PERGAMON

Neural Networks 14 (2001) 355–373

Neural
Networks

www.elsevier.com/locate/neunet

Contributed article

Neural networks for improved target differentiation and localization with sonar

Birsel Ayrulu, Billur Barshan*

Department of Electrical Engineering, Bilkent University, Bilkent, 06533 Ankara, Turkey

Received 25 April 2000; accepted 4 January 2001

Abstract

This study investigates the processing of sonar signals using neural networks for robust differentiation of commonly encountered features in indoor robot environments. Differentiation of such features is of interest for intelligent systems in a variety of applications. Different representations of amplitude and time-of-flight measurement patterns acquired from a real sonar system are processed. In most cases, best results are obtained with the low-frequency component of the discrete wavelet transform of these patterns. Modular and non-modular neural network structures trained with the back-propagation and generating–shrinking algorithms are used to incorporate learning in the identification of parameter relations for target primitives. Networks trained with the generating–shrinking algorithm demonstrate better generalization and interpolation capability and faster convergence rate. Neural networks can differentiate more targets employing only a single sensor node, with a higher correct differentiation percentage (99%) than achieved with previously reported methods (61–90%) employing multiple sensor nodes. A sensor node is a pair of transducers with fixed separation, that can rotate and scan the target to collect data. Had the number of sensing nodes been reduced in the other methods, their performance would have been even worse. The success of the neural network approach shows that the sonar signals do contain sufficient information to differentiate all target types, but the previously reported methods are unable to resolve this identifying information. This work can find application in areas where recognition of patterns hidden in sonar signals is required. Some examples are system control based on acoustic signal detection and identification, map building, navigation, obstacle avoidance, and target-tracking applications for mobile robots and other intelligent systems. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Artificial neural networks; Sonar sensing; Target differentiation; Target localization; Feature extraction; Learning; Discrete wavelet transform; Acoustic signal processing

1. Introduction

Neural networks have been employed efficiently as pattern classifiers in numerous applications (Lippman, 1987). These classifiers are non-parametric and make weaker assumptions on the shape of the underlying distributions of input data than traditional statistical classifiers. Therefore, they can prove more robust when the underlying statistics are unknown or the data are generated by a nonlinear system.

Neural networks have been used in sonar and radar signal processing (Chang, Bosworth & Carter, 1993; Widrow & Winter, 1988); for instance, in the identification of ships from observed parametric radar data (Prieve & Marchette, 1987). The motivation behind the use of neural network classifiers in sonar or radar systems is the desire to emulate the remarkable perception and pattern recognition capabilities of humans and animals, such as the powerful ability of dolphins and bats to extract detailed information about their environments from acoustic echo returns (Au, 1994; Roitblat, Au, Nachtigall, Shizumura & Moons, 1995; Simmons, Saillant, Wotton, Haresign, Ferragamo & Moss, 1995). A comparison between neural networks and standard classifiers for radar-specific emitter identification is provided by Willson (1990). An acoustic imaging system which combines holography with multi-layer feed-forward neural networks for 3-D object recognition is proposed in Watanabe and Yoneyama (1992). A neural network which can recognize 3-D cubes and tetrahedra independent of their orientation using sonar is described in Dror, Zagaeski and Moss (1995). Neural networks have also been used in the classification of sonar returns from undersea targets, for example, in Gorman and Sejnowski (1988), where the correct classification percentage of the network employed

* Corresponding author. Tel.: +90-312-290-2161; fax: +90-312-266-4192.
E-mail address: billur@ee.bilkent.edu.tr (B. Barshan).

0893-6080/01/$ - see front matter © 2001 Elsevier Science Ltd. All rights reserved.
PII: S0893-6080(01)00017-X

(90%) exceeds that of a nearest neighborhood classifier (82%). Another application of neural networks to sonar data is in the classification of cylinders under water or in sediment where the targets are made of different materials (Gorman & Sejnowski, 1988; Roitblat et al., 1995), made of the same material but with different diameters (Roitblat et al., 1995), or in the presence of a second reflector in the environment (Ogawa, Kameyama, Kuc & Kosugi, 1996). Neural networks have also been used in naval friend-or-foe recognition in underwater sonar (Miller & Walker, 1992).

Performance of neural network classifiers is affected by the choice of parameters of the network structure, training algorithm, and input signals, as well as parameter initialization (Alpaydın, 1993; Au, Andersen, Rasmussen, Roitblat & Nachtigall, 1995). This article investigates the effect of various representations of input sonar signals and two different training algorithms on the performance of neural networks with different structures used for target classification and localization. The input signals are different functional forms and transformations of amplitude and time-of-flight characteristics of commonly encountered targets acquired by a real sonar system.

The most common sonar ranging system is based on *time-of-flight* (TOF) which is the time elapsed between the transmission and reception of a pulse. Differential TOF models of targets have been used by several researchers (Bozma & Kuc, 1991; Leonard & Durrant-Whyte, 1992; Manyika & Durrant-Whyte, 1994). In Bozma and Kuc (1991) a single sensor is used for map building. First, edges are differentiated from planes/corners from a single vantage point. Then, planes and corners are differentiated by scanning from two separate locations and using the TOF information in the complete sonar scans of the targets. Rough surfaces have been considered in Bozma and Kuc (1994). In Leonard and Durrant-Whyte (1992) a similar approach has been proposed to identify these targets as beacons for mobile robot localization. Manyika has used differential TOF models for target tracking (Manyika & Durrant-Whyte, 1994). Systems using only qualitative information (Kuc, 1993), combining amplitude, energy and duration of the echo signals together with TOF information (Ayrulu & Barshan, 1998; Barshan & Kuc, 1990; Bozma & Kuc, 1994), or exploiting the complete echo signal (Kuc, 1997) have also been considered.

Sensory information from a *single* sonar has poor angular resolution and is usually not sufficient to differentiate more than a small number of target primitives (Barshan & Kuc, 1990). Improved target classification can be achieved by using multi-transducer pulse/echo systems and by employing both amplitude and TOF information. However, a major problem with using the amplitude information of sonar signals is that the amplitude is very sensitive to environmental conditions. For this reason, and also because the standard electronics typically provide only TOF data, amplitude information is rarely used. In earlier work, Barshan and Kuc (1990) introduce a method based on both amplitude and TOF information to differentiate planes and corners. This algorithm is extended to other target primitives in Ayrulu and Barshan (1998). In the present paper, neural networks are used to process amplitude and TOF information so as to reliably handle the target classification problem.

The paper is organized as follows. Section 2 explains the sensing configuration used in this study and introduces the target primitives. In Section 3, multi-layer feed-forward neural networks are briefly reviewed. Two training algorithms, namely back-propagation and generating–shrinking algorithms, are described in Section 4. In Section 5, preprocessing techniques employed prior to neural network classifiers are briefly described. In Section 6, various types of input signals to the neural network classifiers are proposed. In Section 7, the effect of these input signals and training algorithms on the performance of neural networks in target classification and localization are compared experimentally. In the last section, concluding remarks are made and directions for future work are discussed.

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 measurement $r = ct_0/2$ is obtained when the echo amplitude first exceeds a preset threshold level τ back at the receiver at time t_0. Here, t_0 is the TOF and c is the speed of sound in air (at room temperature, $c = 343.3$ m/s). Many ultrasonic transducers operate in this pulse–echo mode (Hauptmann, 1993). The transducers can function both as receiver and transmitter. Most systems commonly in use are able to detect only the very first echo after pulse transmission.

In this study, the far-field model of a piston-type transducer having a circular aperture is considered (Zemanek, 1971). It is observed that the echo amplitude decreases with increasing range r and azimuth θ, which is the deviation angle from normal incidence as illustrated in Fig. 1(b). The echo amplitude falls below τ when $|\theta| > \theta_0$, which is related to the aperture radius a and the resonance frequency f_0 of the transducer by (Zemanek, 1971) $\theta_0 = \sin^{-1}(0.61c/af_0)$. The radiation pattern is caused by interference effects between different radiating zones on the transducer surface. The transducers used in this study are Panasonic transducers (Panasonic Corporation, 1989) with aperture radius $a = 0.65$ cm and resonance frequency $f_0 = 40$ kHz. Therefore, the half beamwidth angle $\theta_0 \cong 54°$ for these transducers.

The major limitation of ultrasonic transducers comes from their large beamwidth. Although these devices return accurate range data, they cannot provide direct information on the angular position of the object from which the reflection was obtained. The transducer can operate both as transmitter and receiver and detect echo signals reflected from targets within its *sensitivity region* (Fig. 1(a)). Thus, with a single stationary transducer, it is not possible to estimate the

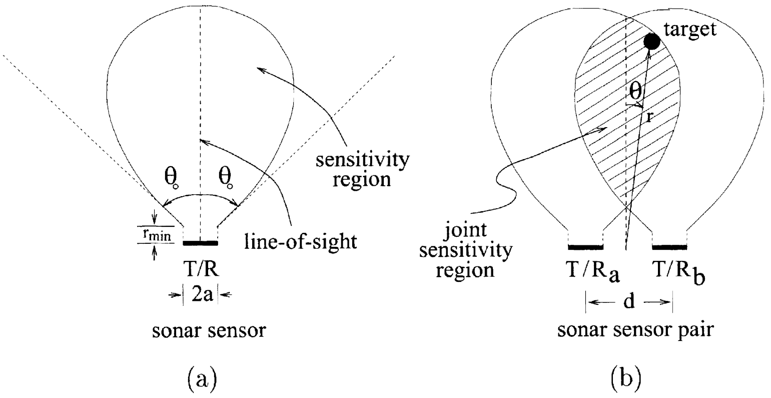

Fig. 1. (a) Sensitivity region of an ultrasonic transducer. Sidelobes are not shown. (b) Joint sensitivity region of a pair of ultrasonic transducers. The intersection of the individual sensitivity regions serves as a reasonable approximation to the joint sensitivity region. (© 2000 IEEE)

azimuth of a target with better resolution than the angular resolution of the device which is approximately $2\theta_0$. The reflection point on the object can lie anywhere along a circular arc (as wide as the beamwidth) at the measured range. More generally, when one sensor transmits and another receives, 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)). In this case, the reflection point lies on the arc of an ellipse whose focal points are the transmitting and receiving transducers. The angular extent of these circular and elliptical arcs is determined by the sensitivity regions of the transducers. In our system, two identical acoustic transducers *a* and *b* with center-to-center separation *d* are employed to improve the angular resolution. These two transducers together constitute what we will refer to as a *sensor node* throughout this paper. The extent of the sensitivity regions is different for different targets, which, in general, exhibit different reflection properties. For example, for edge-like or pole-like targets, this region is much smaller but of similar shape, and for planar targets, it is more extended (Barshan, 1991).

The target primitives employed in this study are *plane*, *corner*, *acute corner*, *edge* and *cylinder* (Fig. 2). Most ultrasonic systems operate below a resonance frequency of 200 kHz so that the propagating waves have wavelengths well above several millimeters. In our case, since the operating wavelength ($\lambda \cong 8.6$ mm at $f_0 = 40$ 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. Details on the objects which are smaller than the wavelength cannot be resolved (Brown, 1986). Specular reflections allow the single transmitting–receiving transducer to be viewed as a separate transmitter T and virtual receiver R (Kuc & Siegel, 1987). Detailed specular reflection models of these target primitives with corresponding echo signal models are provided in Ayrulu and Barshan (1998).

3. Multi-layer feed-forward neural networks

Multi-layer feed-forward neural networks (multi-layer perceptrons) have been widely used in areas such as target detection and classification (Bai & Farhat, 1992), speech processing (Cohen, Franco, Morgan, Rumelhart & Abrash, 1993), system identification (Narendra & Parthasarathy, 1990), control theory (Jordan & Jacobs, 1990), medical applications (Galicki, Witte, Dörschel, Eiselt & Griessbach, 1997), and character recognition (LeCun, Boser, Denker, Henderson, Howard, Hubbard et al., 1990). They consist of an input layer, one or more hidden layers, and a single output layer, each comprised of a number of units called neurons. These networks have three distinctive characteristics. The model of each neuron includes a smooth nonlinearity, the network contains one or more hidden layers to extract progressively more meaningful features, and the network exhibits a high degree of connectivity. Due to the presence of distributed form of nonlinearity and high degree of connectivity, theoretical

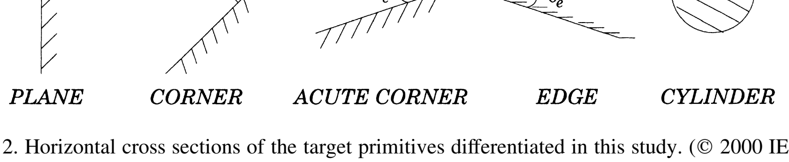

Fig. 2. Horizontal cross sections of the target primitives differentiated in this study. (© 2000 IEEE)

analysis of multi-layer perceptrons is difficult. These networks are trained to compute the boundaries of decision regions in the form of connection weights and biases by using training algorithms. In this study, two training algorithms are employed, namely, back-propagation and generating–shrinking algorithms which are briefly reviewed in the next section.

Two well-known methods for determining the number of hidden layer neurons in feed-forward neural networks are *pruning* and *enlarging* (Haykin, 1994). Pruning begins with a relatively large number of hidden layer neurons and eliminates unused neurons according to some criterion. Enlarging begins with a relatively small number of hidden layer neurons and gradually increases their number until learning occurs.

It is proven that the multi-layer perceptron approximates the Bayes optimal discriminant function in the mean-square sense when it is trained as a classifier using the back-propagation algorithm with infinitely many training samples and uniform losses (Ruck, Rogers, Kabrisky, Oxley & Suter, 1990). The outputs of this classifier also represent the corresponding posterior probabilities (Ruck et al., 1990). However, the accuracy of the approximation is limited by the architecture of the network being trained such that if the hidden layer neurons are too few, then the approximation will not provide a good match. Fortunately, it is not dependent on the number of layers and the type of activation function (nonlinearity) used.

4. Training algorithms

4.1. Back-propagation algorithm

The back-propagation algorithm is used frequently due to its simplicity, extraction power of useful information from examples, and capacity of implicit information storage in the form of connection weights, and applicability to binary or real-valued patterns (Werbos, 1990). While training with the back-propagation algorithm, a set of *training patterns* is represented to the network and propagated forward to determine the resulting signal at the output. The back-propagation algorithm is a gradient-descent procedure that minimizes the error at the output. The average error at a particular cycle of the algorithm is the average of the Euclidean distance between the actual output of the network and the desired output over all training patterns:

$$E_{\text{ave}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\|d_i - o_i\|^2 \qquad (1)$$

Here, N is the number of the training patterns, d_i is the desired output for the ith pattern and o_i is the actual output of the network for the ith pattern. The error is back-propagated through the network in order to adjust the connection weights and biases. Adjustment of these quantities is proportional to the descent gradient of sum of squared errors with a constant called the *learning rate* chosen between zero and one. The speed of the training procedure is very slow with too small learning rates, but there can be stability problems if the learning rate is chosen too large. To avoid these problems, a second term in the adjustment equation, called the *momentum term*, is added (Rumelhart, Hinton & Williams, 1986). This term is proportional to the previous adjustment through a *momentum constant*. In this study, the stopping criterion we have used while training networks with the back-propagation algorithm is as follows. The training is stopped either when the average error is reduced to 0.001 or if a maximum of 10,000 epochs is reached, whichever occurs earlier. The second case occurs very rarely.

4.2. Generating–shrinking algorithm

The generating–shrinking algorithm first builds and then shrinks or prunes a four-layer feed-forward neural network, offering fast convergence rate and 100% correct classification on the training set as reported in the study on scale-invariant texture discrimination by Chen, Thomas and Nixon (1994). The network used in the same study consists of two hidden layers with equal numbers of neurons which is initially set equal to the number of training patterns. Pre-determined connection weights are assigned. Then, the hidden layers are pruned while preserving 100% correct classification rate on the training set. The algorithm is based on the assumption that only one output neuron can take the value one (the winning neuron) and the remaining output neurons take the value zero. Since the initial connection weights take deterministic values, the network has analytically known generalization behavior. At the input layer, a pre-fixed reference number n_r that can take values between zero and infinity is used as an additional input to control the generalization capability of the network. The algorithm achieves scale-invariant generalization behavior as n_r approaches zero, and behaves like a nearest-neighborhood classifier as it tends to infinity. A comparison with the back-propagation algorithm in Chen et al. (1994) indicates that the generating–shrinking algorithm does not have the convergence problems of the back-propagation algorithm and has a substantially faster convergence rate (2.2 versus 1260 s) and perfect generalization capability (100 versus 68%), although both networks have 100% correct classification rate on the training set. For further details of this algorithm, the reader can refer to Chen et al. (1994).

5. Preprocessing of the input signals

In this section, we give a brief description of the preprocessing techniques used on the input signals to the neural networks considered in this study.

5.1. Fourier transform

Fourier analysis is a well known technique, widely used in signal processing to study the spectral behavior of a signal (Bracewell, 1986). The discrete Fourier transform (DFT) of a signal $f(n)$ is defined as:

$$F(k) = \mathscr{F}\{f(n)\} \triangleq \frac{1}{N}\sum_{n=0}^{N-1} f(n)\mathrm{e}^{\frac{-2\pi jnk}{N}} \tag{2}$$

where N is the length of the signal $f(n)$.

5.2. Wavelet transform

Wavelet transform is a relatively new analytical tool for engineers, scientists and mathematicians for time-frequency analysis, and a new basis for representing functions (Chui, 1992). The discrete wavelet transform (DWT) of a function $f(t) \in \mathbf{L}^2$ can be written as:

$$f(t) = \sum_{k=-\infty}^{\infty} c(k)\varphi_k(t) + \sum_{j=0}^{\infty}\sum_{k=-\infty}^{\infty} d(j,k)\psi_{j,k}(t) \tag{3}$$

where

$$c(k) = \langle f(t), \varphi_k(t)\rangle = \int f(t)\varphi_k(t)\mathrm{d}t \tag{4}$$

and

$$d(j,k) = \langle f(t), \psi_{j,k}(t)\rangle = \int f(t)\psi_{j,k}(t)\mathrm{d}t \tag{5}$$

The coefficients $\{c(k)\}_{k=-\infty}^{\infty}$ and $\{d(j,k)\}_{j=0,k=-\infty}^{\infty}$ are called DWT of the function $f(t)$. These coefficients completely describe the original signal and can be used in a way similar to Fourier series coefficients. At this point, it is necessary to consider the functions $\varphi_k(t)$ and $\psi_{j,k}(t)$ in Eq. (3). A set of scaling functions in terms of integer translations of a basic scaling function $\varphi(t)$ is represented as $\varphi_k(t) = \varphi(t-k)$; $k \in \mathbf{Z}$ and $V_0 = \mathrm{Span}\{\varphi_k(t)\} \subset \mathbf{L}^2$. A family of functions generated from the basic scaling function $\varphi(t)$ by scaling and translation is represented by $\varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k)$ and $V_j = \mathrm{Span}\{\varphi_{j,k}(t)\}$ such that $\cdots \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset \mathbf{L}^2$, $V_{-\infty} = 0$, $V_\infty = \mathbf{L}^2$.

Since $\varphi(t) \in V_1$, it can be represented in terms of basis functions of V_1. Then:

$$\varphi(t) = \sum_{n=0}^{M-1} h(n)\varphi(2t-n) \tag{6}$$

where $h(n)$, $n = 0, \ldots, M-1$ is called the *scaling filter*. Important features of the signal can be better described by not using $\varphi_{j,k}(t)$ with increasing j to increase the size of the subspace spanned by the scaling functions, but by defining a slightly different set of functions that spans the differences between spaces spanned by various scales of $\varphi(t)$. These functions are called *wavelet functions*. If the orthogonal complement of V_j in V_{j+1} is denoted as W_j, then

$$\begin{aligned} V_1 &= V_0 \oplus W_0 \\ V_2 &= V_0 \oplus W_0 \oplus W_1 \\ &\vdots \\ \mathrm{L}^2 &= V_0 \oplus W_0 \oplus W_1 \oplus \cdots \end{aligned} \tag{7}$$

where $\oplus$ is the orthogonal sum operator.

Since these wavelets reside in the space spanned by the next narrower scaling function, they can be represented in terms of the scaling function as:

$$\psi(t) = \sum_{n=0}^{M-1} g(n)\varphi(2t-n) \tag{8}$$

where $g(n)$ is called the *wavelet filter* simply related to the scaling filter by

$$g(n) = (-1)^n h(M-n-1) \qquad n = 0, \ldots, M-1 \tag{9}$$

where M is the length of $h(n)$.

Finally, the procedure of finding the wavelet transform coefficients can be summarized as:

$$c_j(k) = \sum_{m=0}^{M-1} h(m-2k)c_{j+1}(m) \tag{10}$$

$$d_j(k) = \sum_{m=0}^{M-1} g(m-2k)c_{j+1}(m) \tag{11}$$

Here, $k = 0, 1, \ldots, 2^jN - 1$ where N is the number of samples of the original signal that should be a power of 2. This equation shows that the scaling and wavelet coefficients at different scales j can be obtained by convolving scaling coefficients at scale $j+1$ by $h(-n)$ and $g(-n)$ and then downsampling (take every other term) (Fig. 3(a)).

In the reconstruction part,

$$c_{j+1}(k) = 2\left[\sum_{m=0}^{M-1} c_j(m)h(k-2m) + \sum_{m=0}^{M-1} d_j(m)g(k-2m)\right]$$
$$k = 0, 1, \ldots, 2^{j+1}N - 1 \tag{12}$$

Eq. (12) shows that $c_{j+1}(k)$s can be evaluated by upsampling the scaling and wavelet coefficients, which means doubling their length by inserting zeroes between each term, then convolving them with $h(n)$ and $g(n)$, respectively, and finally adding the resulting terms and multiplying by two (Fig. 3(b)). Usually, $c_0(k)$s are taken as the samples of the original signal.

5.3. Self-organizing feature map

Self-organizing neural networks are generated by unsupervised learning algorithms that have the ability to form internal representation of the network that model the

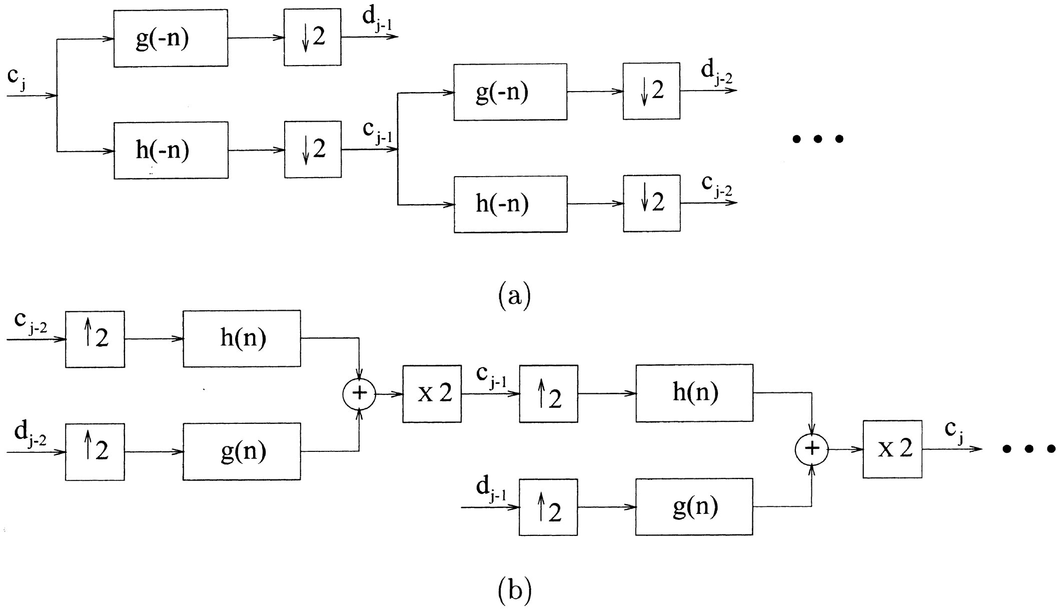

Fig. 3. (a) Analysis and (b) synthesis of DWT coefficients.

underlying structure of the input data. These networks are commonly used to solve the scaling problem encountered in supervised learning procedures. However, it is not recommended to use them by themselves for pattern classification or other decision-making processes (Haykin, 1994). Instead, best results are achieved with these networks when they are used as feature extractors prior to a linear classifier or a supervised learning process for pattern classification. The most commonly used algorithm for generating self-organizing neural networks is Kohonen's self-organizing feature-mapping algorithm (Kohonen, 1982). In this algorithm, weights are adjusted from the input layer towards the output layer where the output neurons are interconnected with local connections. These output neurons are geometrically organized in one, two, three, or even higher dimensions. This algorithm can be summarized as follows.

- Initialize the weights randomly
- Present new input from the training set
- Find the winning neuron at the output layer
- Select the neighborhood of this output neuron
- Update weights from input towards selected output neurons
- Continue with the second step until no considerable changes in the weights occur

For further details of this algorithm, one can refer to Haykin (1994).

6. Input signals to the neural network

An important issue in target differentiation with neural networks is to select those input signals to the network that carry sufficient information to differentiate all target types. Input signals resulting in a minimal network configuration (in terms of the number of layers and the number of neurons in these layers) with minimum classification error are preferable. There are many different ways of choosing input signals to the network. Apart from the sonar signals themselves, differential amplitude and TOF patterns have been used frequently in previous studies on sonar sensing (Ayrulu & Barshan, 1998; Barshan, Ayrulu & Utete, 2000; Bozma & Kuc, 1991; Leonard & Durrant-Whyte, 1992; Manyika & Durrant-Whyte, 1994). In this study, amplitude and TOF patterns and their differentials are used either in their raw form or after some preprocessing as inputs to the neural networks.

Each target is scanned with a rotating sensing unit (a pair of ultrasonic transducers with separation $d = 25$ cm mounted on a stepper motor) from $\alpha = -52°$ to $52°$ with $1.8°$ increments at 25 different locations. (Here, α is the scan angle. The scanning process and the experimental procedures are described in more detail in Section 7.) At each step of the scan, four sonar echo signals are acquired as a function of time. Typical sonar echoes from a planar target located at $r = 60$ cm and $\theta = 0°$ are illustrated in Fig. 4. In the figure, A_{aa}, A_{bb}, A_{ab}, and A_{ba} denote the maximum values of the echo signals, and t_{aa}, t_{bb}, t_{ab}, and t_{ba} denote the TOF readings extracted from the same signals. The first index in the subscript indicates the transmitting transducer, the second index denotes the receiver. At each step of the scan, only these eight amplitude and TOF values extracted from the four echo signals are recorded. For the given scan range and motor step size, 58 $(= 2 \times 52°/1.8°)$ angular samples of each of $A_{aa}(\alpha)$, $A_{bb}(\alpha)$, $A_{ab}(\alpha)$, $A_{ba}(\alpha)$, $t_{aa}(\alpha)$, $t_{bb}(\alpha)$, $t_{ab}(\alpha)$, $t_{ba}(\alpha)$ are acquired at each target location.

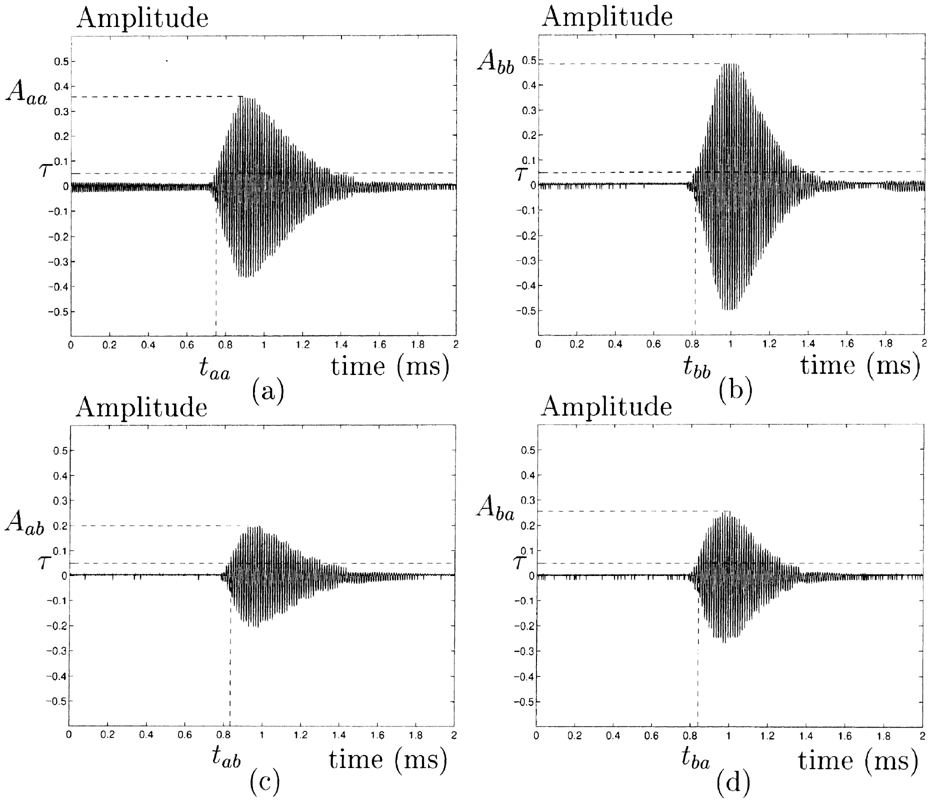

Fig. 4. Real sonar signals obtained from a planar target when (a) transducer *a* transmits and transducer *a* receives (b) transducer *b* transmits and *b* receives (c) transducer *a* transmits and *b* receives (d) transducer *b* transmits and *a* receives.

Experimentally obtained amplitude and TOF patterns of the target primitives are illustrated in Figs. 5 and 6 as a function of the target scan angle α. In these figures, the solid lines correspond to the average over multiple data sets. The amount of amplitude and TOF noise is also illustrated by plotting the $\pm 3\sigma_A$ and $\pm 3\sigma_t$ curves together with the average amplitude and TOF curves, respectively. Here, σ_A (or σ_t) is the amplitude (or TOF) noise standard deviation. To provide additional statistics on the repeatability of the sonar returns from each target type, the values of σ_A and σ_t are presented in Table 1 for $r = 45$ cm, $\theta = 0°$ and $\alpha = 0°$ which corresponds to the center of the joint sensitivity region.

Table 1
The standard deviations σ_A and σ_t for each target type at $r = 45$ cm, $\theta = 0°$, and $\alpha = 0°$. The maximum signal amplitude is 1

Target type	σ_A	σ_t (ms)
Plane	0.017 (−35.4 dB)	0.150
Corner	0.029 (−30.8 dB)	0.005
Edge ($\theta_e = 90°$)	0.002 (−54.0 dB)	0.101
Acute corner ($\theta_c = 60°$)	0.030 (−30.5 dB)	0.035
Cylinder ($r_c = 2.5$ cm)	0.004 (−48.0 dB)	0.207
Cylinder ($r_c = 5.0$ cm)	0.007 (−43.1 dB)	0.198
Cylinder ($r_c = 7.5$ cm)	0.008 (−41.9 dB)	0.201

We considered the samples of the following 21 different signals as alternative inputs to the neural networks:

I_1: $A_{aa}(\alpha)$, $A_{bb}(\alpha)$, $[A_{ab}(\alpha) + A_{ba}(\alpha)]/2$, $t_{aa}(\alpha)$, $t_{bb}(\alpha)$, and $[t_{ab}(\alpha) + t_{ba}(\alpha)]/2$

I_2: $A_{aa}(\alpha) - A_{ab}(\alpha)$, $A_{bb}(\alpha) - A_{ba}(\alpha)$, $t_{aa}(\alpha) - t_{ab}(\alpha)$, and $t_{bb}(\alpha) - t_{ba}(\alpha)$

I_3: $[A_{aa}(\alpha) - A_{ab}(\alpha)][A_{bb}(\alpha) - A_{ba}(\alpha)]$, $[A_{aa}(\alpha) - A_{ab}(\alpha)] + [A_{bb}(\alpha) - A_{ba}(\alpha)]$, $[t_{aa}(\alpha) - t_{ab}(\alpha)][t_{bb}(\alpha) - t_{ba}(\alpha)]$, and $t_{aa}(\alpha) - t_{ab}(\alpha)] + [t_{bb}(\alpha) - t_{ba}(\alpha)]$

I_4–I_6: Magnitude of the discrete Fourier transform $[|\mathscr{F}(I_i)|,\ i = 1,\ 2,\ 3]$

I_7–I_9: Phase of the discrete Fourier transform $[\angle\mathscr{F}(I_i),\ i = 1,\ 2,\ 3]$

I_{10}–I_{18}: Discrete wavelet transform of I_1, I_2, I_3 at different resolutions

I_{19}–I_{21}: Features extracted by using Kohonen's self-organizing feature map [SOFM(I_i), $i = 1,\ 2,\ 3$]

To the best of our knowledge, these input signals have not been used earlier for target classification with sonar. The first signal I_1 is taken as the original form of the patterns without any processing, except for averaging the cross terms. [$A_{ab}(\alpha)$ is averaged with $A_{ba}(\alpha)$, and $t_{ab}(\alpha)$ is averaged with $t_{ba}(\alpha)$. Since these cross terms should ideally be equal, their averages are more representative.] The choice of

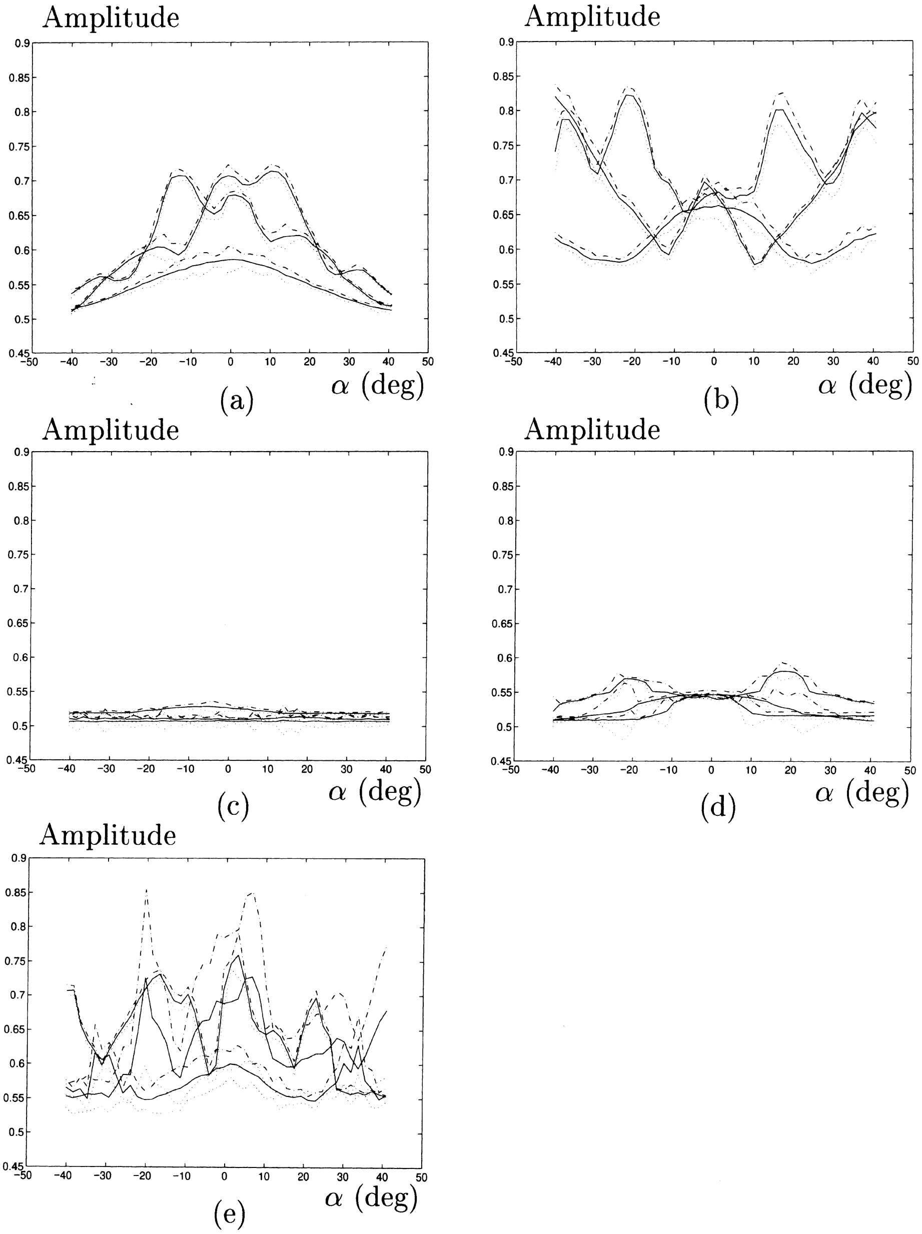

Fig. 5. Amplitude characteristics which incorporate the amplitude noise (±$3\sigma_A$) for the targets: (a) plane; (b) corner; (c) edge with $\theta_e = 90°$; (d) cylinder with $r_c = 5$ cm; and (e) acute corner with $\theta_c = 60°$. Here, solid, dashed, and dotted lines correspond to the average over eight data sets, average + $3\sigma_A$ and average − $3\sigma_A$, respectively. (© 2000 IEEE)

the second signal I_2 has been motivated by the target differentiation algorithm developed by the authors (Ayrulu & Barshan, 1998) and used with neural network classifiers in Barshan et al. (2000). The third input signal I_3 is motivated by the differential terms which are used to assign belief values to the target types classified by the target differentiation algorithm (Ayrulu & Barshan, 1998). These three input signals have been used both in their raw form and after taking discrete Fourier and wavelet transforms, as well as after feature extraction by Kohonen's self-organizing feature map. Since complex numbers cannot be given as input to neural network classifiers, magnitude and phase of the Fourier transform of each signal are used separately [$|\mathscr{F}(I_i)|$ and $\angle\mathscr{F}(I_i)$, $i = 1, 2, 3$]. It is observed that although simultaneous use of magnitude and phase of the Fourier transform makes the neural network structure more complicated, it does not bring much improvement in target classification. Next, discrete wavelet transforms of each signal at different resolution levels j are used. Initially, wavelet transform of each signal at resolution level $j = -1$ is used as the input [DWT(I_i), $i = 1, 2, 3$]. Secondly, only the low-frequency component of the wavelet transform, c_{-1}s, are employed [LFC(DWT(I_i))$_1$]. Finally, the low-frequency component of wavelet transform at resolution $j = -2$,

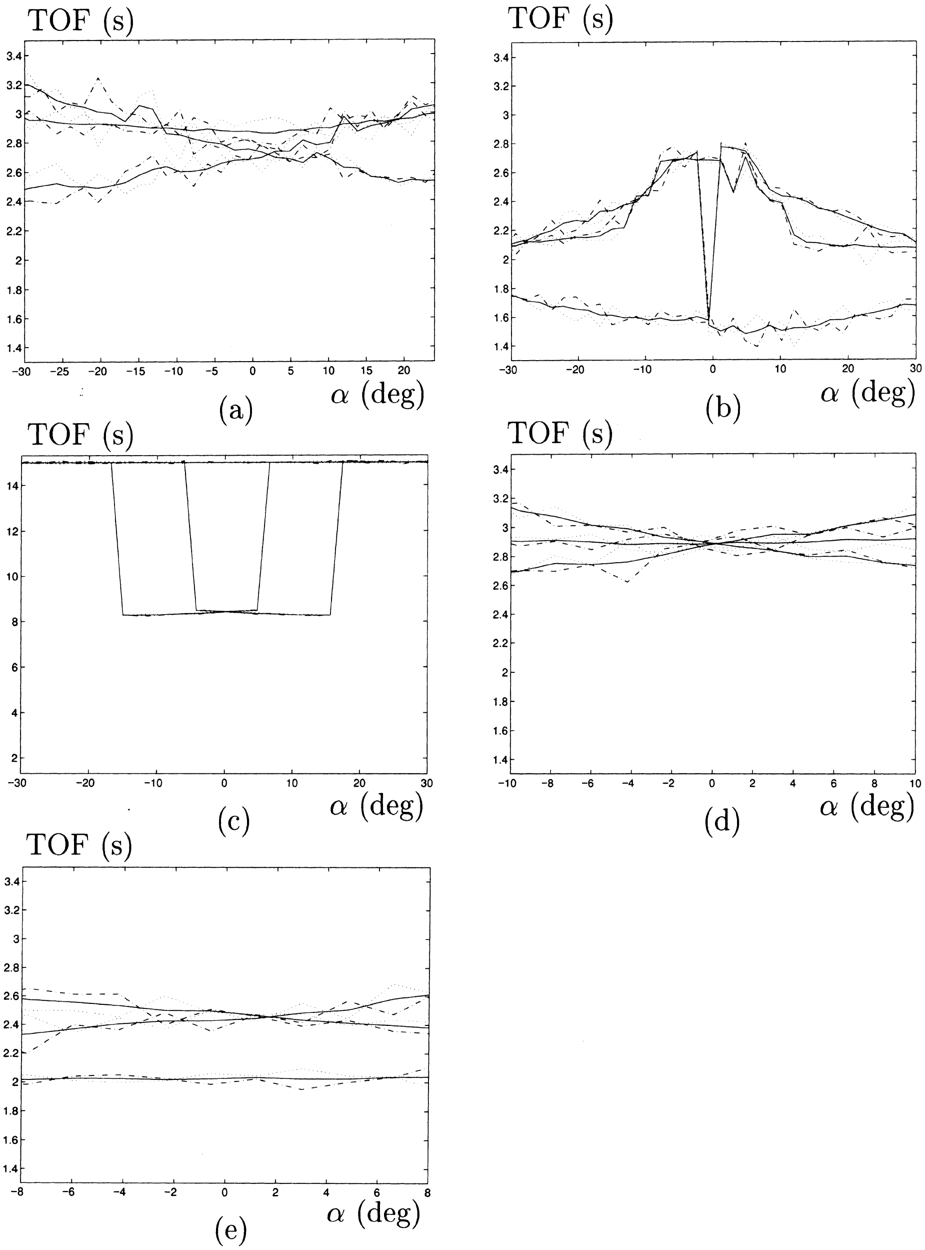

Fig. 6. TOF characteristics which incorporate the TOF noise ($\pm 3\sigma_t$) for the targets: (a) plane; (b) corner; (c) edge with $\theta_e = 90°$; (d) cylinder with $r_c = 5$ cm; and (e) acute corner with $\theta_c = 60°$. Here, solid, dashed, and dotted lines correspond to the average over eight data sets, average + $3\sigma_t$ and average − $3\sigma_t$, respectively. (© 2000 IEEE)

c_{-2}s, are used [LFC(DWT(I_i))$_2$]. The low-frequency components of the wavelet transform are more similar to the original signal. When the resolution is further decreased, the performance of the networks deteriorates since the number of samples in the low-frequency component decreases with decreasing resolution level j. For this reason, we have stopped at resolution $j = -2$. While obtaining these wavelet transforms, original signal samples are taken as c_0, and the scaling filter whose first 12 coefficients are given in Table 2 is used. Note that this filter is symmetrical with respect to $n = 0$. Finally, the features extracted by using Kohonen's self-organizing feature map are used as input signals [SOFM(I_i), $i = 1, 2, 3$]. In this case, the extracted features are used both prior to neural networks trained by the back-propagation algorithm and prior to linear classifiers designed by using a least-squares approach.

7. Experimental studies

The aim of this study is to employ neural networks to identify and resolve parameter relations embedded in the characteristics of sonar echo returns from all target types considered, in a robust and compact manner in real time.

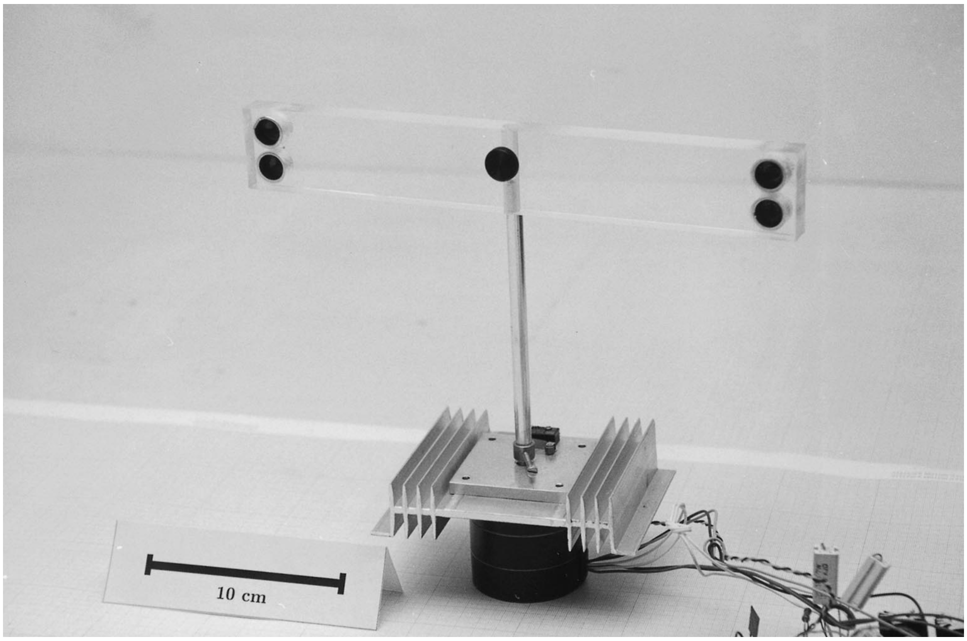

Fig. 7. Configuration of the Panasonic transducers in the real sonar system. The two transducers on the left collectively constitute one transmitter/receiver. Similarly, those on the right constitute another.

Performance of neural network classifiers is affected by the choice of parameters related to the network structure, training algorithm, and input signals, as well as parameter initialization (Alpaydın, 1993). In this work, various input signal representations described in the previous section and two different training algorithms, reviewed in Section 4, are considered to improve the performance of neural networks in target classification and localization with sonar.

The transducers used in our experimental setup are Panasonic transducers that have a much larger beamwidth than the more commonly used Polaroid transducers (Panasonic Corporation, 1989; Polaroid Corporation, 1997). The aperture radius of the Panasonic transducer is $a = 0.65$ cm, its resonance frequency is $f_0 = 40$ kHz, and therefore $\theta_0 \cong 54°$ for these transducers (Fig. 1). In the experiments, separate transmitting and receiving elements with a small vertical spacing have been used, rather than a single transmitting-receiving transducer. This is because, unlike Polaroid transducers, Panasonic transducers are manufactured as separate transmitting and receiving units (Fig. 7). The horizontal center-to-center separation of the transducers used in the experiments is $d = 25$ cm. The entire sensing unit (or the sensor node) is mounted on a small 6 V 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 PC 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 to extract the TOF information. The amplitude information is obtained 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 of $\theta_e = 90°$ and an acute corner of $\theta_c = 60°$. Amplitude and TOF patterns of these targets are collected with the sensing unit described above 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. 8). The target primitive located at range r and azimuth θ is scanned by the sensing unit for scan angle $-52° \leq \alpha \leq 52°$ with 1.8° increments. The reason for using a wider range for the scan angle is the possibility that a target may still generate returns outside of the range of θ. The angle α is always measured with respect to $\theta = 0°$ regardless of target location (r, θ). (That is, $\theta = 0°$ and $\alpha = 0°$ always coincide.)

Table 2
First 12 coefficients of the scaling filter $h(n)$ which is symmetrical with respect to the origin

n	$h(n)$
0	0.542
1	0.307
2	− 0.035
3	− 0.078
4	0.023
5	− 0.030
6	0.012
7	− 0.013
8	0.006
9	0.006
10	− 0.003
11	− 0.002

Table 3
Number of neurons used in the input, hidden and output layers of the non-modular networks trained with the back-propagation algorithm

Input signal	Input	Hidden	Output
I_1	348	120	21
I_2	232	100	21
I_3	232	105	21
$\|\mathcal{F}(I_1)\|$	348	175	21
$\angle\mathcal{F}(I_1)$	348	125	21
$\|\mathcal{F}(I_2)\|$	232	125	21
$\angle\mathcal{F}(I_2)$	232	135	21
$\|\mathcal{F}(I_3)\|$	232	135	21
$\angle\mathcal{F}(I_3)$	232	135	21
DWT(I_1)	384	50	21
LFC(DWT(I_1))$_1$	192	50	21
LFC(DWT(I_1))$_2$	96	66	21
DWT(I_2)	256	50	21
LFC(DWT(I_2))$_1$	128	68	21
LFC(DWT(I_2))$_2$	64	54	21
DWT(I_3)	256	65	21
LFC(DWT(I_3))$_1$	128	64	21
LFC(DWT(I_3))$_2$	64	62	21
SOFM(I_1)	175	55	21
SOFM(I_2)	175	50	21
SOFM(I_3)	175	50	21

Table 4
Number of neurons used in the input, hidden and output layers of each modular network designed for target classification, r and θ estimation. Note that the number of input and output neurons of the modules are equal

Input signal	Input	Target type	r	θ	Output
I_1	348	90	125	75	7
I_2	232	25	49	30	7
I_3	232	30	65	45	7
$\|\mathcal{F}(I_1)\|$	348	155	180	175	7
$\angle\mathcal{F}(I_1)$	348	110	155	120	7
$\|\mathcal{F}(I_2)\|$	232	90	120	150	7
$\angle\mathcal{F}(I_2)$	232	110	150	130	7
$\|\mathcal{F}(I_3)\|$	232	110	120	145	7
$\angle\mathcal{F}(I_3)$	232	135	150	140	7
DWT(I_1)	384	30	60	45	7
LFC(DWT(I_1))$_1$	192	35	55	50	7
LFC(DWT(I_1))$_2$	96	50	95	80	7
DWT(I_2)	256	34	37	30	7
LFC(DWT(I_2))$_1$	128	26	35	35	7
LFC(DWT(I_2))$_2$	64	45	75	36	7
DWT(I_3)	256	50	50	50	7
LFC(DWT(I_3))$_1$	128	35	40	38	7
LFC(DWT(I_3))$_2$	64	58	80	50	7
SOFM(I_1)	175	34	55	45	7
SOFM(I_2)	175	30	50	40	7
SOFM(I_3)	175	30	50	42	7

For the given scan range and motor step size, $58 = (2 \times 52°/1.8°)$ angular samples of each of amplitude and TOF patterns $[A_{aa}(\alpha), A_{bb}(\alpha), A_{ab}(\alpha), A_{ba}(\alpha);\ t_{aa}(\alpha), t_{bb}(\alpha), t_{ab}(\alpha), t_{ba}(\alpha)]$ are acquired at each target location. Four similar sets of scans are collected for each target primitive at each location, resulting in 700 ($= 4$ data sets $\times$ 25 locations $\times$ 7 target types) sets of signals to be used for training. Neural networks trained with the back-propagation algorithm consist of one input, one hidden, and one output layer. The number of input layer neurons is determined by the total number of samples of the amplitude and TOF patterns used by a particular type of input signal, described in Section 6. These numbers for the networks trained with the back-propagation algorithm are listed in Tables 3 and 4. For example, for the input signal I_1, the original forms of the amplitude and TOF patterns are used without any processing, except for averaging the cross terms as explained in the previous section. After averaging, there are six patterns each with 58 samples; therefore 348 ($= 6 \times 58$) input units are used. For the second input signal I_2, four amplitude and TOF differentials are used, therefore 232 ($= 4 \times 58$) input units are needed. Similarly, for the input signal I_3, there are also four input patterns and 232 is the number of input neurons. When the Fourier transforms of I_1, I_2, and I_3 are taken, the resulting signal has the same number of samples as the original signal. For the wavelet transform, the number of samples used needs to be a power of two. Therefore, the number of samples (58) is increased to 64 by padding with zeroes. In this case, for DWT(I_1), we have $6 \times 64 = 384$, for DWT(I_2) and DWT(I_3), we have $4 \times 64 = 256$ input units to the neural network. For Kohonen's self-organizing feature-mapping algorithm, a two-dimensional output layer (7×25) is used which is presented as input to the neural network. Therefore, 175 ($= 7 \times 25$) input layer neurons are needed. The number of hidden layer neurons is determined by *enlarging*. 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 azimuth θ of the target with respect

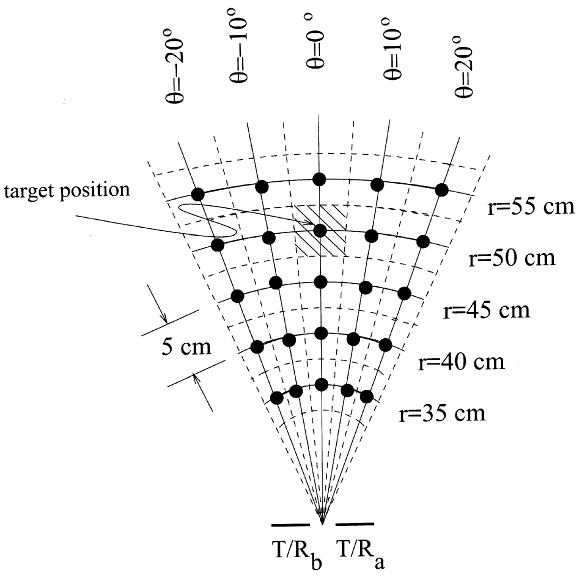

Fig. 8. Discrete network training locations. (© 2000 IEEE)

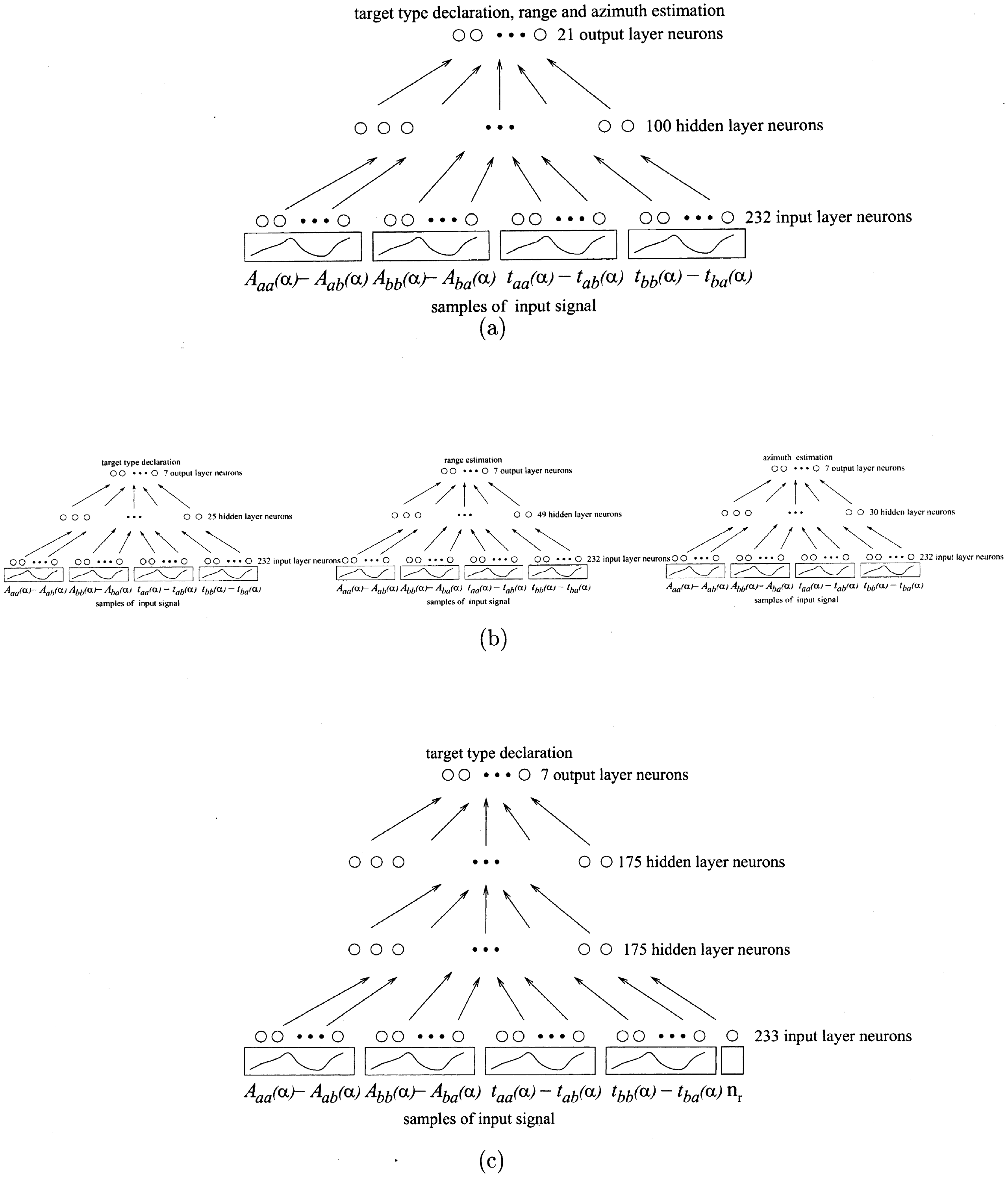

Fig. 9. The structure of the (a) non-modular and (b) modular networks trained with the back-propagation algorithm; (c) non-modular network trained with the generating–shrinking algorithm when the input signal I_2 is used.

to the line-of-sight of the sensing unit, which is also binary coded with resolution 0.5°.

In addition, modular network structures for each type of input signal have been implemented in which three separate networks for target type, range, and azimuth, each trained with the back-propagation algorithm, are employed. The different network structures implemented in this study are illustrated in Fig. 9 for the input signal I_2. In the modular case, each of the three modules has the same number of input layer neurons as the corresponding non-modular

Table 5
Average percentages of correct classification, range (r), and azimuth (θ) estimation for non-modular networks with different input signals. The numbers given in parentheses are the results when the test objects are located arbitrarily at non-grid locations (in continuous estimation space), whereas the numbers before the parentheses are for when the grid positions are used for testing

Input signal	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		Error tolerance ϵ_r				Error tolerance ϵ_θ			
		± 0.125 cm	± 1 cm	± 5 cm	± 10 cm	± 0.25°	± 2°	± 10°	± 20°
I_1	88 (88)	30 (17)	41 (32)	63 (55)	86 (78)	65 (37)	76 (47)	87 (75)	97 (91)
I_2	95 (90)	74 (59)	77 (63)	87 (78)	93 (88)	89 (70)	92 (75)	95 (92)	97 (94)
I_3	86 (58)	79 (63)	82 (63)	89 (76)	94 (83)	83 (66)	89 (74)	95 (93)	97 (94)
$\lvert\mathscr{F}(I_1)\rvert$	83 (83)	33 (21)	41 (31)	66 (60)	85 (81)	37 (20)	43 (26)	71 (62)	89 (86)
$\angle\mathscr{F}(I_1)$	55 (40)	14 (10)	27 (20)	51 (48)	72 (72)	30 (18)	43 (28)	70 (65)	87 (65)
$\lvert\mathscr{F}(I_2)\rvert$	82 (82)	30 (21)	41 (29)	66 (60)	83 (80)	32 (17)	44 (26)	71 (59)	88 (81)
$\angle\mathscr{F}(I_2)$	56 (39)	19 (14)	29 (23)	53 (49)	76 (73)	28 (16)	39 (27)	65 (57)	83 (79)
$\lvert\mathscr{F}(I_3)\rvert$	71 (52)	26 (19)	35 (29)	58 (56)	83 (80)	28 (16)	37 (23)	61 (51)	82 (75)
$\angle\mathscr{F}(I_3)$	52 (33)	15 (9)	23 (19)	52 (50)	73 (73)	28 (16)	39 (23)	60 (52)	80 (74)
DWT(I_1)	82 (82)	15 (12)	30 (24)	59 (50)	80 (76)	46 (26)	58 (37)	77 (64)	94 (87)
LFC(DWT(I_1))$_1$	85 (85)	18 (11)	28 (22)	58 (50)	82 (75)	54 (33)	65 (41)	80 (70)	95 (87)
LFC(DWT(I_1))$_2$	98 (98)	71 (60)	76 (60)	87 (76)	95 (91)	90 (71)	93 (77)	97 (96)	100 (96)
DWT(I_2)	92 (92)	63 (53)	69 (53)	84 (72)	93 (85)	85 (65)	88 (67)	93 (87)	96 (92)
LFC(DWT(I_2))$_1$	95 (91)	65 (53)	70 (53)	84 (70)	94 (80)	87 (68)	90 (72)	94 (91)	97 (91)
LFC(DWT(I_2))$_2$	89 (86)	28 (16)	34 (28)	58 (51)	84 (80)	58 (33)	68 (40)	86 (74)	95 (86)
DWT(I_3)	86 (82)	58 (49)	62 (53)	76 (68)	93 (78)	85 (57)	88 (63)	93 (85)	96 (87)
LFC(DWT(I_3))$_1$	82 (80)	56 (52)	60 (52)	75 (68)	89 (80)	73 (60)	77 (67)	86 (85)	93 (88)
LFC(DWT(I_3))$_2$	83 (80)	29 (21)	37 (30)	63 (60)	83 (81)	53 (28)	65 (38)	78 (65)	87 (84)
SOFM(I_1)	75 (75)	17 (12)	25 (19)	49 (45)	80 (77)	64 (38)	67 (40)	81 (75)	90 (88)
SOFM(I_2)	78 (78)	22 (19)	28 (23)	59 (53)	88 (82)	69 (39)	73 (45)	86 (77)	92 (88)
SOFM(I_3)	66 (65)	24 (21)	30 (26)	57 (51)	84 (78)	51 (29)	54 (34)	78 (69)	89 (81)

network. The number of hidden layer neurons is again determined by enlarging and varies as shown in Table 4. The number of output layer neurons of each module is seven. Referring to Tables 3 and 4 for non-modular and modular network structures, the minimum number of total neurons in the network layers is obtained with the input signal LFC(DWT(I_2))$_2$ and the maximum number is obtained with the input signal $\lvert\mathscr{F}(I_1)\rvert$ for both cases.

Neural networks using the same input signals are also trained with the generating–shrinking algorithm. This algorithm can only be applied for target type classification since it is based on the assumption that only one output neuron takes the value one (the winning neuron) and the others are zero. For this reason, range and azimuth estimation cannot be made with this approach. In these networks, the number of input layer neurons for each type of input signal is determined as described above for back-propagation networks, except that there is an additional input neuron for the reference number n_r. The reference number n_r is taken as 0.01, after making a number of simulations with n_r varying between 0.005 and 0.1. The output layer has seven neurons. Initially, each of the two hidden layers has 700 neurons (equal to the number of training patterns) which is reduced by one fourth to 174 or 175 after training. Since the numbers of neurons in the two hidden layers are approximately equal (174 or 175) and the number of output neurons is fixed for all types of input signals, the complexity of these networks can be assessed by the number of their input neurons.

The networks are tested as follows. Each target primitive is placed in turn in each of the 25 training positions shown in Fig. 8. Four sets of patterns are collected for each combination of target type and location, again resulting in 700 sets of experimentally acquired patterns. Based on these data, neural networks trained with the back-propagation algorithm estimate the target type, range, and azimuth; those trained with the generating–shrinking algorithm determine only the target type. The test data are not collected at the same time as the training data. Rather, each target is first moved through all the grid locations and a complete training set is fully completed (700 sets of patterns). The test data for the grid locations are obtained later by repositioning the objects at the grid locations and acquiring another 700 sets of patterns. This means that there will inevitably be some differences in the object positions and orientations, as well as the ambient conditions (i.e. temperature and humidity) even though the targets are nominally placed at the same grid points. In the testing stage, the targets are not presented to the sensing node following the same order used in training. Rather, a random strategy is followed.

For those networks trained with the back-propagation algorithm, the resulting average percentages over all target types for correct type classification, correct range and correct azimuth estimation are given in Tables 5 and 6. 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. Referring to Table 5, the highest average

Table 6
Average percentages of correct classification, range (r), and azimuth (θ) estimation for modular networks with different input signals

Input signal	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		Error tolerance ϵ_r				Error tolerance ϵ_θ			
		± 0.125 cm	± 1 cm	± 5 cm	± 10 cm	± 0.25°	± 2°	± 10°	± 20°
I_1	88 (88)	33 (18)	46 (30)	70 (56)	87 (83)	65 (38)	72 (47)	84 (74)	97 (94)
I_2	95 (93)	73 (60)	88 (69)	93 (83)	96 (88)	95 (71)	96 (76)	97 (97)	99 (98)
I_3	88 (59)	73 (60)	75 (62)	83 (76)	91 (85)	87 (69)	91 (73)	95 (93)	98 (97)
$\lvert\mathscr{F}(I_1)\rvert$	84 (84)	34 (26)	40 (33)	64 (61)	84 (83)	39 (21)	47 (30)	68 (53)	85 (79)
$\angle\mathscr{F}(I_1)$	59 (43)	15 (10)	25 (20)	52 (49)	77 (74)	28 (18)	41 (27)	66 (58)	82 (77)
$\lvert\mathscr{F}(I_2)\rvert$	76 (76)	29 (18)	38 (28)	63 (58)	83 (81)	31 (16)	43 (23)	69 (54)	87 (80)
$\angle\mathscr{F}(I_2)$	59 (38)	16 (12)	27 (20)	53 (48)	77 (73)	30 (17)	42 (26)	64 (57)	83 (80)
$\lvert\mathscr{F}(I_3)\rvert$	73 (51)	29 (19)	36 (30)	61 (57)	83 (81)	32 (18)	43 (27)	65 (50)	83 (72)
$\angle\mathscr{F}(I_3)$	50 (22)	15 (12)	24 (20)	50 (45)	74 (70)	27 (16)	41 (27)	59 (59)	82 (82)
DWT(I_1)	74 (74)	21 (14)	27 (20)	59 (53)	82 (79)	51 (29)	63 (38)	80 (64)	94 (89)
LFC(DWT(I_1))$_1$	98 (98)	21 (13)	33 (22)	59 (53)	79 (75)	49 (31)	62 (43)	79 (71)	94 (91)
LFC(DWT(I_1))$_2$	99 (99)	80 (64)	82 (64)	91 (79)	96 (89)	92 (72)	93 (77)	98 (94)	100 (95)
DWT(I_2)	96 (93)	64 (54)	69 (57)	82 (71)	92 (81)	87 (66)	90 (69)	94 (90)	96 (92)
LFC(DWT(I_2))$_1$	97 (94)	66 (56)	71 (56)	84 (71)	91 (80)	88 (66)	90 (70)	94 (88)	96 (90)
LFC(DWT(I_2))$_2$	84 (80)	32 (20)	44 (29)	68 (60)	88 (79)	53 (28)	61 (34)	80 (72)	92 (88)
DWT(I_3)	89 (85)	58 (51)	62 (52)	76 (67)	89 (81)	76 (59)	80 (65)	88 (85)	94 (88)
LFC(DWT(I_3))$_1$	91 (86)	61 (54)	66 (54)	78 (65)	87 (77)	79 (62)	83 (68)	89 (86)	94 (90)
LFC(DWT(I_3))$_2$	79 (78)	33 (20)	44 (32)	69 (62)	88 (83)	41 (23)	52 (31)	75 (66)	89 (84)
SOFM(I_1)	74 (73)	14 (10)	23 (18)	46 (41)	72 (69)	61 (34)	64 (37)	79 (69)	89 (86)
SOFM(I_2)	76 (76)	19 (16)	28 (21)	57 (52)	81 (78)	66 (38)	71 (42)	85 (76)	93 (87)
SOFM(I_3)	63 (61)	21 (19)	31 (25)	55 (51)	81 (73)	49 (27)	51 (33)	75 (67)	87 (80)

percentage of correct classification of 98% is obtained with the input signal LFC(DWT(I_1))$_2$. The highest average percentage of correct azimuth estimation is achieved with the same input signal, and lies in the range 90–100% (depending on the error tolerance level ϵ_θ). The highest average percentage of correct range estimation lies in the range 79–89% and is obtained with the input signal I_3. This is followed by the input signals I_2 and LFC(DWT(I_1))$_2$. Statistics over 10 non-modular networks trained with the back-propagation algorithm using different initial conditions for the connection weights are provided in Table 7 for the input signal I_2.

For modular networks (Table 6), the highest average percentage of correct classification is 99%, the highest average percentage of range estimation for $\epsilon_r = 0.125$ cm is 80%, and that for correct azimuth estimation for $\epsilon_\theta = 10$ and 20° are 98 and 100%, all of which are obtained with the input signal LFC(DWT(I_1))$_2$. The highest average percentage of correct range estimation for the remaining error tolerances lies in the range 88–96% and that for correct azimuth estimation for $\epsilon_\theta = 0.25$ and 2° are 95 and 96% for the input signal I_2 (Table 6). For both modular and non-modular cases, the average percentages obtained using the magnitude or the phase of the Fourier transform are always much less than those obtained with the corresponding untransformed signals. In addition, the classification and localization capability of the networks employing the magnitude of the Fourier transform always outperforms that of networks employing phase information. When wavelet transformed signals are used, the results are comparable to the results of the original signal. However, employing only the low-frequency component of the wavelet transform at the resolution level $j = -1$ (i.e. c_{-1}) results in better classification and estimation performance than employing both c_{-1} and d_{-1}. While classification and estimation performance further increases by using the low-frequency component of the wavelet transform at the resolution level $j = -2$ for the input signal I_1, they decrease for I_2 and I_3.

Table 7
The mean and the standard deviation of the average percentages of correct classification, range (r), and azimuth (θ) estimation over 10 non-modular networks trained with the back-propagation algorithm using different initial conditions for the connection weights. Input signal I_2 is used

	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		Error tolerance ϵ_r				Error tolerance ϵ_θ			
		± 0.125 cm	± 1 cm	± 5 cm	± 10 cm	± 0.25°	± 2°	± 10°	± 20°
mean	95 (92)	73 (59)	78 (61)	88 (76)	96 (85)	90 (69)	93 (73)	97 (94)	100 (94)
std	2.8 (3.2)	4.2 (2.3)	3.1 (2.3)	1.7 (1.5)	1.7 (2.0)	2.8 (1.8)	1.8 (1.4)	1.1 (2.6)	1.1 (1.2)

Table 8
Average percentages of correct classification, range (r), and azimuth (θ) estimation for Kohonen's self-organizing feature map used prior to a linear classifier

Input signal	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		Error tolerance ϵ_r				Error tolerance ϵ_θ			
		± 0.125 cm	± 1 cm	± 5 cm	± 10 cm	± 0.25°	± 2°	± 10°	± 20°
SOFM(I_1)	81 (81)	33 (21)	37 (27)	61 (55)	85 (79)	75 (65)	76 (68)	88 (88)	94 (91)
SOFM(I_2)	85 (85)	41 (26)	44 (30)	71 (59)	90 (84)	80 (65)	82 (68)	93 (88)	97 (88)
SOFM(I_3)	73 (73)	42 (28)	45 (34)	69 (60)	86 (78)	64 (58)	67 (63)	85 (81)	94 (84)

The results obtained with Kohonen's self-organizing feature map used prior to linear classifiers are given in Table 8. This combination results in better classification performance than when the self-organizing feature map is applied prior to neural networks, the results of which are given in the last three rows of Tables 5 and 6. The classification and azimuth estimation performance of a linear classifier using features extracted by Kohonen's self-organizing feature map are also comparable to those obtained with corresponding unprocessed signals. However, range estimation performance deteriorates dramatically compared to the results obtained with the corresponding unprocessed signals (Table 8).

For networks trained with the generating–shrinking algorithm, the resulting average percentages of correct target classification over all target types are given in Table 9. Referring to this table, maximum average percentage of correct classification is 97% which is obtained with the input signals LFC(DWT(I_1))$_1$ and LFC(DWT(I_1))$_2$. In this case, resulting percentages (73–97%) are almost comparable for all input signals except the features obtained by using Kohonen's self-organizing feature map which are much lower (≤13%). Unlike networks trained with the back-propagation algorithm, a substantial deterioration in performance is not observed when Fourier transformed signals are used as input.

Table 9
Average percentages of correct classification for networks trained with the generating–shrinking algorithm for the different input signals

Input signal	Correct classif. (%)
I_1	95 (95)
I_2	90 (90)
I_3	76 (76)
$\lvert\mathcal{F}(I_1)\rvert$	93 (93)
$\angle\mathcal{F}(I_1)$	86 (85)
$\lvert\mathcal{F}(I_2)\rvert$	87 (87)
$\angle\mathcal{F}(I_2)$	83 (80)
$\lvert\mathcal{F}(I_3)\rvert$	73 (73)
$\angle\mathcal{F}(I_3)$	77 (74)
DWT(I_1)	95 (95)
LFC(DWT(I_1))$_1$	97 (97)
LFC(DWT(I_1))$_2$	97 (97)
DWT(I_2)	91 (91)
LFC(DWT(I_2))$_1$	90 (90)
LFC(DWT(I_2))$_2$	90 (90)
DWT(I_3)	75 (75)
LFC(DWT(I_3))$_1$	77 (77)
LFC(DWT(I_3))$_2$	80 (80)
SOFM(I_1)	5 (8)
SOFM(I_2)	13 (11)
SOFM(I_3)	8 (5)

The networks are also tested for targets situated arbitrarily in the continuous estimation space and not necessarily confined to the 25 locations of Fig. 8. This second set of test data was acquired with about a month's delay after collecting the training data. Randomly generated locations within the area shown in Fig. 8, not necessarily corresponding to one of the 25 grid locations, are used as target positions. The r, θ values corresponding to these locations are generated by using the uniform random number generator in MATLAB. The range for r is [32.5 cm, 57.5 cm] and that for θ is [−25°, 25°]. The results given in parentheses in the corresponding tables (Tables 5–9) are for this second case where the test points are randomly chosen from a continuum. The maximum correct target classification percentages of 98% (non-modular network structure) and 99% (modular structure) are maintained when the input signal LFC(DWT(I_1))$_2$ is used. These values are the same as those achieved at the grid positions. The best performance of LFC(DWT(I_1))$_2$ is followed by I_2 and I_3 when target classification and localization are considered together. As expected, the percentages for the non-grid test positions can be lower than those for the grid test positions by 0–30 percentage points; the networks give the best results when a test target is situated exactly at one of the training sites. Noting that the networks were trained only at 25 locations and at grid spacings of 5 cm and 10°, it can be concluded from the percentage of correct range and azimuth estimates obtained at error tolerances of $|\epsilon_r| = 0.125$ and 1 cm and $|\epsilon_\theta| = 0.25$ and 2°, that the networks demonstrate 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. 8. The correct target type percentages in the corresponding tables are quite high and the accuracy of the range/azimuth estimates would be acceptable for most of the input signals in many applications. If better estimates are required, this can be achieved by reducing the training grid spacing in Fig. 8. Moreover, these percentages for the modular network

Table 10
Average percentages of correct classification, range [r], and azimuth [θ]) estimation with different input signals when targets not scanned during training (rough plane, edge of $\theta_e = 60°$, rough cylinder with $r_c = 7.5$ cm, and two smooth cylinders with $r_c = 4$ and 10 cm) are used for testing. The results in the brackets are for modular networks, whereas those before the brackets are for non-modular networks

Input signal	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		Error tolerance ϵ_r				Error tolerance ϵ_θ			
		± 0.125 cm	± 1 cm	± 5 cm	± 10 cm	± 0.25°	± 2°	± 10°	± 20°
I_1	85 [73]	18 [21]	28 [32]	49 [55]	74 [76]	35 [40]	45 [45]	61 [56]	80 [72]
I_2	80 [80]	59 [60]	59 [65]	72 [77]	83 [84]	68 [70]	73 [75]	75 [76]	76 [80]
I_3	57 [54]	60 [59]	60 [59]	69 [69]	80 [80]	64 [68]	72 [75]	73 [78]	74 [79]
$\|\mathscr{F}(I_1)\|$	78 [79]	19 [23]	28 [31]	52 [58]	74 [83]	22 [23]	26 [33]	51 [57]	73 [80]
$\angle\mathscr{F}(I_1)$	41 [41]	12 [12]	23 [20]	48 [45]	70 [70]	20 [18]	30 [29]	53 [48]	75 [71]
$\|\mathscr{F}(I_2)\|$	75 [78]	17 [18]	26 [25]	48 [51]	70 [75]	20 [19]	29 [29]	55 [53]	77 [74]
$\angle\mathscr{F}(I_2)$	34 [35]	13 [12]	20 [21]	44 [45]	70 [70]	17 [18]	25 [27]	48 [50]	69 [70]
$\|\mathscr{F}(I_3)\|$	46 [47]	16 [18]	25 [25]	51 [50]	79 [73]	18 [20]	24 [29]	47 [50]	71 [75]
$\angle\mathscr{F}(I_3)$	34 [32]	12 [11]	19 [19]	46 [45]	69 [69]	17 [17]	26 [25]	47 [45]	71 [70]
DWT(I_1)	78 [75]	12 [15]	23 [19]	50 [51]	75 [80]	27 [32]	35 [45]	53 [62]	79 [80]
LFC(DWT(I_1))$_1$	69 [84]	12 [14]	18 [27]	47 [50]	78 [70]	32 [33]	41 [40]	59 [60]	80 [77]
LFC(DWT(I_1))$_2$	85 [83]	56 [63]	58 [63]	68 [74]	82 [85]	67 [71]	69 [75]	76 [80]	76 [80]
DWT(I_2)	82 [80]	53 [54]	54 [55]	71 [68]	85 [83]	65 [69]	69 [73]	77 [75]	79 [73]
LFC(DWT(I_2))$_1$	80 [84]	53 [55]	56 [56]	69 [68]	79 [79]	67 [68]	71 [72]	78 [73]	78 [73]
LFC(DWT(I_2))$_2$	76 [74]	16 [19]	22 [28]	48 [51]	75 [74]	32 [33]	39 [39]	59 [57]	73 [73]
DWT(I_3)	73 [75]	49 [50]	49 [50]	59 [63]	75 [75]	67 [61]	72 [67]	76 [76]	79 [78]
LFC(DWT(I_3))$_1$	74 [80]	50 [52]	50 [52]	60 [63]	73 [75]	62 [63]	68 [69]	73 [72]	76 [74]
LFC(DWT(I_3))$_2$	73 [72]	17 [20]	26 [30]	51 [52]	73 [75]	30 [23]	40 [32]	56 [52]	72 [68]
SOFM(I_1)	73 [72]	9 [7]	13 [12]	35 [33]	60 [56]	32 [31]	34 [32]	51 [50]	65 [65]
SOFM(I_2)	75 [74]	17 [15]	21 [21]	56 [55]	85 [81]	44 [43]	47 [46]	67 [66]	76 [76]
SOFM(I_3)	66 [64]	16 [14]	19 [20]	47 [46]	73 [71]	32 [31]	36 [35]	60 [59]	83 [82]

structures are slightly better than those for neural networks in which type classification and range and azimuth estimation are done simultaneously. When arbitrary test positions are used, the decreases in the percentages of the networks trained by employing the generating–shrinking algorithm are much smaller than those of the modular and non-modular structures trained by employing the back-propagation algorithm. Unlike the latter case, for most of the input signal representations, the two results are identical. In a few cases, there is ±1–3% difference. The highest classification rate of 97%, which is identical for both grid and non-grid testing, is again obtained with the low-frequency components of the discrete wavelet transform for the input signals LFC(DWT(I_1))$_1$ and LFC(DWT(I_1))$_2$.

We have carried out further tests with the same system using targets not scanned during training, which are slightly different in size, shape, or roughness than the targets used for training. These are two smooth cylinders of radii 4 and 10 cm, a cylinder of radius 7.5 cm covered with bubbled packing material, a 60° smooth edge, and a plane covered with bubbled packing material. The packing material with bubbles has a honeycomb pattern of uniformly distributed circular bubbles of diameter 1.0 cm and height 0.3 cm, with a center-to-center separation of 1.2 cm. The test data are collected at the 25 grid locations used for training. Testing is performed for both modular and non-modular networks. These results are presented in Tables 10–12. When the non-modular network trained with the back-propagation algorithm is tested, there is a decrease of 11.7 percentage points on the average of all the different input signals. This number is 6.7 percentage points for the modular network trained with the back-propagation algorithm. For the generating–shrinking

Table 11
Average percentages of correct classification, range (r), and azimuth (θ) estimation for Kohonen's self-organizing feature map used prior to a linear classifier when targets not scanned during training (rough plane, edge of $\theta_e = 60°$, rough cylinder with $r_c = 7.5$ cm, and two smooth cylinders with $r_c = 4$ and 10 cm)

Input signal	% of correct classif.	% of correct r estimation				% of correct θ estimation			
		Error tolerance ϵ_r				Error tolerance ϵ_θ			
		± 0.125 cm	± 1 cm	± 5 cm	± 10 cm	± 0.25°	± 2°	± 10°	± 20°
SOFM(I_1)	78	20	23	50	74	46	46	68	77
SOFM(I_2)	77	28	30	58	80	47	48	63	76
SOFM(I_3)	67	28	30	59	81	44	46	69	84

Table 12
Average percentages of correct classification for networks trained with the generating–shrinking algorithm for the different input signals when targets not scanned during training (rough plane, edge of $\theta_e = 60°$, rough cylinder with $r_c = 7.5$ cm, and two smooth cylinders with $r_c = 4$ and 10 cm)

Input signal	Correct classif. (%)
I_1	89
I_2	78
I_3	68
$\lvert\mathcal{F}(I_1)\rvert$	87
$\angle\mathcal{F}(I_1)$	83
$\lvert\mathcal{F}(I_2)\rvert$	83
$\angle\mathcal{F}(I_2)$	72
$\lvert\mathcal{F}(I_3)\rvert$	68
$\angle\mathcal{F}(I_3)$	69
DWT(I_1)	89
LFC(DWT(I_1))$_1$	89
LFC(DWT(I_1))$_2$	88
DWT(I_2)	80
LFC(DWT(I_2))$_1$	79
LFC(DWT(I_2))$_2$	79
DWT(I_3)	67
LFC(DWT(I_3))$_1$	68
LFC(DWT(I_3))$_2$	71
SOFM(I_1)	5
SOFM(I_2)	9
SOFM(I_3)	6

algorithm, the reduction is 7.2 points. When Kohonen's self-organizing feature map is used prior to a linear classifier, the average deterioration in performance is 5.7 points. Overall, we can conclude that the networks exhibit some degree of robustness to variations in target shape, size and roughness.

8. Conclusions

In this study, various input signal representations, two different training algorithms, and different network structures have been considered for neural networks for improved target classification and localization with sonar. The input signals are different functional forms of amplitude and TOF patterns acquired by a real sonar system, and in most cases they are preprocessed before being used as inputs to the neural networks. The preprocessing techniques employed are discrete Fourier and wavelet transforms and Kohonen's self-organizing feature map. Kohonen's self-organizing feature map is commonly used to extract the features of input data without supervision, resulting in scale-invariant classification. Here, it is used for feature extraction both prior to neural networks and also prior to a linear classifier. The performance of the different input signals are compared in terms of the successful classification and localization rates of the networks and their complexity. The training algorithms employed are back-propagation and generating–shrinking algorithms. The networks trained with the generating–shrinking algorithm can only be used for determining the correct target type. Networks with modular structures have also been trained with the back-propagation algorithm for target classification and localization. When the results for non-modular and modular networks are compared, it is observed that the results for modular networks are in general slightly better than the results for non-modular ones. In most cases, the low-frequency component of the wavelet transform of the signal I_1 at resolution level $j = -2$ results in better classification and localization performance.

The classification and localization capability of both non-modular and modular networks employing the magnitude of the Fourier transform always outperforms that of networks employing the phase information. However, such a substantial difference is not observed in the performance of networks trained with the generating–shrinking algorithm when the Fourier transforms are taken. For all input signals, the correct target differentiation rates of networks trained with the back-propagation and the generating–shrinking algorithms are comparable except when the features obtained by using Kohonen's self-organizing feature map are used as input. In this case, the success rate obtained with using the generating–shrinking algorithm is much lower (≤13%). Linear classifiers are also used to process the features extracted by Kohonen's self-organizing feature map and gave better results than processing the same features with neural networks. The minimum and maximum number of total neurons in the network layers is obtained with the input signals LFC(DWT(I_2))$_2$ and $\lvert\mathcal{F}(I_1)\rvert$, respectively, for both non-modular and modular networks.

Testing of the networks is performed both at the training locations and at arbitrary locations. As expected, the success rates for the non-grid test locations can be lower than those for the grid test positions by 0–30 percentage points; the networks give the best results when a test target is situated exactly at one of the training sites. Although trained on a discrete and relatively coarse grid, the networks are able to interpolate between the grid locations and offer higher resolution than that implied by the grid size. However, the interpolation capability of the networks generated with the generating–shrinking algorithm is much better. The correct estimation rates for target type, range and azimuth can be further increased by employing a finer grid for training.

For target differentiation based purely on raw data, the algorithm in Ayrulu and Barshan (1998) gives a correct differentiation percentage of 61%. In Utete, Barshan and Ayrulu (1999), 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 of combination 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%. However, using these two fusion methods, only planes, corners and acute corners can be differentiated. On the other hand, using the neural networks 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 (99%) than with the earlier-used decision rules employing multiple sensing nodes. The fact that the neural networks are able to distinguish all target types indicates that they must be making more effective use of the available data than the methods used earlier. The performance of the neural networks shows that the original training data set does contain sufficient information to differentiate the seven target types, but the other methods mentioned above are not able to resolve this identifying information. Neural networks are capable of differentiating more targets with increased accuracy by exploiting the hidden identifying features in the differential amplitude and TOF patterns of the targets. Furthermore, the networks are tested using targets not presented during training, which are somewhat different in size, shape or roughness than the targets used for training. The results indicate that the networks can reasonably successfully identify these modified targets, exhibiting some degree of robustness to variations in target shape, size and roughness.

There is scope for further application of neural networks to sonar, based on the facts that sonar data are difficult to interpret, that the physical models involved 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.

Acknowledgements

This research was supported by TÜBİTAK under grant 197E051. Figures 1, 2, 5, 6 and 8 have been reproduced with permission from IEEE Publication Services from Barshan, B., Ayrulu, B. & Utete, S. W. (2000). Neural network based target differentiation using sonar for robotics applications. *IEEE Transactions on Robotics and Automation*, *16* (4), 435–442.

References

Alpaydın, E. (1993). Multiple networks for function learning. *IEEE International Conference on Neural Networks* (pp. 9–14).

Au, W. W. L. (1994). Comparison of sonar discrimination—dolphin and artificial neural network. Part I. *The Journal of the Acoustical Society of America*, *95* (5), 2728–2735.

Au, W. W. L., Andersen, L. N., Rasmussen, A. R., Roitblat, H. L., & Nachtigall, P. E. (1995). Neural network modeling of a dolphin's sonar discrimination capabilities. *The Journal of the Acoustical Society of America*, *98* (1), 43–50.

Ayrulu, B., & Barshan, B. (1998). Identification of target primitives with multiple decision-making sonars using evidential reasoning. *International Journal of Robotics Research*, *17* (6), 598–623.

Bai, B. C., & Farhat, N. H. (1992). Learning networks for extrapolation and radar target identification. *Neural Networks*, *5* (3), 507–529.

Barshan, B. (1991). *A sonar-based mobile robot for bat-like prey capture*. PhD Thesis, Yale University, Department of Electrical Engineering, New Haven, CT.

Barshan, B., Ayrulu, B., & Utete, S. W. (2000). Neural network based target differentiation using sonar for robotics applications. *IEEE Transactions on Robotics and Automation*, *16* (4), 435–442.

Barshan, B., & Kuc, R. (1990). Differentiating sonar reflections from corners and planes by employing an intelligent sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12* (6), 560–569.

Bozma, Ö., & Kuc, R. (1991). Building a sonar map in a specular environment using single mobile sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13* (12), 1260–1269.

Bozma, Ö., & Kuc, R. (1994). A physical model-based analysis of heterogeneous environments using sonar—ENDURA method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16* (5), 497–506.

Bracewell, R. N. (1986). *The Fourier transform and its applications*, New York: McGraw-Hill.

Brown, M. K. (1986). The extraction of curved surface features with generic range sensors. *International Journal of Robotics Research*, *5* (1), 3–18.

Chang, W., Bosworth, B., & Carter, G. C. (1993). Results of using an artificial neural network to distinguish single echoes from multiple sonar echoes. Part I. *The Journal of the Acoustical Society of America*, *94* (3), 1404–1408.

Chen, Y. Q., Thomas, D. W., & Nixon, M. S. (1994). Generating–shrinking algorithm for learning arbitrary classification. *Neural Networks*, *7* (9), 1477–1489.

Chui, C. K. (1992). *An introduction to wavelets*, London: Academic Press.

Cohen, M., Franco, H., Morgan, N., Rumelhart, D., & Abrash, V. (1993). Context-dependent multiple distribution phonetic modelling with MLPs. In S. J. Hanson, J. D. Cowan & C. L. Giles, *Advances in neural information processing systems* (pp. 649–657). San Mateo, CA: Morgan Kaufmann.

Dror, I. E., Zagaeski, M., & Moss, C. F. (1995). 3-dimensional target recognition via sonar—a neural network model. *Neural Networks*, *8* (1), 149–160.

Galicki, M., Witte, H., Dörschel, J., Eiselt, M., & Griessbach, G. (1997). Common optimization of adaptive processing units and a neural network during the learning period: application in EEG pattern recognition. *Neural Networks*, *10* (6), 1153–1163.

Gorman, R. P., & Sejnowski, T. J. (1988). Learned classification of sonar targets using a massively parallel network. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *36* (7), 1135–1140.

Hauptmann, P. (1993). *Sensors: principles and applications*, Englewood Cliffs, New Jersey: Prentice Hall.

Haykin, S. (1994). *Neural networks: a comprehensive foundation*, New Jersey: Prentice Hall.

Jordan, M. A., & Jacobs, R. A. (1990). Learning to control an unstable system with forward modeling. In D. S. Touretzky, *Advances in neural information processing systems 2* (pp. 324–331). San Mateo, CA: Morgan Kaufmann.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43* (1), 59–69.

Kuc, R. (1993). Three-dimensional tracking using qualitative bionic sonar. *Robotics and Autonomous Systems*, *11* (2), 213–219.

Kuc, R. (1997). Biomimetic sonar recognizes objects using binaural information. *The Journal of the Acoustical Society of America*, *102* (2), 689–696.

Kuc, R., & Siegel, M. W. (1987). Physically-based simulation model for acoustic sensor robot navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-9* (6), 766–778.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In D. S. Touretzky, *Advances in neural information processing systems 2* (pp. 396–404). San Mateo, CA: Morgan Kaufmann.

Leonard, J. J., & Durrant-Whyte, H. F. (1992). *Directed sonar navigation*, London, UK: Kluwer Academic Press.

Lippman, R. P. (1987). An introduction to computing with neural networks. *IEEE ASSP Magazine*, 4–22.

Manyika, J., & Durrant-Whyte, H. F. (1994). *Data fusion and sensor management: a decentralized information-theoretic approach*, New York: Ellis Horwood.

Miller, R. K., & Walker, T. C. (1992). *Neural network applications and products*, Norcross, GA: SEAI Technical Publications.

Narendra, K. S., & Parthasarathy, K. (1991). Gradient methods for the optimization of dynamic systems containing neural networks. *IEEE Transactions on Neural Networks*, *2* (2), 252–262.

Ogawa, T., Kameyama, K., Kuc, R., & Kosugi, Y. (1996). Source localization with network inversion using an answer-in-weights scheme. *IEEE Transactions on Information and Systems*, *E79-D* (5), 608–619.

Panasonic Corporation (1989). *Ultrasonic microphones*. 12 Blanchard Road, Burlington, MA.

Polaroid Corporation (1997). Ultrasonic Components Group, 119 Windsor St., Cambridge, MA. *Polaroid Manual*.

Prieve, C. & Marchette, D. (1987). An application of neural networks to a data fusion problem. *1987 Tri-Service Data Fusion Technical Proceedings, Johns Hopkins University, Baltimore, MD* (pp. 226–236).

Roitblat, H. L., Au, W. W. L., Nachtigall, P. E., Shizumura, R., & Moons, G. (1995). Sonar recognition of targets embedded in sediment. *Neural Networks*, *8* (7/8), 1263–1273.

Ruck, D. W., Rogers, S. K., Kabrisky, M., Oxley, M. E., & Suter, B. W. (1990). The multi-layer perceptron as an approximation to a Bayes optimal discriminant function. *IEEE Transactions on Neural Networks*, *1* (4), 296–298.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagation errors. *Nature*, *323*, 533–536.

Simmons, J. A., Saillant, P. A., Wotton, J. M., Haresign, T., Ferragamo, M. J., & Moss, C. F. (1995). Composition of biosonar images for target recognition by echolocating bats. *Neural Networks*, *8* (7/8), 1239–1261.

Utete, S. W., Barshan, B., & Ayrulu, B. (1999). Voting as validation in robot programming. *International Journal of Robotics Research*, *18* (4), 401–413.

Watanabe, S., & Yoneyama, M. (1992). An ultrasonic visual sensor for three-dimensional object recognition using neural networks. *IEEE Transactions on Robotics and Automation*, *8* (2), 240–249.

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

Widrow, B., & Winter, R. (1988). Neural nets for adaptive filtering and adaptive pattern recognition. *IEEE Computer*, *21* (3), 25–40.

Willson, G. B. (1990). Radar classification using a neural network. *Proceedings of SPIE, Optical Engineering and Photonics in Aerospace Sensing: Application of Neural Networks*, *1294*, 200–210.

Zemanek, J. (1971). Beam behavior within the near-field of a vibrating piston. *The Journal of the Acoustical Society of America*, *49* (1(Part 2)), 181–191.