

AUV Robot's Real-time Control Navigation System Using Multi-layer Neural Networks Management

Amir M. Anvar^a and Amir P. Anvar^a

*^a School of Mechanical Engineering, The University of Adelaide
SA 5005 AUSTRALIA*

Email: amir.anvar@adelaide.edu.au

Abstract: This paper describes the detection and tracking of static and dynamic underwater object(s). It addresses the case study application of a multi-layer artificial neural network prototype model on the bases of an analytical approach. It supports an Autonomous Underwater Vehicle (AUV) robot's controller system with automated detection of processed-obstacle-signals. The significance of this work is to investigate the neural network learning perception process of signal detection within operational environments. In this case, the acoustic-sound density is the source of detection and classification processes. The outcomes of this work are presented as simulated results that illustrate the error-detection control system. It activates due to a range of training forces originating from encountered acoustic-sensors' signals. In addition, the benefit of further simulation of the proposed technique can provide sufficient knowledge on the set-up of the controller's cyclic triggering towards actuators. The other benefits are included with control overshoot and rotational alignment of thrusters for precise navigational trajectory in real-time.

Keywords: *Multi-layer Artificial Neural Networks Model, Simulation trial, Analytical Approach, Autonomous Underwater Vehicle, AUV, Robot, Controller system, Automated Detection, Classification, Time domain, Real-time, Acoustic Signals*

1. INTRODUCTION

This paper proposes an investigation into the development (model, design and operation) of an artificial neural network to perform as an AUV robot's controller automated detection system application using treated real-time acoustic signatures. In general, the parameter-estimation within a cyclic-system-identification process is a principle factor for control-system design, in particular for sensitive systems with real-time adaptive or non-adaptive control-system duties.

As Madkour *et al.* (2004) revealed, the accurate and consistent parameter-estimation technique is essential for the design and development of high performance control systems in which the estimated parameters are often used in the field orientation, motion control, self-sensing and other advanced algorithms. In many cases of AUV robot's real-life applications, it is not possible to update parameters within a time-domain slot between successive samples. Therefore it is essential to apply a clear-cut algorithm for real-time execution.

A case study has been considered to demonstrate the capabilities of algorithms to simulate detection and classification (Anvar 1999) with real-time system identification. In this paper the authors propose a flexible prototype model implemented using real-time signatures.

The model has up to ten inputs (input layer), a hidden-layer (with optional choice of 2 or 4 nodes setting (i.e. explicitly, there are two hidden nodes decided and tested for the present scenario) and output layer (with alternative of one to three outputs depending on the application type) in real-time. Where needed, the number of inputs and outputs can be changed or extended in the future.

Furthermore, the system has been tested and validated for real-time state-identification and classification within the simulation framework. To demonstrate the capabilities of the algorithms and the system applications several real-time simulation scenarios have also been tested and the results are presented. In addition, the model provides an explicit control on the trade-off to achieve the true-model.

2. TARGET TRACKING AND DETECTION SYSTEM

The qualitative automated detection control system should have the ability to sense its environment, process information, reduce uncertainty, plan, generate and execute (e.g. detect, classify and inform AUV robot's controller) in real-time (see Figures 1 & 2). Alternatively, the robust-system should be capable of handling various tasks within its environment. Depending on its applications, the other features would be an online and/or off-line learning ability. For the real-world application when AUV is engaged with real-time tasks, the system should be pre-trained and non-adaptive. To be on the safe side, a real-time on-line training scheme in some cases may cause dramatic effects so as to exceed robustness limits.

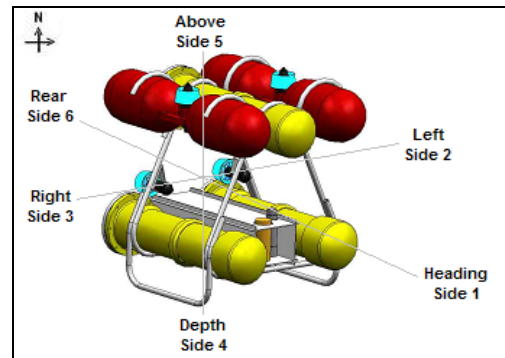


Figure 1: AUV Robot test-bed and Expected Directional Sensors Response.

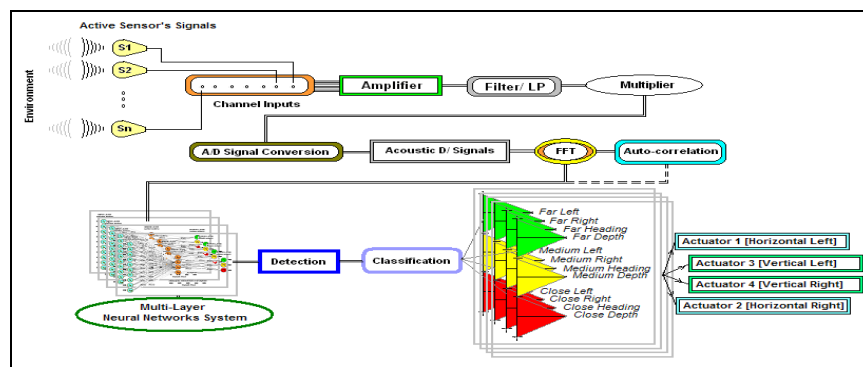


Figure 2: An AUV Robot's Automated Detection and Control Navigation System Processing Diagram

3. MULTI-LAYER PERCEPTRON ARCHITECTURE

The artificial neural networks are well known in learning separable-functions, which can often be identified as step-function-units, particularly when the multi-layer network is fairly large. They can be engaged in the learning process of multi-tasks. However, the learning algorithm can be derived analytically, using differential calculus. In this arrangement a differentiable threshold function plays a major-role and in most cases it can perform effectively and accurately.

In general the neural network is integrated with several inputs joined to a summing function by various coefficient-weights (see Figure 3). The outcomes of this operation is presented as the following expression, where the input vector is x_i ($i=1:n$) and w_i symbolizes the coefficient-weight that connects i^{th} input to the corresponding neuron in the next hidden layer.

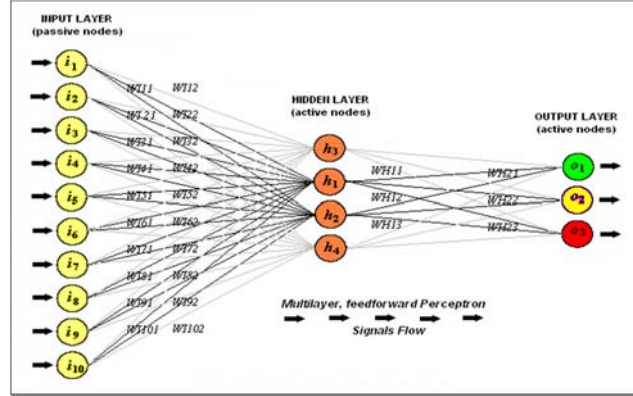


Figure 3: The Neural Network System Model (Prototype) Architecture.

$$y = \sum(x_i * w_i) \quad (1)$$

The signals are then transmitted through a non-linear-function or sigmoid-operator after the weighted input is summed, to form the output at the hidden layer.

The activation function (sigmoid-units) has been well used as an alternative unit with similar property to the step function. The general form of the sigmoid function is presented in the following relation, where x is the input real-value,

$$\rho_x = \frac{1}{1 + e^{-x}} \quad (2)$$

Where ρ is in the range from 0 to 1. The term e^{-x} cannot become negative, so when the function ρ_x denominator becomes a large value (infinite), it increases in negative direction (turn towards 0). It also inclined to 1 towards positive direction when ρ_x denominator becomes 1.

In reality the output value would be (+1) if the input is positive-real ($x > 0$). Likewise the output value would be (-1) if the input is negative-real ($x < 0$).

There is a similarity between the step-function and sigmoid-function. The following relation represents the sigmoid- derivative:

$$\rho'_x = \rho_x[1 - \rho_x] \quad (3)$$

It is obvious that the power of a multi-layer neural net approach is generated as a result of the process within the neurons that interact together and by using a training method that can define the paradigm of a multi-layer neural networks perception. Alternatively, the network learns the problem “space” as well as cross correlation between data connections due to parallel processing that minimises defined error signals.

4. Self-Organising Threshold

An onboard-automated tracking of the object's location is an essential role of the command and control system of intelligent self-motivated robots. These problems usually involve comparing the acquired data-density against that of the threshold. If the target or the object-situation exceeds a given threshold, it means the object is assumed to be within a close range of navigational path. The following relation acknowledges the defined *activation-function*.

$$S_y = \sum_{i=0}^n x_i * w_{ij} \quad (4)$$

Subsequently, the following levels of discrete-senses (threshold categorization) are the basis of the region of the state-space limitations to be used by a dynamic-system:

$$\begin{aligned} S_y &> \theta \\ -\theta &< S_y \leq \theta \\ S_y &\leq \theta \end{aligned}$$

Where θ is a predetermined threshold value (Estebon 1997) & (Howell, et al 2003).

5. Training the Network using Backpropagation

The coefficient-weights-training can be considered by the estimation of the weight changes for a particular scenario-trial. In this case, the a_k is measured as an aim-value, o_k that is known as an output unit and h_k is identified as the trial-value. The following consecutive terms (5) and (6) would estimate the output-node and hidden-node error-rate associated with each unit.

$$\Delta O_k = o_k(1 - o_k)(a_k - o_k) \quad (5)$$

$$\Delta H_k = h_k(1 - h_k) \sum_{i \in \text{outputs}} w_{ki} * \Delta O_i \quad (6)$$

The outcomes of the above calculation would act as a source of coefficient-weight-changes within the iteration cycle by employing the following relations.

$$\delta_{xH} = \mu \Delta H_j * x_i \quad (7)$$

Where, δ_{xH} is the weight change between *input-unit* x_i and *hidden-unit* H_j and μ is the learning rate.

Therefore, with each iteration-cyclic of δ_{xH} the coefficient-weight value would be modified (Rumelhart, et al 1995). Eventually, the learning rates used for each scenario-trial would be changed only with a short-gape of coefficient-weights of each scenario-trial. It is also noticeable that with each learning process the training for the previous trial is also included and available. In addition, the relation (7) can be modified and applied for weights between *hidden-unit* h_i and *output-unit* O_j as follows:

$$\delta_{hO} = \mu \Delta O_j * h_i \quad (8)$$

6. Error Evaluation

The number of misclassified instances of the multi-layer neural networks can be measured by the estimation of the error-rate over a training-set. In most cases due to the network systematic error, there are various dramatic instances that may take place. One assumption would be a case of a misfired situation, when the output-node value assigns a value close to 1 where the correct output value should be 0 and so on.

To evaluate the error rate (E_R), when A_O is defined as an aim-output and O_O is determined as an observed-output the following function computes the overall error function (Madkour, et al 2004) & (Shin, et al 2002).

$$E_R = \frac{1}{2} \sum (\sum_{k \in \text{outputs}} (A_O(k) - O_O(k))^2) \quad (9)$$

However, the backward searching configuration process of the network space is a sensible way to configure the coefficient-weight with least error.

7. Fitting and Errors Generalisation

The choices of best fit are governed by various applications of specific features (Yerramalla 2003). Alternatively, the process to examine and select a reliable model is involved with fitting among training-set-error and test-set-error. As presented in Figure 4, the fitting of the two sets capturing the underlying

relationship of the true model is evident when the number of training iteration reaches 1200 epochs. This phenomenon is possible due to the output relationship of the true model being deterministic and smooth from 14000 up to 20000 epochs (see Figure 5). However, this figure seems to be a good representative of the true model, although, different error rates appear among two sets. Figures 6 & 7 represents the weight changes before and after training.

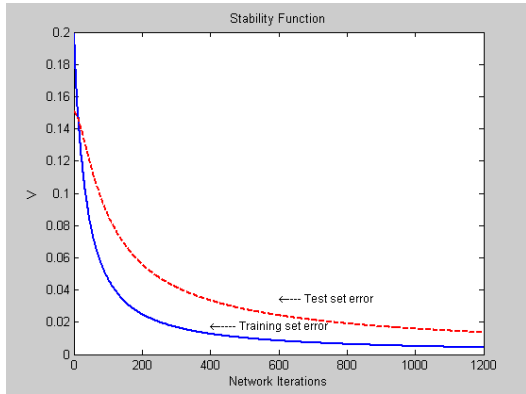


Figure 4: Convergence of V to a stable State within 1200 epochs of NN Learning.

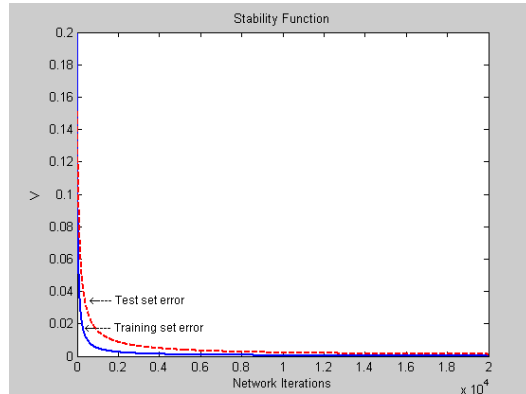


Figure 5: Convergence of V to a stable State within 20,000 epochs of NN Learning.



Figure 6: Input Coefficient-Weights Training Weights.

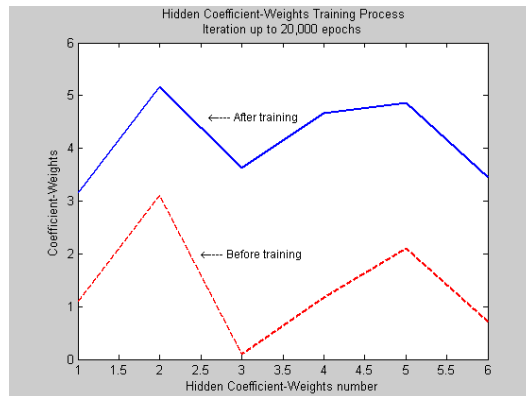


Figure 7: Hidden Coefficient-Weights Training.

Where V on the above figures represents error rate (E_R).

8. CASE STUDY

A practical case study has been undertaken for better understanding of the sensitivity and capability of various functions that were used within the given system. Three sets of scenario-trials are presented in Figure 8 and depict nominal conditions of autonomous detection processes of real-time acoustic signals.

Figure 8, represents AUV robot's real-time environmental situation. The detection-range is classified into three regions, (region III is the neutral or protected area, region II is the alert zone and region I is the close range situation or alarm zone). Where A is the position of the AUV robot, B and C are objects at rest and D is a moving object.

Three possibilities are considered: possibility 1 is when the, moving object D is directed towards AUV robot

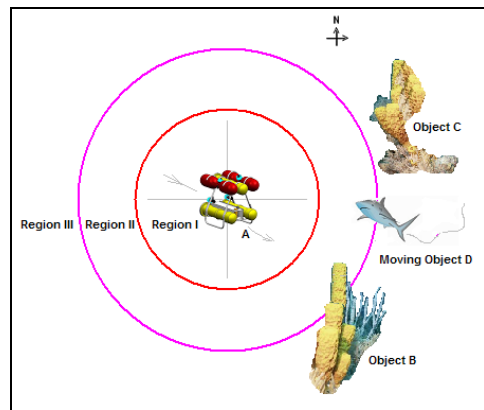


Figure 8: AUV Robot's Real-time Navigational Situation. Simulating three Possibilities.

with medium or long-range distance to each other (no real threat to the robot). Possibility 2 is when the AUV navigates too close to the object *B* that is stationed on the seabed. Possibility 3 is when the moving object *D* and AUV are moving towards each other and are within close proximity of each other (real threat to the AUV).

Figures 9, 10 & 11 show results of the network detection and classification process for several possible routes when the moving-object is directed towards the AUV robot within three regions. The process is simulated in a real-time cycle with network input considered to be of 250 digits within 25 segments. Each input data frame consisted of 10 consecutive sensor readings controlled via the neural networks system using parameter identification and derivative error correction capabilities.

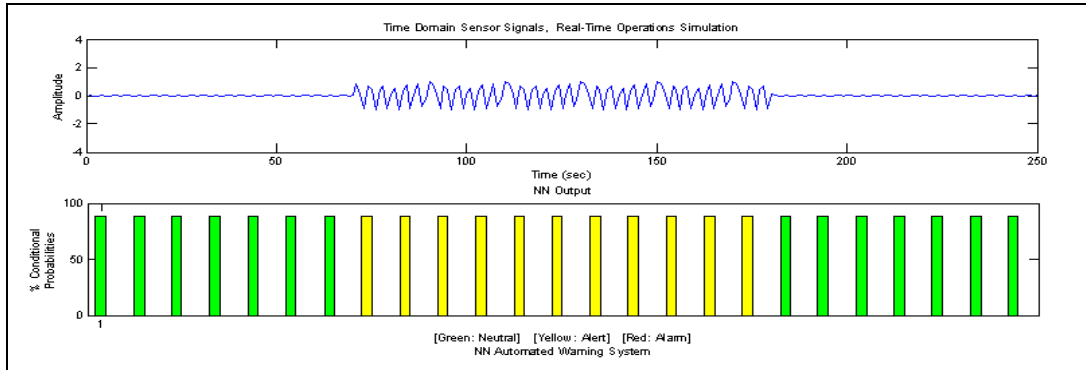


Figure 9: Real-time Autonomous Detection Simulation trial, Possibility 1.

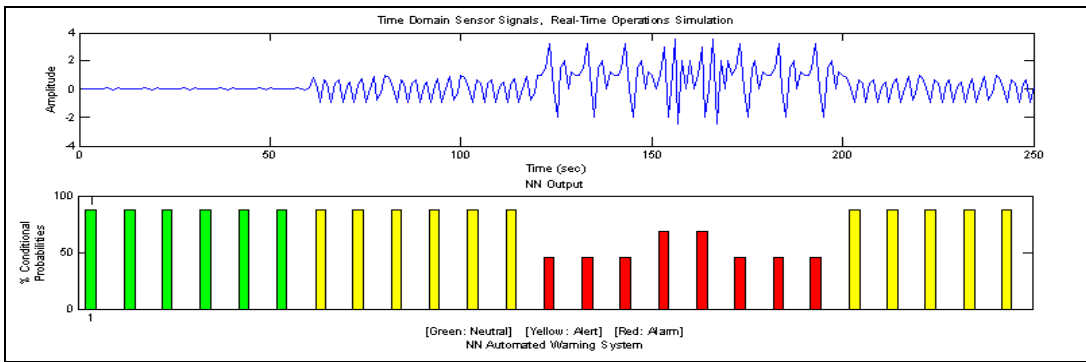


Figure 10: Real-time Autonomous Detection Simulation trial, Possibility 2.

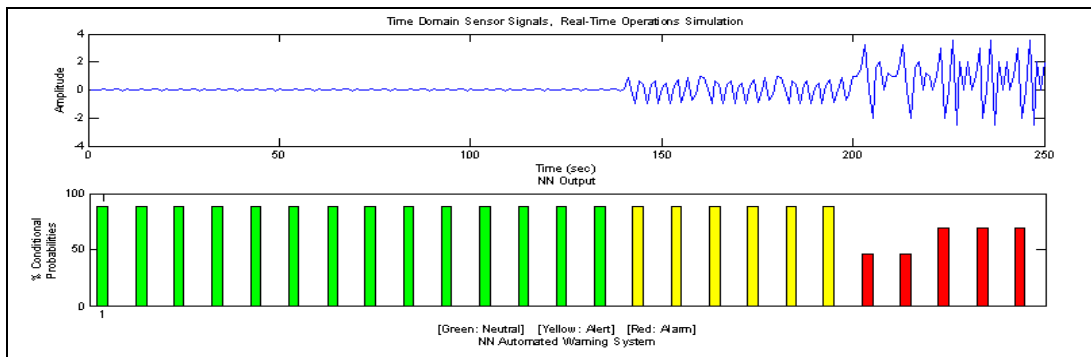


Figure 11: Real-time Autonomous Detection, Simulation trial Possibility 3.

There are two windows shown in Figures 9, 10 & 11, described as follows:

The first window represents a real-time signature to be captured by controller and the second window symbolizes a dimensional sub-space of the neural networks autonomous detection and classification process in real time. In every detected segment the percentage of conditional probabilities (confidence factor) has been estimated.

It is evident that since the adaptive systems are associated with uncertainties, due to the environmental situation there may be the possibility of the noise-level-variations within underwater real-life situation. Therefore, due to the circumstances, the system should be calibrated and validated if such a case occurs. The model here corresponds to three out of numerous possible scenarios to be measured via an autonomous detection system. The uses of the best-trained coefficient-weights can be effective to enhance the output classification probabilities accuracy (see Figures 4, 5, 6 & 7).

9. DISCUSSION AND CONCLUSION

In this paper an AUV robot autonomous detection system for the purpose of detection and tracking of underwater object(s) (at rest) and/ or moving object(s), using a signal detection and classification procedure and its practical limitations are discussed. The on-line signal preparation and treatment as well as underlay functions of the neural networks are also considered.

The simulated networks are trained to be able to distinguish and classify object(s) within several regions on the basis of the noise-density and its magnitude. The limits of the networks at the time can be due to the limited training coefficients data. The system also can be trained and expanded to map more sensitive signatures.

In this work a real-time monitoring simulation methodology is used to investigate the AUV robot's on-line autonomous-control-navigation with applying the safety detection classification process and its applications. A simulation case study was addressed and contributes towards further evidence of the self-stabilization of the neural-network learning process. Further more the resulting network can be also tested on a wide variety of different scenarios with similar sort of noise densities.

Bibliography

- Anvar, Amir (1999). Object Oriented Fuzzy Expert System Techniques for Detection, Classification and Background Historical Tracking of Aural Sonar Sources, *Technical Report (U)*, Internal Publication.
- Estebon, Michele D. (1997). Perceptrons: An Associative Learning Network, *University of Craiova*, <http://ei.cs.vt.edu/~history/Perceptrons.Estebon.html>.
- Howell, B.P., Wood, S. and Koksai, S. (2003). Passive Sonar Recognition and Analysis Using Hybrid Neural Networks, *OCEANS 2003*. Proceedings, Volume: 4, 22-26, Pages: 1917 – 1924.
- Madkour, A. A. M., Hossain, M. A., Dahal, K. P. and Yu, H. (2004). Real-time System Identification using Intelligent Algorithms, *Proceeding of IEEE SMC UK-RI Chapter Conference on Intelligent Cybernetic Systems*.
- Shin, M., Sargent, R. G., Goel, A.L (2002). Gaussian Radial Basis Functions for Simulation Methodology, *Proceedings of the Winter Simulation Conference*.
- Yerramalla S., Cukic, B. and Fuller, E. (2003). Lyapunov Stability Analysis of Quantisation Error for DCS Neural Networks, *Int'l Joint Conference on Neural Networks*.
- Rumelhart D. E., Durbin R., Golden R. and Chauvin Y. (1995). Backpropagation: the basic theory. In Y. Chauvin and D. E. Rumelhart (Eds.) *Backpropagation: Theory, Architectures, and Applications*, pp. 1-34, Lawrence Erlbaum: Hillsdale, NJ.