Applied Technology Institute (ATIcourses.com)

Stay Current In Your Field • Broaden Your Knowledge • Increase Productivity

349 Berkshire Drive • Riva, Maryland 21140
888-501-2100 • 410-956-8805
Website: www.ATIcourses.com • Email: ATI@ATIcourses.com

Boost Your Skills With ATIcourses.com!

ATI Provides Training In:

- Acoustic, Noise & Sonar Engineering
- Communications and Networking
- Engineering & Data Analysis
- Information Technology
- Radar, Missiles & Combat Systems
- Remote Sensing
- Signal Processing
- Space, Satellite & Aerospace Engineering
- Systems Engineering & Professional Development

Check Our Schedule & Register Today!

The Applied Technology Institute (ATIcourses.com) specializes in training programs for technical professionals. Our courses keep you current in state-of-the-art technology that is essential to keep your company on the cutting edge in today's highly competitive marketplace. Since 1984, ATI has earned the trust of training departments nationwide, and has presented On-site training at the major Navy, Air Force and NASA centers, and for a large number of contractors. Our training increases effectiveness and productivity.

Learn From The Proven Best!

MODEL BASED CLASSIFICATION USING MULTI-PING DATA

Christopher P. Carbone

Naval Undersea Warfare Center
Division Newport
1176 Howell St.
Newport, RI 02841

Steven M. Kay

University of Rhode Island
Department of Electrical and
Computer Engineering
4 East Alumni Ave., Kingston, RI 02881

ABSTRACT

This paper proposes a method of target classification using three dimensional (3-D) data. The data consists of multiple realizations (pings) of range versus bearing plots, so the three dimensions of the data are range, bearing and time (or pings). The data is assumed to consist of independent non-identically distributed complex gaussian noise, and a target. The Target (TGT) is of known constant size (extent in range and bearing) and known speed. The TGT power, and heading are unknown. In the derivation of the classifier a normalization step is necessary and we propose an approach to the normalization of multidimensional (m-D) data. This paper contains the derivation of the classifier, a description of the normalizer, a description of the algorithm that follows from the classifier and simulation results.

1. INTRODUCTION

In the past SONAR data has been processed by breaking up the 3-D data into one (and sometimes 2-D) pieces (see [1] and [2]). This has been done to simplify processing and reduce computational load. The normalization step has been done by windowing the data in one or sometimes two dimensions as well. This has the disadvantage of not using all available information to perform the normalization. This paper will present an algorithm developed by using the generalized likelihood ratio test (GLRT) (see [3]). The resulting algorithm operates on 3-D data without windowing or breaking the data up into one or two dimensional parts. This algorithm can be used to normalize, cluster (group like threshold crossings) and classify the data. In particular, this algorithm inputs pings of data that have been beamformed and outputs a list of possible TGTs.

2. PROBLEM STATEMENT

Sonar data can be thought of as existing in three dimensions. In this case, the 3-D data is formed by stacking N range-bearing plots (see Figure 1 with $N = 3$). Each range-bearing plot is referred to as a ping's worth of data. Other dimensions could be added such as doppler but in this paper we will be considering the discrete 3-D case of range, bearing and ping.

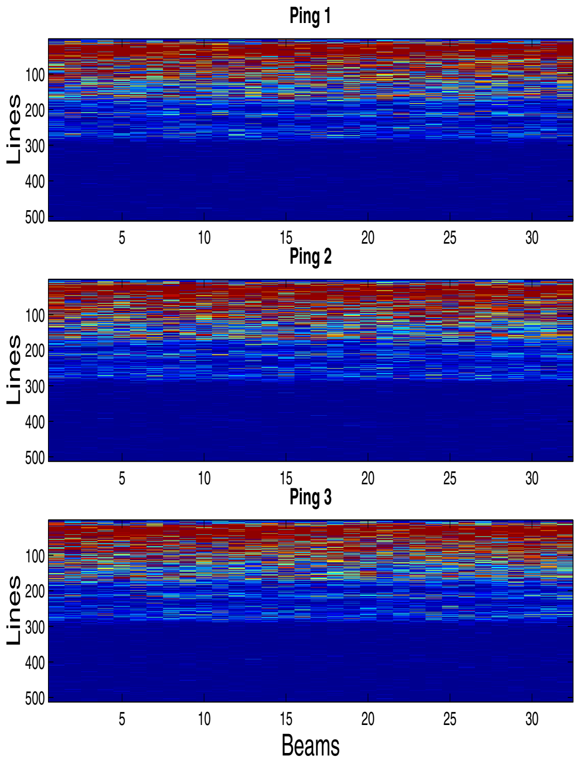

Fig. 1. Three range bearing plots

The following derivation and simulation are independent of units. Throughout the paper, range is in lines, bearing is in beams, and time is in pings. Lines could be any distance. Beams could be any number of degrees, and the time between pings is not specified.

3. DERIVATION OF THE ALGORITHM

3.1. Derivation of the Classifier

Consider multiple range-bearing plots (multiple pings of range-bearing data). We are assuming the noise is independent non-identically distributed complex gaussian noise (CN). A range-bearing-ping cell of CN is denoted by $w(m_0, n_0, t_0) = u(m_0, n_0, t_0) + jv(m_0, n_0, t_0)$ where u and v are real, independent and distributed as $u \sim N(0, \sigma_w^2/2), v \sim N(0, \sigma_w^2/2)$ and σ_w^2 is the power of the cell.

In the absence of a TGT we assume, the "signal" is noise only,

$$x(m, n, t) = w(m, n, t) \tag{1}$$

where $w(m, n, t) \sim CN(0, P_{m,n,t})$ for m-lines, $m = 0, \ldots, M-1$, for n-beams, $n = 0, \ldots, N-1$, and for, t-pings, $t = 0, \ldots, T-1$.

When a TGT is present we observe

$$x(m, n, t) = s_i(m, n, t) + w(m, n, t) \tag{2}$$

where $i = 1, 2, \ldots k$ with k being the number of TGT Models and with,

$$s_i(m, n, t) = \begin{cases} CN(0, P_{tgt}) & (m, n, t) \in A(m_0, n_0)_i \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

P_{tgt} is the TGT power, an unknown constant and

$$A(m_0, n_0)_i = \{a(t = 0)_i,\ a(t = 1)_i,\ \ldots a(t = T-1)_i\}$$

$$\begin{aligned} a(t = 0)_i &= [m_0 + 0rs_i, m_0 + re + 0rs_i] \times \\ &\quad [n_0 + 0bs_i, n_0 + be + 0bs_i] \\ a(t = 1)_i &= [m_0 + 1rs_i, m_0 + re + 1rs_i] \times \\ &\quad [n_0 + 1bs_i, n_0 + be + 1bs_i] \\ &\ \vdots \\ a(t = T-1)_i &= [m_0 + (T-1)rs_i, \\ &\quad m_0 + re + (T-1)rs_i] \times \\ &\quad [n_0 + (T-1)bs_i, \\ &\quad n_0 + be + (T-1)bs_i] \end{aligned} \tag{4}$$

where,

- m_0 is the start of the non-zero range cells
- n_0 is the start of the non-zero bearing cells
- re is the range extent
- be is the bearing extent
- rs_i is the range speed
- bs_i is the bearing speed

The TGT speed is assumed known, but the heading is not, so that rs_i and bs_i are constrained but not known.

To set up the classifier we have:

$$\begin{aligned} H_0 : x(m, n, t) &= w(m, n, t) \\ H_i : x(m, n, t) &= s_i(m, n, t) + w(m, n, t) \end{aligned} \tag{5}$$

Assumptions in the classifier:

- All range-bearing-ping cells are independent (but not identically distributed) random variables.
- P_{mnt} is known.
- P_{tgt} is known and constant
- One of the TGT models matches the true TGT heading. See section 5 on relaxing this requirement.

Under $H_i : p(x; m_0, n_0, s_i, H_i) =$

$$\begin{aligned} &\prod\prod\prod_{(m,n,t)\in A(m_0,n_0)_i} \frac{1}{\pi S_{mnt}} \exp\left(\frac{-|x(m, n, t)|^2}{S_{mnt}}\right) \\ \cdot &\prod\prod\prod_{(m,n,t)\notin A(m_0,n_0)_i} \frac{1}{\pi P_{mnt}} \exp\left(\frac{-|x(m, n, t)|^2}{P_{mnt}}\right) \end{aligned} \tag{6}$$

with $S_{mnt} = P_{tgt} + P_{mnt}$.

Under $H_0 : p(x; H_0) =$

$$\begin{aligned} &\prod\prod\prod_{(m,n,t)\in A(m_0,n_0)_i} \frac{1}{\pi P_{mnt}} \exp\left(\frac{-|x(m, n, t)|^2}{P_{mnt}}\right) \\ \cdot &\prod\prod\prod_{(m,n,t)\notin A(m_0,n_0)_i} \frac{1}{\pi P_{mnt}} \exp\left(\frac{-|x(m, n, t)|^2}{P_{mnt}}\right) \end{aligned} \tag{7}$$

As a result we have, $\frac{p(x;m_0,n_0,s_i,H_i)}{p(x;H_0)} =$

$$\frac{\prod\prod\prod_{(m,n,t)\in A(m_0,n_0)_i} \frac{1}{\pi S_{mnt}} \exp\left(\frac{-x|(m,n,t)|^2}{S_{mnt}}\right)}{\prod\prod\prod_{(m,n,t)\in A(m_0,n_0)_i} \frac{1}{\pi P_{mnt}} \exp\left(\frac{-x|(m,n,t)|^2}{P_{mnt}}\right)} \tag{8}$$

Simplifying we have $\frac{p(x;m_0,n_0,s_i,H_i)}{p(x;H_0)} =$

$$\frac{\mathcal{A} \exp\left(\sum\sum\sum_{(m,n,t)\in A(m_0,n_0)_i} \frac{-|x(m,n,t)|^2}{S_{mnt}}\right)}{\mathcal{B} \exp\left(\sum\sum\sum_{(m,n,t)\in A(m_0,n_0)_i} \frac{-|x(m,n,t)|^2}{P_{mnt}}\right)} \tag{9}$$

with $\mathcal{A} = \prod\prod\prod_{(m,n,t)\in A(m_0,n_0)_i} \frac{1}{\pi S_{mnt}}$
and $\mathcal{B} = \prod\prod\prod_{(m,n,t)\in A(m_0,n_0)_i} \frac{1}{\pi P_{mnt}}$

Taking ln of both sides we have $\ln\left(\frac{p(x;m_0,n_0,s_i,H_i)}{p(x;H_0)}\right) =$

$$\ln\left(\frac{\mathcal{A}}{\mathcal{B}}\right)+$$

$$\sum\sum\sum_{(m,n,t)\in A(m_0,n_0)_i} |x(m,n,t)|^2\left(\frac{1}{P_{mnt}}-\frac{1}{S_{mnt}}\right) \quad (10)$$

Equation 10 leads to the test statistic $T(x) =$

$$\max_{m_0,n_0,i}\sum\sum\sum_{(m,n,t)\in A(m_0,n_0)_i} |x(m,n,t)|^2\left(\frac{P_{tgt}}{P_{mnt}S_{mnt}}\right) \quad (11)$$

The combination of m_0,n_0, and i that produces the largest test statistic is the most likely TGT Model at the most likely initial position in range and bearing.

In practice P_{mnt} is not known. It is estimated and the data normalized (whitened). After, normalization P_{mnt} and S_{mnt} are constant. Therefore ,

$$\left(\frac{P_{tgt}}{P_{mnt}S_{mnt}}\right) = c > 0 \quad (12)$$

where c is a constant.

So, equation 11 simplifies to,

$$\max_{m_0,n_0,i}\sum\sum\sum_{(m,n,t)\in A(m_0,n_0)_i} |x(m,n,t)|^2 c \quad (13)$$

Ignoring the constant term in we have,

$$\max_{m_0,n_0,i}\sum\sum\sum_{(m,n,t)\in A(m_0,n_0)_i} |x(m,n,t)|^2 \quad (14)$$

which is the test statistic used in this paper. Eq. 14 could be written as,

$$\max_{m_0,n_0,i}\sum\sum\sum_{(m,n,t)} I(m_0,n_0)_i|x(m,n,t)|^2 \quad (15)$$

where,

$$I(m_0,n_0)_i = \begin{cases} 1 & (m,n,t)\in A(m_0,n_0)_i \\ 0 & \text{otherwise} \end{cases}$$

$\sum\sum\sum_{(m,n,t)} I(m_0,n_0)_i|x(m,n,t)|^2$ is a correlation between the data and the TGT models. If we were using one TGT model we would be maximizing in 2-D. The maximum range-bearing (m_0,n_0) value would be the most likely initial range and bearing of the true TGT. In this paper we are assuming k TGT models and therefore, will therefore maximize over k sets of range-bearing values.

3.2. Description of the Normalizer

The range-bearing-time data is normalized by estimating the background power with a 3-D minimum variance spectral estimator (MVSE) and then dividing the data by the estimate. For a discussion on why it was chosen see section 4. This section follows closely the description of the 2D MVSE found in [4].

The MVSE is defined as:

$$\widehat{P}_{mv}(m,n,t) = \frac{1}{\mathbf{e}^H\widehat{\mathbf{R}}_{xx}^{-1}\mathbf{e}} \quad (16)$$

where H is the hermitian transpose, $\mathbf{e} =$

$$[z_1^0 z_2^0 z_3^0 \;\; z_1^1 z_2^0 z_3^0 \ldots z_1^{M-1} z_2^0 z_3^0 \ldots z_1^0 z_2^1 z_3^0 \ldots$$
$$z_1^0 z_2^{N-1} z_3^0 \ldots z_1^{M-1} z_2^{N-1} z_3^{T-1}]$$

with $z_1 = exp(j2\pi m/M)$, $z_2 = exp(j2\pi n/N)$, $z_3 = exp(j2\pi t/T)$ and $\widehat{\mathbf{R}}_{xx}$ is an estimate of the autocorrelation matrix. $\widehat{\mathbf{R}}_{xx}$ is composed of estimates of correlation lags. In one dimension the lags are usually time so that the kth lag is estimated by $\frac{1}{N}\sum_{n=0}^{N-1} x(n)^* x(n+k)$. In 3-D the lags are in lines, beams and ping. A detailed discussion of constructing the $\widehat{\mathbf{R}}_{xx}$ matrix is beyond the scope of this paper. For more details see [4] and [5].

4. ALGORITHM

The algorithm in this paper is developed from eq. 15. It is implemented as follows:

1. Store N range-bearing plots to form 3-D data.

2. Estimate 3-D power spectral density using a 3-D estimator on above data.

3. Divide data by the spectral estimate to generate normalized data.

4. Implement the GLRT using normalized data.

5. Save the maximum initial range-bearing cell and the associated TGT model. This is the range, bearing and heading of the true TGT.

6. Center a $re \times be$ rectangle over the maximum range-bearing cell from each of the k 2-D outputs and replace all the values covered by the rectangle with zeros.

7. Repeat steps 5-6 until you have the desired number of possible TGTs.

A more detailed description is given next.

4.1. Form 3-D Data

To form 3-D data in a realtime system, save the range-bearing plots (pings of data) until the desired number of range-bearing plots have been accumulated. Then stack the range-bearing plots to form 3-D data. In a non-realtime system (or a simulation) just form the 3-D data using the desired number of range-bearing plots.

4.2. Estimate 3-D Data

The data can be thought of as the power in range, bearing, and time. Because of this, power spectral density techniques can be used. In this paper the 3-D MVSE was chosen, as noted above. The MVSE was chosen because it:

1. Has better resolution than moving average (MA). (see [4])
2. Avoids the instability problems of the autoregressive (AR) and autoregressive moving average (ARMA) estimators. (see [5])
3. Is extendable to any number of dimensions.
4. Is readily implemented in MATLAB or C.

An example of the spectral estimate is shown in Figure 3, which corresponds to the data in Figure 2. Note Figures 2 and 3 are 2-D slices of 3-D data and a 3-D spectral estimate.

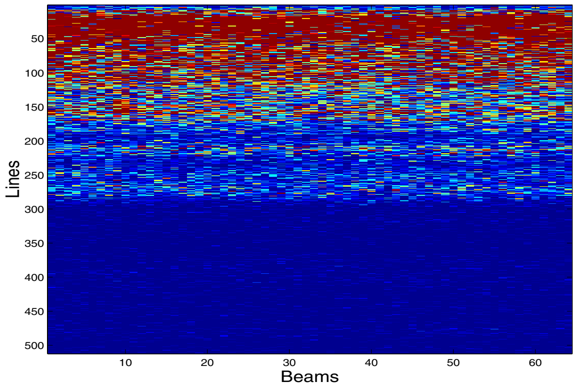

Fig. 2. One ping slice of the range-bearing-time data before normalization.

4.3. Generate Normalized Data

Point-wise divide the original 3-D Data by the estimate of the power in each range-bearing-ping cell. The estimate was found by using the MVSE on the original data containing *both target and noise*.

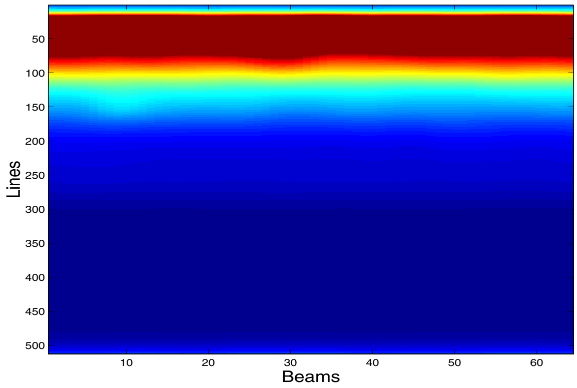

Fig. 3. Slice of the power spectral density estimate using the MVSE.

4.4. Correlate Smoothed Data

The GLRT is now implemented see (15). This entails correlating the normalized data with the TGT models. If the TGT size (in range lines and beams), and the speed and heading are all known, we would need only one TGT Model. In MATLAB the TGT Model is a 3-D array, with ones in the space that the TGT would occupy and zeros elsewhere. For example, if the data set had four pings, Figure 4 is the TGT model for a TGT that is twenty lines long, two beam wide, and moving up twenty lines per ping. In this paper heading is not assumed known. Therefore, we will need one TGT model for each heading we wish to detect a TGT moving in. In practice, over small numbers of pings or for a slowly moving TGT, the number of search headings would be small.

4.5. Max Output of the Correlation

Find the maximum range-bearing cell over all the TGT models. This is the most likely TGT. This should have come from the TGT model that is the closest match to the true TGT in location and movement heading. Save this information as it is a possible TGT.

4.6. Zero Out Possible TGT

If the TGT model (and the true TGT) were points (existing in only one range-bearing cell) then correlating the data and the TGT model would produce a largest test statistic for the true TGT with low outputs elsewhere. Since we are not assuming a point TGT there will be large values close in range and bearing to the true TGT location. So, if we are looking for the location of the N most likely TGTs we cannot just find the N largest correlation outputs from each TGT model, because one true TGT or one area of high reverberation will produce many possible TGTs all close in range and bearing. To eliminate this possibility, center a

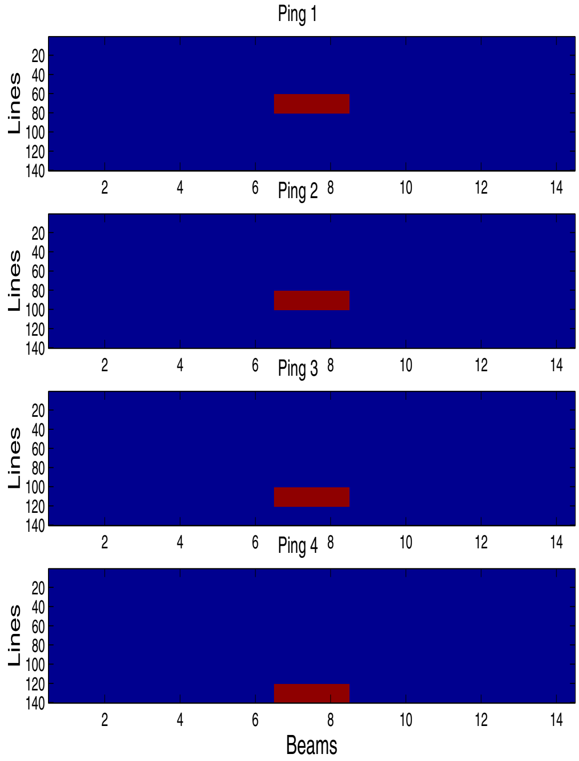

Fig. 4. Example of TGT model (red area is unity and blue area is zero)

$re \times be$ rectangle over the possible TGT range and bearing in each TGT model and replace the values in the rectangle with zero.

4.7. Repeat Search for Possible TGTs

Eq. 15 produces the most likely true TGT range, bearing and heading (TGT model). However, in practical systems the possibility of missing the true TGT necessitates accepting a few false alarms. Therefore, the N largest test statistics are found and treated as possible TGTs. The possible TGTs could be passed to an automatic tracker or displayed on a screen.

5. COMPUTER SIMULATION

5.1. Noise Data

The data for each trial is made by generating a 3-D array of independent non-identically distributed squared CN random variables. The size of the array is 512 range lines, 64 beams and 4 pings. The noise is non-identically distributed because the power (variance) of the noise changes with each cell. The power in range starts low, quickly grows, then slowly decays (similar to what one might see in a reverberation-limited environment). The beams are combined to make range-bearing plots and the range-bearing plots are combined to make range-bearing-ping plots (see Figure 1).

5.2. Target Data

The TGT is then placed into the data. The TGT consists of the square of CN random variables with power equal to twice the average power in the area where the TGT is to be injected. The TGT is twenty range lines long and two beams wide. In the first 100 trials the simulated TGT is moving up in beam number at the rate of two beams per ping. In the second 100 trials the simulated TGT is moving up in line number at the rate of twenty lines per ping.

5.3. Algorithm

The normalization step is accomplished with a MVSE as noted above. The MVSE used in the simulation makes use of a one ping, four lines, and four beams lag estimate.

The eight TGT models that are used in this simulation are matched in speed to the injected TGT. They cover eight combinations of moving left/right in beam and up/down in line (see Figure 5).

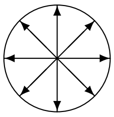

Fig. 5. The eight TGT model vectors

Fig 4 is an example of one of the TGT models. This algorithm allows for the addition of more unknowns. For instance if speed were unknown, a different TGT set matched to each speed could be be used. Also non-constant velocity TGT models could be used and even different TGT models for different size TGTs. But in this simulation only the eight TGT models mentioned above are used.

5.4. Simulation Details

The performance of the algorithm is tested by repeatedly injecting a simulated TGT into the range-bearing-ping data at a position when the data is reverberation dominated. The reverberation dominated portions are in the early lines (20 to 60) of the data (see Figure 6) The data is processed with the above algorithm. The algorithm then produces a list of 100 possible TGTs sorted by likelihood. The 20-line by 2-beam rectangle (the size of the true TGT) is placed over the possible TGTs, starting with the highest likelihood. If the true TGT is within the rectangle the true TGT is considered found. The possible TGT location and model are recorded. If none of the 100 possible TGTs find the true TGT the algorithm is considered as having missed the true TGT for that realization. The results are summarized in tabular form in Figures 7 and 8.

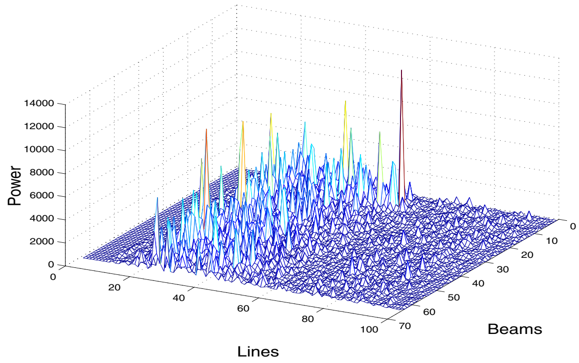

Fig. 6. Example of reverberation dominated portion of one ping of data

One hundred range-bearing-ping data sets were simulated. The data sets were comprised of four pings worth of range-bearing plots, each consisting of 512 lines and 64 beams.

5.5. Simulation Results

The results of 100 trials where the TGT is moving up in beams is shown in Figure 7.

true TGT found in the	Percent
first	82
2-25	13
26-50	5
51-75	0
76-100	0
Not Found	0

Fig. 7. Performance of the algorithm

The results of a second 100 trials where the TGT is moving up in lines is shown in Figure 8.

true TGT found in the	Percent
first	49
2-25	38
26-50	13
51-75	0
76-100	0
Not Found	0

Fig. 8. Performance of the algorithm

In the above simulations, the performance of the algorithm for a TGT moving up in beams is better than the performance for a TGT moving up in lines. This could be caused by the much greater changes in noise power down the lines versus the lesser changes in noise power across the beams.

6. CONCLUSIONS

In this paper we have developed an algorithm that inputs multiple pings of range-bearing data and outputs a list of possible TGTs. In the simulation the estimated TGT that matches the true TGT is usually one of the highest ranking possible TGTs. Often the highest ranking possible TGT matches the true TGT. This means the true TGT can be found with very few false alarms. A complete implementation in MATLAB is available upon request.

7. REFERENCES

[1] R. J. Urick, *Principles of Underwater Sound*, New York, NY; McGraw-Hill Book Company, 1983

[2] R. O. Nielsen, *Sonar Signal Processing*, Boston, MA; Artech House, 1991

[3] S. M. KAY, *Fundamentals of Statistical Signal Processing: Detection Theory*, Upper Saddle River, NJ; Prentice-Hall, 1998

[4] S. M. KAY, *Modern Spectral Estimation*, Englewood Cliffs, NJ; Prentice-Hall, 1988

[5] D. E. DUDGEON, AND R. M. MERSEREAU, *Multidimensional Digital Signal Processing*, Englewood Cliffs, NJ; Prentice-Hall, 1988