of the network inputs equated to zero. The results are tabulated in Table 2 for the cases where testing is done at the training locations and at arbitrary locations. Rows 1–4 correspond to the case when one of the differential input channels is made completely unavailable to the network. Rows 4–8 represent failure of one of the transducers. Rows 7 and 8 also correspond to the case when the target does not fall within the joint sensitivity region of the two transducers. Rows 9–12 correspond to the case when the amplitudes of the echo waveforms fall completely below the preset threshold level. This happens when the target is very far away from the sensor or too far off its line-of-sight. In this case, TOF information cannot be extracted although amplitude information is still available. Rows 13–16 correspond to the complementary case where TOF information is available but amplitude information is not. Finally, the effect of the absence of randomly selected samples of the input data is investigated and presented in row 17. Here, 25% of the input data is made unavailable to the network by randomly setting some of the input samples to zero. Note that this percentage is the same as the percentage of samples excluded when one of the input channels is completely blocked. The result of these tests indicate that amplitude information is much less dispensable, despite the fact that TOF is the more commonly used parameter. In addition, there is a 1% to 16% decrease in performance when the object is at an arbitrary location as compared to when it is at a training location.

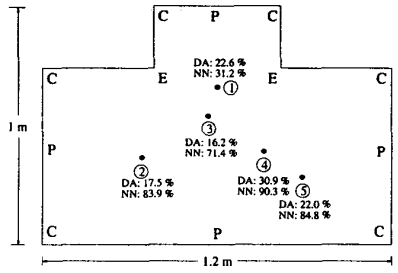


Figure 4: The experimental test room. For each node, correct decision percentages are given for the differentiation algorithm (DA) and the neural network (NN).

The sensing unit has been mounted on a small mobile robot navigating in a test room consisting of eight planes, six corners and two edges. Data were collected at five different positions by scanning the environment (Fig.4). At each scan angle α, a decision about the type of the target which is closest to the line-of-sight of the sensing unit is made by employing both the differentiation algorithm and the neural network classifier. The correct decision percentages averaged over the entire scan are shown in Fig.4, indicating that the neural network outperforms the differentiation algorithm at all five positions. We emphasize that the neural network employed here is the same one trained with the discrete locations of Fig.3; it was not trained specifically for this room. Yet, it had a success rate varying between 71% to 90%, except for position 1

where the proximity of the several features to each other, and the fact that the two edges are closer to the robot than the minimum training distance overwhelm the network. This example also shows that the neural network classifier is robust to higher-order and multiple reflections in the environment.

The next example involves path planning of a mobile robot in an environment cluttered with cylinders. In addition, there is a corner behind the cylinders which causes higher-order and multiple reflections. A mobile robot equipped with the previously described sensing unit is needs to find the minimum path from its current position to the goal (Fig.5). We again compare the differentiation algorithm with the neural network classifier. The decisions made when the sensor is looking towards each of the three cylinders are also indicated on the same figure. The differentiation algorithm detects two of the cylinders as planes and one as a corner, whereas the neural network classifier can correctly identify all cylinders. (In this example, use of the differentiation algorithm would have resulted in doubling of the path length.)

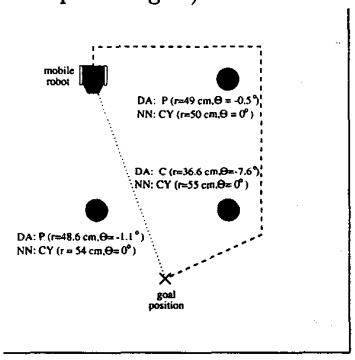


Figure 5: Mobile robot path planning: The dotted and dashed lines correspond to the minimum paths designed by employing the neural network (NN) and the differentiation algorithm (DA) respectively.

5 Comparative Analysis & Discussion

In this part, the network is compared to the algorithm discussed in Section 3 (for the three target types identified by that algorithm) and two previously proposed methods [14]: Dempster-Shafer evidential reasoning [13] and majority voting [7], which are employed to resolve conflicts in the decisions of multiple sensors. Dempster-Shafer evidential reasoning is useful for representing and manipulating beliefs of decision-makers about an event, especially in situations where their beliefs reflect uncertainty or ignorance. Voting is a simple tool for resolving conflicts when the emphasis is on the view of the majority. Both of these fusion methods and neural networks share the feature of being non-parametric (i.e., no underlying distribution of input data or noise is assumed).

method	target types discriminated	%	sensor configuration
DA [1, 2]	P,C,AC	≤61%	single scanning node
DS evidential reasoning [14]	P,C,AC, {E,CY,U}	87%	15 scanning sensing nodes
voting [14]	P,C,AC, {E,CY,U}	88%	15 scanning sensing nodes
		90%	ordered 15 scanning sensing nodes
radius of curv. estimation [3]	P,CY,{C,E}	80%	single stationary node
NN	P,C,AC,E,CY	95%	single scanning node

Table 3: Comparison of the various methods. Target types in braces can be resolved only as a group.

For target differentiation based purely on raw data, the algorithm of Section 3 gives a correct differentiation percentage of 61%. In [14], based on this algorithm, sensors assign probability masses to plane, corner and acute corner target types using Dempster-Shafer evidential reasoning. Combining the opinions of 15 sensing nodes using Dempster's rule improves the correct decision percentage to 87%. When the sensors' beliefs about target types are counted as votes and the majority vote is taken as the outcome, the number rises to 88%. Moreover, using various ordering strategies in the voting algorithm further increases this number to 90%. Using these two fusion methods, only planes, corners and acute corners can be differentiated. Using the neural network described in this paper, seven different target types can be differentiated and localized employing only a *single* sensor node, with a higher correct decision percentage (95%) than with the earlier-used decision rules employing several sensors. The fact that the neural network is able to distinguish all target types indicates that it must be making more effective use of the available data than the methods used earlier. The neural network's performance shows that the original training data set does contain the information sufficient to differentiate the seven target types, but the other methods mentioned above are not able to resolve this identifying information. The neural network allows differentiation of more targets with increased accuracy by exploiting the hidden identifying features in the differential ampli-

tude and TOF characteristics of the targets. A comparison of the processing methods mentioned is presented in Table 3.

6 Conclusion

In this study, neural networks are employed to process real sonar data after trained to learn identifying parameter relations for the target primitives. This system uses amplitude as well as TOF data, allowing for improved differentiation and localization. The robustness of the network to partial removal of amplitude and TOF information has been investigated, demonstrating that the network is robust to different failure modes. The results recommend that amplitude measurements should be more widely exploited, rather than limiting oneself to the more conveniently available TOF measurements.

A comparison with previously investigated approaches to decision fusion (evidential reasoning and voting) indicates improved performance. The neural network approach can classify a larger number of targets than these methods, often with greater accuracy. Furthermore, it accomplishes this using only a single sensing node. Had the number of sensing nodes (15) been reduced in the other methods, their accuracy would have been even worse. Therefore, among the methods which have been considered, neural networks emerge as the strongest alternative. Although trained on a discrete and relatively coarse grid, the network is able to interpolate between the grid locations and offers higher resolution (especially in θ) than that implied by the grid size. The correct estimation rates for target type, range and azimuth can be further increased by employing a finer grid for training.

The results presented here recommend wider use of neural networks as robust pattern classifiers in sensor-based robotics. There is scope for further application of neural networks to sonar, based on the facts that sonar data is difficult to interpret, physical models can be complex even for simple TOF sonar, and expressions for sonar returns are very complicated even for the simplest target types. Acoustic propagation is also subject to distortion with changes in environmental conditions. Future work will investigate scale- and shift-invariant features which are not affected by the target's location and orientation, in order to reduce the number of training patterns needed. Unsupervised learning algorithms will also be considered to make the classification process more robust to changes in environmental conditions.

References

[1] B. Ayrulu and B. Barshan. Identification of target primitives with multiple decision-making sonars using evidential reasoning. *Int. J. Robotics Res.*, 17(6):598–623, 1998.

[2] B. Barshan and R. Kuc. Differentiating sonar reflections from corners and planes by employing an intelligent sensor. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(6):560–569, 1990.

[3] B. Barshan and A. Ş. Sekmen. Radius of curvature estimation and localization of targets using multiple sonar sensors. *J. Acoust. Soc. Am.*, 105(4):2318–2331, 1999.

[4] Ö. Bozma and R. Kuc. A physical model-based analysis of heterogenous environments using sonar-ENDURA method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(5):497–506, 1994.

[5] R. P. Lippman. An introduction to computing with neural nets. *IEEE ASSP Magazine*, pp.4–22, April 1987.

[6] R. Kuc. Biomimetic sonar recognizes objects using binaural information. *J. Acoust. Soc. Am.*, 102(2):689–696, 1997.

[7] L. Lam and C. Y. Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Trans. Systems Man and Cybernetics*, 27(5):553–568, 1997.

[8] J. J. Leonard and H. F. Durrant-Whyte. *Directed Sonar Navigation.* Kluwer Academic, London, 1992.

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

[10] T. Ogawa, K. Kameyama, R. Kuc, and Y. Kosugi. Source localization with network inversion using an answer-in-weights scheme. *IEICE Trans. Information and Systems*, E79-D(5):608–619, 1996.

[11] Panasonic Corp. Ultrasonic ceramic microphones. Burlington, MA, 1989.

[12] Y. Miyata, A User's Guide to PlaNet Version 5.6, A Tool for Constructing, Running, and Looking into a PDP Network", Computer Science Dept., Univ. of Colorado, Boulder, Colorado, January 1991.

[13] G. Shafer. *A Mathematical Theory of Evidence.* Princeton Univ. Press, NJ, 1976.

[14] S. W. Utete, B. Barshan, and B. Ayrulu. Voting as validation in robot programming. *Int. J. Robotics Res.*, 18(4):401–413, 1999.

[15] S. Watanabe and M. Yoneyama. An ultrasonic visual sensor for three-dimensional object recognition using neural networks. *IEEE Trans. Rob. Autom.*, 8(2):240–249, 1992.

[16] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[17] J. Zemanek. Beam behavior within the nearfield of a vibrating piston. *J. Acoust. Soc. Am.*, 49(1 (Part 2)):181–191, 1971.