ENVIRONMENTALLY ADAPTIVE SONAR CONTROL IN A TACTICAL SETTING

WARREN L. J. FOX, MEGAN U. HAZEN, AND CHRIS J. EGGEN
University of Washington, Applied Physics Laboratory, 1013 NE 40th St., Seattle, WA 98105, USA
E-mail: warren@apl.washington.edu

ROBERT J. MARKS II AND MOHAMED A. EL-SHARKAWI
University of Washington, Dept. of Electrical Engineering, Box 352500, Seattle, WA 98105, USA

Automatic environmentally adaptive sonar control in littoral regions characterized by high spatial/temporal acoustic variability is an important operational need. An acoustic model-based sonar controller requires an accurate model of how the sonar would perform in the current environment while in any of its possible configurations. Since high-fidelity acoustic models are computationally intensive, and finding the optimal sonar mode may require a large number of these model runs, such a controller may not be able to provide optimal line-up solutions in tactically useful time frames. We have explored a method of statistically characterizing a given operations area, generating a large ensemble of acoustic model runs, and training specialized artificial neural networks to emulate acoustic model input/output relationships. The neural networks reproduce the acoustic model outputs to a good degree of accuracy in a small fraction of the compute time needed for one of the original model runs. In this paper, the neural network training method is described, examples of neural network performance are given, and an example of controller solutions in a variable environment are presented. *(Approved for Public Release; Distribution is Unlimited.)*

1 Introduction

Naval sonar systems continue to evolve and become more capable, while at the same time becoming more complex to operate. With the emergence of littoral areas as the prime regions of interest, characterized by underwater acoustic environments that change quickly in both the temporal and spatial domains, automatically optimizing sonar line-ups has become a key operational need. The desire is for sonar operators to concentrate on the key tasks of target detection and classification, while the sonar system automatically determines an optimal line-up based on the current goals of the operator and an estimate of the current environment. Also, as autonomous systems are developed for operational use, the need for automation of environmentally adaptive sonar control becomes paramount.

Sonar control schemes generally fall into two categories: rule-based, and acoustic model-based. Rule-based systems are developed by acoustic and sonar system experts, who first define generic sets of environmental conditions, and then apply acoustic modeling techniques and the sonar equation to determine the best line-up for the sonar in those conditions. In practice, the environment in which the sonar is deployed must be assessed as to which of the generic design environments is closest to the real environment, so that the proper sonar line-up can be set. Although relatively simple and with a low real-time computational burden, rule-based controllers may not be able to take into account all of the environmental variability that may confront the sonar system and its operators.

Model-based controllers embed an acoustic model in the real-time controller. Typically, the best available estimate of the current environment is fed into the controller which makes acoustic performance predictions for the various possible line-ups of the sonar system. The line-up that best satisfies some performance metric (which may change based on the current employment of the sonar) is chosen. Although a model-based controller is more readily able to adapt to finer scale environmental conditions than a rule-based controller, high-fidelity acoustic models are computationally intensive. Assessing the performance of the various modes of the sonar may take too much time to be useful in an environment with high temporal/spatial variability, or may require excessive computing resources.

In this paper, we present a method of training artificial neural networks to emulate the input/output relations of a computationally intensive acoustic model for use in a sonar controller, either shipboard or aboard an autonomous vehicle. The advantage of using the neural networks is that they generate these acoustic model emulations orders of magnitude faster than it would take the original high-fidelity acoustic model to run, using modest computing resources. We describe the basic neural network training methodology, along with some special techniques developed specifically for this application. We also show some examples of the training performance. Finally, we give an example of control solutions obtained using a neural network.

2 Neural Network Training

2.1 Basic Idea

Neural networks are mathematical constructs loosely modeled on biological neural interconnections [1]. Figure 1 shows a schematic of how the neural network training is performed in this application. In the figure, a multilayer perceptron neural network is established with an input layer, one hidden layer, and an output layer. The number of hidden layers and number of nodes in each layer are design parameters, and must be considered carefully for each application. In Fig. 1, only some of the connections between layers are shown for illustration purposes. In reality, each node in any given layer is connected to every other node in the preceding and following layers.

The input layer contains parameters describing the sonar (e.g., center frequency, bandwidth, vertical steering angle, etc.) and parameters describing the environment (e.g., wind speed, bottom type, sound speed, etc.). These values on the input layer of the neural network are also used as inputs to an acoustic model. In the case of Fig. 1, the model computed signal-to-interference ratio for hypothetical locations of a target in a vertical slice of the ocean. The outputs of the model are assigned to nodes of the output layer of the neural network. The network is then trained using error back-propagation [1], so that when these inputs are presented to the neural network, a forward computation through the neural network reproduces, to some level of fidelity, the outputs of the acoustic model. Neural networks used in this fashion are sometimes referred to as ”regression machines,” or as ”associative memories.”

For neural networks with three hidden layers and roughly 50 nodes per layer, one of these forward computations takes about 5 milliseconds on a standard (circa 2002) desktop

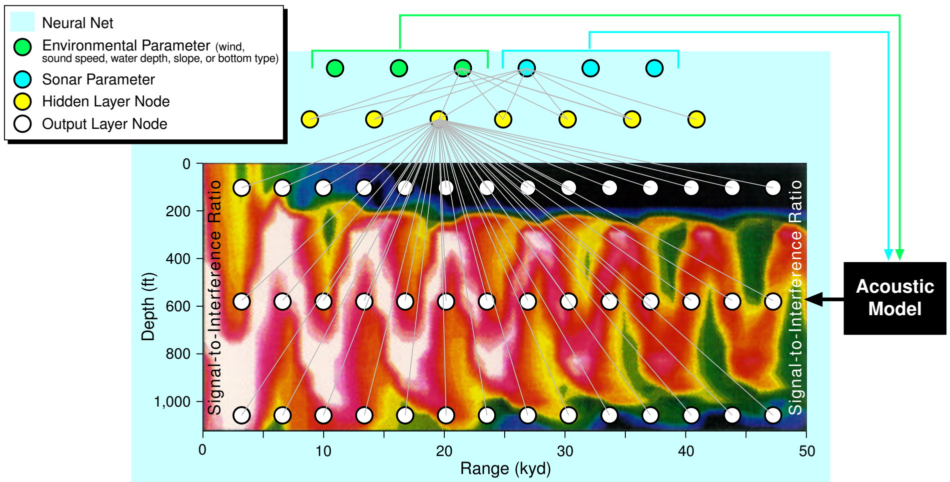

Figure 1. Neural network training on acoustic model output.

workstation. High-fidelity acoustic models may take on the order of 60 seconds for the original computation. In practice, the training data generation and neural network training would take place in laboratories with high-performance computing facilities, and the neural network subroutines would be installed on the sonar platform for use in the real-time sonar controller.

2.2 Training Data

The neural networks are actually trained on an ensemble of acoustic model runs. In order to generate this ensemble, we define a geographic region in which the sonar system will be operating, and statistically characterize both the environmental parameters in this region and the possible settings of the sonar parameters. For example, we may allow wind speed to be a uniform random variable between 0 and 12 m/s, while volume scattering strength is Gaussian with mean -75 dB/m^3 and variance of 5 dB/m^3.

For training the neural network, a standard set of depths at which sound speed in the water column will be specified must be defined. Care must be taken with variables where correlation is important, such as sound speed profile. Simply allowing the various points to vary independently as a function of depth might produce unrealistic sound speed profiles. A technique has been developed [2] for generating realistic sound speed profiles by collecting sets of historical sound speed profiles for an area, computing the covariance matrix, and multiplying the Cholesky factors of the covariance matrix by a set of independent random numbers. The first and second order statistics of the data are thereby maintained in the randomly generated sound speed profile data.

In our development of this technology, we have limited the geographic region being characterized to boxes $O \sim 1°$ square, and generated sets of acoustic models with 20,000-40,000 members. As the geographic area grows and encompasses more environmental variability, the training set would need to grow and the structure of the neural network may also need to be modified.

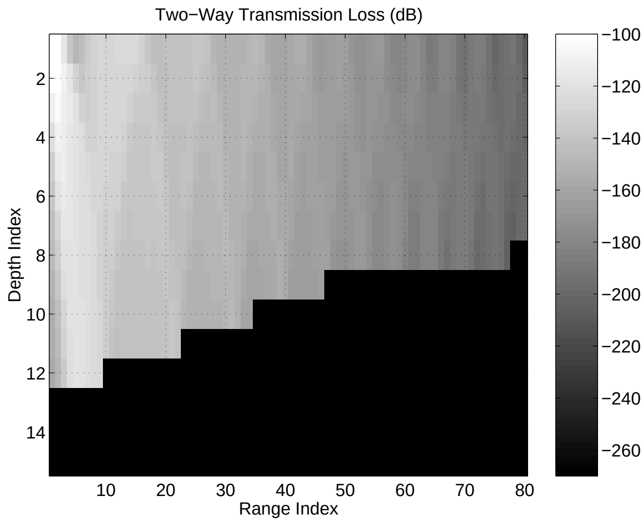

Figure 2. Example of two-way transmission loss used for neural network training.

2.3 Training for variable bathymetry

Figure 2 shows a typical transmission loss plot (two-way) used in neural network training. The dark portion at the bottom of the figure corresponds to the sea bottom, in which we assume the acoustic field levels are very low and are not part of our modelling process. We have made a bilinear approximation to the bottom bathymetry for our current development, meaning that the input layer to the neural network has four nodes corresponding to a description of bathymetry (depth at source, range of breakpoint, depth of breakpoint, and depth at final range). This approximation could be expanded at the cost of more nodes on the input layer, although the bilinear approximation has shown to be fairly robust in our testing [2].

An important issue with the neural network training has to do with the bathymetry. As with the other parameters, the bathymetry is generated randomly for the different members of the ensemble input sets, meaning that certain range-depth pixels would be in the water column for some cases and in the sea bottom for others. Early training efforts simply considered all output pixels in a uniform manner, but the transition from water column to below the seabed was difficult for the networks to learn, resulting in unacceptably high errors. A novel technique for avoiding this problem was developed [3,4], dubbed ”don’t care training.” Here, during error backpropagation training, weights connected to output nodes associated with pixels in the sea bottom are not updated. See references [3,4] for a more complete description of the technique.

3 Neural network results

3.1 Training data

Figure 3 shows some examples from a neural network trained on modelled two-way transmission loss for an active sonar. The first column contains four samples from a training

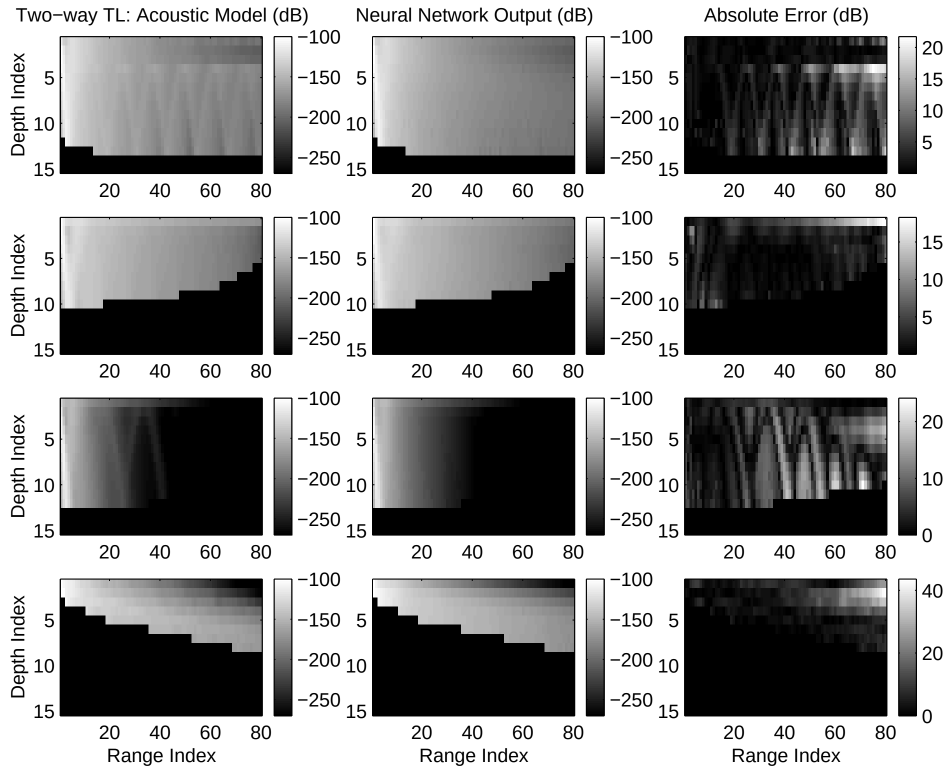

Figure 3. Examples of neural network performance on two-way transmission loss: training data.

set of 40,000 model runs, generated as described in the previous section. The second column contains the corresponding neural network outputs when the model parameters used to generate the results in the first column are placed on the input nodes of the neural network. The third column containes the absolute values of the difference between the first and second column.

The example in the first row is for a case of a downward refracting sound speed profile and a low-loss bottom, resulting in the arching pattern seen in the first plot. The neural network reproduces the pattern early in range, but tends to smear the arch energy in range and depth at longer ranges. The error plot shows the residual of the arching pattern at the longer ranges. One reason for this behavior is the sensivity of the acoustic model to small changes in input values. For example, a small change in sound speed profile can change the exact location in the range/depth plane of the arching patterns, and this sensitivity is difficult to train for. For use in a controller, it may be enough to know that there are significant amounts of energy propagating out to a particular range withough knowing the detailed structure of the acoustic field. Even if the actual acoustic model were embedded in the controller, imprecise knowledge of model input parameters would lead to imprecise location of these types of structures.

The second row is a case of a shallow surface duct and a low-loss bottom resulting in significant sub-duct propagation. The surface duct is fairly well reproduced by the neural

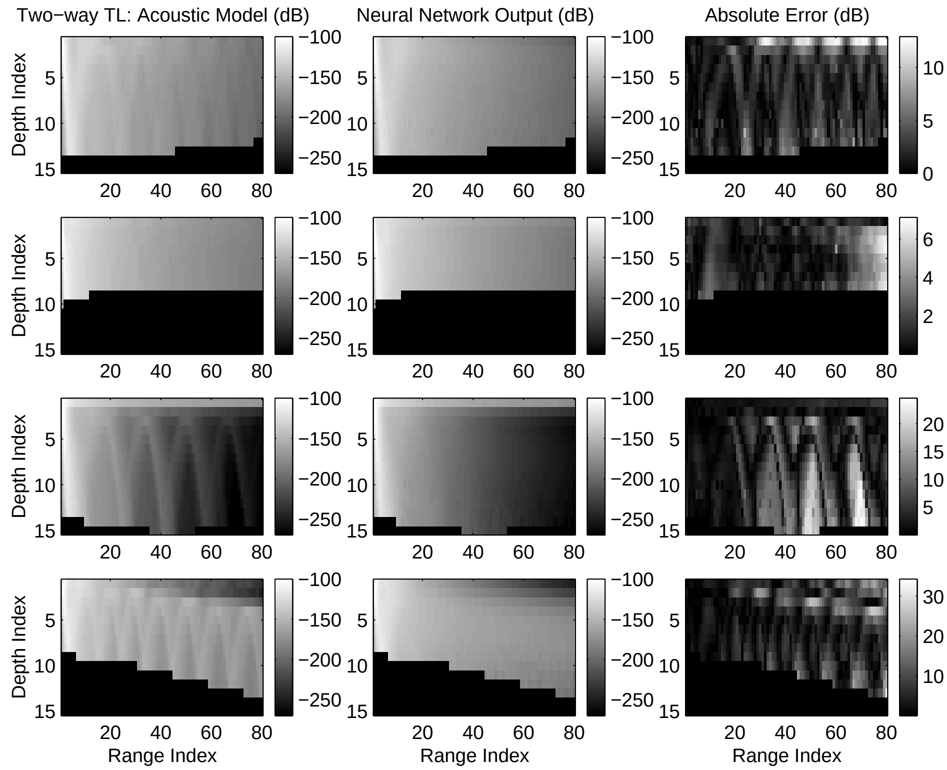

Figure 4. Examples of neural network performance on two-way transmission loss: testing data.

network, with increasing errors at long range. Again, for use in a controller, the most important piece of information may be the existence of the duct, not the exact level of the acoustic field in the duct.

The third row is a case of a downward refracting sound speed profile and a high loss bottom. Note that the neural network correctly reproduced the differences in the general trends between the low loss bottom in the first example and the high loss bottom here. Note that an arching pattern also exists here, although it is only evidenced by the residual seen in the error plot. Since it is occurring in an area of very high transmission loss, it would probably not be an issue for a controller.

The fourth row shows an example of down-slope propagation. The neural network reproduces fairly well the general trends of the propagation, with the exception of some high errors at the furthest ranges.

3.2 Testing data

As with any neural network application, assessment of the performance must be made using testing data, i.e., data that was not used during the training process. Figure 4 is formatted similar to Fig. 3, but has examples of testing data. The input parameters were generated similarly to the training data.

Note that similar characteristics are evident in the training examples: general acoustic

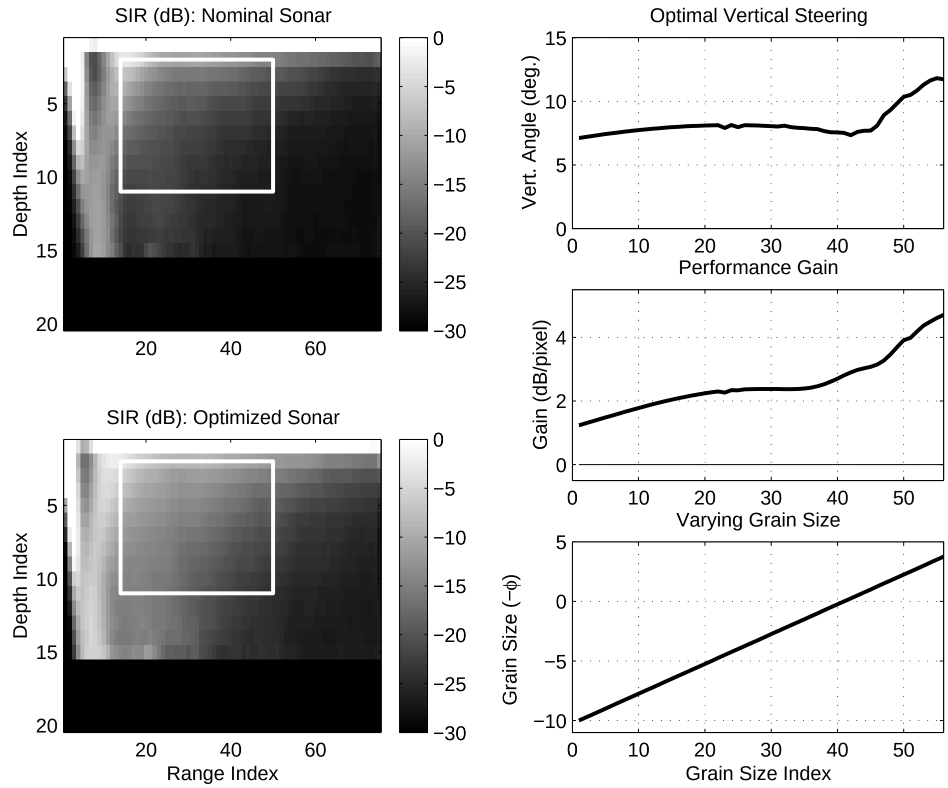

Figure 5. Examples of controller results.

trends are represented fairly well, and some details (i.e., arching patterns) are smeared (or averaged) in range and depth. This gives us confidence that the neural network has not over-trained on the data (referred to as ”memorizing”).

4 Controller results

Figure 5 shows the results of using a neural network trained using the above-described method in a sonar controller. In this case, the data used in the training was signal-to-interference ratio (SIR) for an active sonar, assuming a hypothetical target with a fixed low target strength at all possible locations in the water column.

The environment was held constant except for the bottom type, which was varied from very soft (high loss) to very hard (low loss). The bottom is characterized by the parameter ϕ, where $\phi = -log_2(d)$, where d is the grain size in mm. The parameter $-\phi$ (note the sign change) was used as an input to the acoustic model used for neural network training, and subsequently to the neural network itself on one of the nodes in the input layer. The variation of grain size is shown in the bottom right plot of Fig. 5.

The left two plots of Fig. 5 show range-depth maps of SIR, with a target search area outlined in a white box. This is the region over which the controller was required to obtain the maximum average SIR. The controller was allowed to control the vertical steering angle of the sonar. The top plot shows the SIR obtained for the final $-\phi = 4$ with a nominal sonar steering angle of 3 degrees, and the bottom plot shows the SIR obtained using the controller-optimized steering angle of approximately 12 degrees. The optimal steering angles for all bottom types is plotted in the upper right, and the average gain per pixel obtained by using the optimal steering angle vice the nominal is plotted in the middle axis on the right.

In this example, it is shown that even for softer bottoms, steeper vertical steering angles around 7 degrees are more advantageous, with a sharp increase for harder bottoms, accompanied by higher gains in performance. The key point here, however, is not the specific results for this example, but that the results can be generated in a very short period of time compared to what would be necessary if the actual acoustic model were used. This allows rapid investigation of how sonar systems should be employed in particular environments.

5 Discussion

It is important to keep in mind that the technique described in this paper for emulating acoustic model input/output relations is intended for use in a sonar controller. In other words, the emulation must be ”good enough” to put the sonar in the correct configuration. The emulation is not intended to be used for detailed acoustic analysis. As mentioned before, the main benefit of this technique is the reduced computational complexity it brings for real-time applications. The primary thrust of our continuing work is assessing trained neural networks in actual controller scenarios, and comparing the controller solution performance against controllers with embedded acoustic models.

Also of note is that the training and testing examples presented in Section 3 are the result of fairly straightforward data set generation and neural network training with the ”don’t care” technique for range-depth pixels below the sea bottom. We have also developed several techniques whose details are beyond the scope of this paper for improving the performance of the neural networks. One of these involves detailed examination of the input/output sensitivities of the underlying acoustic model (from the generated data set), and insertion of additional data (model runs) where the sensitivities are high. These techniques are also important aspects of our continuing work.

Acknowledgements

This work was performed under sponsorship from the Office of Naval Research, contract number N00014-98-G-0001.

References

1. Reed R. D. and Marks II R. J., *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks.* Cambridge, MA: MIT Press, 1999.
2. Eggen C. J., unpublished APL/UW memorandum, Oct. 2001.
3. Jung J.-B., El-Sharkawi M. A., Marks II R. J., Miyamoto R., Fox W. L. J., Anderson G. M., and Eggen C. J., ”Neural network training for varying output node dimension,” in *Proc. Intl. Joint Conf. on Neural Networks,* vol. 3, pp. 1733-8, 2001.
4. Jung J.-B., ”Neural network ensonification emulation: training and application,” Ph.D. dissertation, University of Washington, Seattle, Washington, 2001.