Refereed

The Prediction of Ship Motions and Attitudes using Artificial Neural Networks

A. A. Khan[*], K. E. Marion[†] and C. Bil[*]

Abstract

Due to the random nature of the ship's motion in an open water environment, the deployment and the landing of air vehicles from a ship can often be difficult and even dangerous. The ability to reliably predict the motion will allow improvements in safety on board ships and facilitate more accurate deployment of vehicles off ships. This paper presents an investigation into the application of artificial neural network methods trained using singular value decomposition and conjugate gradient algorithms for the prediction of ship motion. It is shown that accurate predictions of up to ten seconds can be achieved.

Introduction

This paper presents an algorithm that is designed to be applied to the prediction of ship motion. The algorithm is intended to be used to aid the successful deployment of aircraft vehicles that are currently used on ships that operate in open sea environments. The future ship motion is likely to aid deployment and subsequently recovery of air vehicles from ship platforms. The motion of a ship in an open water environment is the result of complex hydrodynamic forces between the ship, the water and unknown random processes. This leads to the necessity to use statistical prediction methods for the prediction of this motion rather then a deterministic analysis, which would lead to a ship specific model that involves highly complex calculations [1], a lot of ship specific information which will make portability of the algorithm difficult and will require a significant simplifications and assumptions which is likely to introduce errors. Past attempts at ship motion prediction [2, 3, 4, 5] have shown that traditional statistical prediction techniques such as the autoregressive moving average models and Kalman filters are unable to maintain a high degree of accuracy when the prediction interval is increased above 2-3 seconds when predicting ship motion in high sea states of 5 and above. The traditional statistical techniques used for time series prediction have difficulty dealing with noisy data, do not have much parallelism and fail to adapt to circumstances. This paper explores the use of artificial neural networks which is a form of artificial intelligence to develop an algorithm that is capable of predicting ship motions. Artificial neural networks, in contrast to traditional statistical techniques, promise to produce predictions with high accuracy as well as high efficiency due to their ability to learn and adapt according to the conditions present.

Artificial Neural Networks

A neural network is simply a series of neurons that are interconnected to create a network [6]. Artificial neural networks (ANN) have been inspired by biological neural networks. The use of ANN in time series prediction relate to the application of ANN for the nonlinear system identification. The use of ANN is particularly appealing because of their ability to learn and adapt which will be important for this investigation as one of the underlying goals is to create an algorithm that is able to work in all conditions and environments. The ANN architecture that will be used to create the ANN for time series prediction will be the multi-layer feed-forward ANN. This type of architecture has a minimum of two layers consisting of the input layer and the output layer. In this investigation a three layered feed forward neural network consisting of an input layer, a hidden layer and an output layer is used. In a feed-forward ANN the inputs for each layer come from the preceding layer. A single neuron is shown in Figure 1. It has n inputs including a bias term, which has been set to

[*] School of Aerospace, Mechanical and Manufacturing Engineering, RMIT University, Melbourne, Australia. mailto:ameerkhan@optusnet.com.au
[†] School of Mathematical and Geospatial Sciences, RMIT University, Melbourne, Australia.

1 in this investigation. The inputs are each multiplied by their corresponding weight value, which are summed together and subsequently entered into an activation function. The output of the activation function will correspond to the output of the neuron.

$$\left.\begin{array}{c} I_0.w_0 \\ I_1.w_1 \\ I_2.w_2 \\ \vdots \\ I_{n-1}.w_{n-1} \\ 1.w_n \end{array}\right\} Sum \rightarrow f(Sum) \rightarrow Output)$$

Figure 1: A representation of a single neuron

Mathematically, the output of a neuron is given as:

$$out = f(net) = f\left(\sum_{i=0}^{n-1} x_i w_i + w_n\right) \tag{1}$$

where the inputs are $x_i, i = 0, \cdots, n-1$. Generally the neuron's operation is not affected significantly by the activation function (f (net)) but the training speed is affected somewhat [7]. The activation function is usually a non-linear function that will determine the output of the neuron. Its domain is generally all real numbers. The range of the output for an activation function is usually limited between 0 to 1 and sometimes -1 to 1. The majority of activation functions use a sigmoid (S shaped) function. In this investigation the activation function shown below was primarily used:

$$f(net) = \tanh(net) \tag{2}$$

The use of ANN for time series prediction has a number of distinct advantages. Firstly, there is also no need to choose any particular model for the ANN. A validation process is included to ensure that the ANN is working correctly. If the architecture of the ANN is poorly designed, the ANN may be able to learn irrelevant details specific to the training set which will lead to an ANN that is only relevant to the training set. Conversely, the ANN may have a deficient architecture where the ANN is not able to learn the subtleties required for accurate outputs. The validation process should reveal these problems and is discussed in detail in section 2.2.

Training the Network

The training of the network can be viewed as a minimization process where the weights in the ANN are systematically adjusted in a manner that reduces the error between the output of the ANN and the desired output. Therefore the process of training the neural network becomes an optimization problem where the performance of the neural network will be dependent upon the quality of the solution found after the training process has been completed. The aim of this investigation is to develop a methodology to predict ship motion in real time. Singular value decomposition (SVD) is a linear regression technique that can quickly obtain an approximate set of optimum weights which is far superior to randomly generating the weights. A detailed description of the SVD technique is beyond the scope of this paper but essentially the matrix $\mathbf{X}$ which satisfies the function:

$$\mathbf{A}.\mathbf{X} = \mathbf{B} \tag{3}$$

when A and B are known can be calculated efficiently using SVD. When applying it to the ANN process the weights between the input layer and the hidden layer are initially randomly generated. The training samples are then inserted into the ANN and the hidden layer activation functions are calculated creating a matrix equivalent to A. Also, the values for the inverse transfer function of the output are also calculated creating a matrix equivalent to B. Applying SVD and solving Equation 3, the approximate optimal weights X are found. The conjugate gradient (CG) algorithm created for the ANN in this investigation was based on the Polak-Ribiere algorithm and is used with the SVD method. The mathematical justifications for the algorithm are beyond the scope of this paper but a detailed description can be found in Polak [15]. In a general sense, the algorithm generates a sequence of vectors and search directions. It can be shown that the exact minimum will be obtained if the multi-dimensional function can be expressed as a quadratic. The ANN error function is quadratic close to the minimum so it is expected that once close to a minimum, convergence to the local minimum will be very rapid [10]. Therefore the best values returned from running the SVD can be used as the initial starting point for the conjugate gradient search.

Validation of ANN model

To ensure that the weights in the ANN have been correctly set and that the output of the ANN is sufficiently reliable, a validation process is applied after training has been completed. The set of known inputs with their desired output needs to be divided into two distinct sets. The first set is the training set and is used throughout the training period to adjust the weights to the appropriate values as discussed previously in section 2.1. The second set is referred to as the validation set and is used to test the ANN. Once the values of the training set have been determined, the inputs from the validation set are inserted into the ANN and the output of ANN is compared with the target values in the validation set. The validation process is included to ensure that the ANN is working correctly and to ensure that the ANN has not overfitted the data. The architecture of the ANN refers to the number of neurons that are used in the input and hidden layers. If the architecture of the ANN is poorly designed, the ANN may be able to learn irrelevant details specific to the training set which will lead to an ANN that is only relevant to the training set. Conversely, the ANN may have a deficient architecture where the ANN is not able to learn the subtleties required for accurate outputs. The validation process should reveal these problems. The entire ANN process including the validation process is shown in Figure 2.

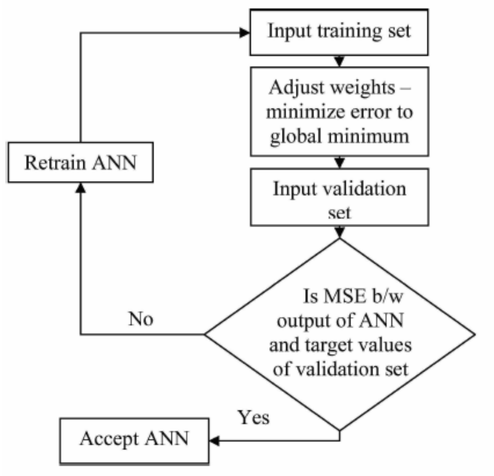

Figure 2: The ANN process

It can be clearly seen that the first stage involves inputting the training set into the ANN. The ANN adjusts its weights in the 'learning' process until the error between the target values and the output of the ANN is reduced to a minimum. Next, the validation set is inputted into the ANN. The output of the ANN is compared to the target values of the validation set and the ANN is accepted if the error is of a low enough value or alternatively rejected if the error is too high. The error in the validation set may be higher than the error found at the end of training but should not be significantly larger or there is a problem with the ANN. Also, the validation set should be independent to the training set to ensure that there is no bias added into the validation process. It is not permissible to use any of the training data in the validation stage, as this will not give a good indication of the ANN's validity.

Application of ANN to Ship Motion Prediction

The algorithms developed were subsequently applied to measured ship roll angle data taken from a cruiser size vessel operating in sea states 5-6. The term sea state is a description of the properties of sea surface waves at a given time and place [14]. The greater the sea state the rougher the conditions. The results shown in this section are the average results obtained by applying the algorithm to four separate ship motion databases. Each database had approximately 600 seconds of roll angle data available sampled at 2Hz. The training data was set to two thirds of the data sets and the validation set was designated as the final third of the data sets. All results shown in this section are the predictions made using the validation set only. A graphical representation of the results is shown in Figure 3 while an example prediction is shown in Figure 4.

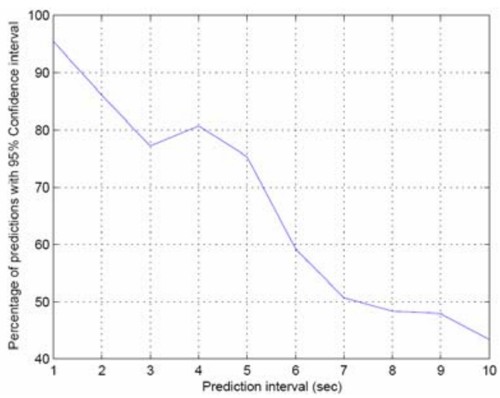

Figure 3: Prediction of roll motion (frigate class ship in sea state 5-6)

In this investigation the ANN had eight neurons in the input layer and three neurons in the hidden layer. The output layer naturally only had one neuron. Therefore, for every lead prediction interval a single ANN is used. The ANN is capable of creating multiple predictions of different magnitude but it is better that a separate ANN is used for every prediction interval. The basis for the presumption is that the weights for an optimal prediction will vary according to the prediction interval desired. By having the ANN create multiple predictions, the overall optimal prediction cannot be made. By having separate ANN create separate predictions, the optimal weight configuration can be obtained for each prediction and therefore, higher accuracy can be expected. During the training process thirty separate trials were conducted using the SVD training algorithm and then the best solution was chosen and inserted into the CG algorithm as the initial starting point for the CG search. Figure 3 shows that at low prediction intervals the accuracy levels are extremely high and the quality of the predictions reduces as the prediction interval is increased. This is as expected. One of the anomalies that can be seen in Figure 3 is that the 4 second prediction is better than the 3 second prediction. The explanation for this anomaly relates to the operation of the SVD algorithm. As stated in section 0, when applying the SVD to the ANN process the weights between the input layer and the hidden layer are initially randomly generated. Therefore the ANN was able to produce exceptionally high levels of accuracy for one of the databases because it had a very good set of randomly generated weights and therefore the overall average was very high. To ensure that the ANN has be best set of initial weights it would be necessary to conduct as many trials as possible of the SVD as each trial would have a unique set of input weights and potentially lead to better overall accuracy. However, there is an associated cost in the form of computational processing time. It can be clearly seen in Figure 3 that the accuracy level when the prediction interval is ten seconds is approximately 40%. This may seem quite poor but if Figure 4 is examined it is evident that the ship motion is still very well represented.

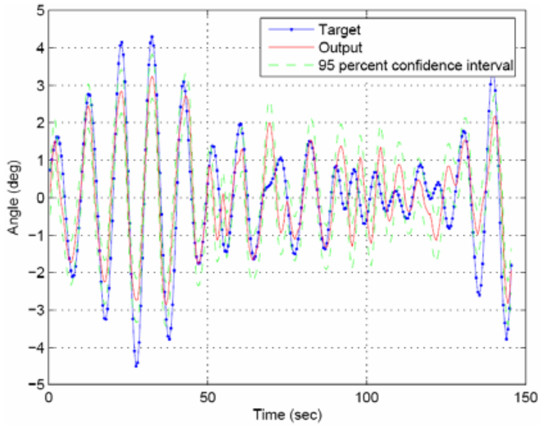

Figure 4: Sample 10 second in advance prediction generated using the ANN algorithm with 8 neurons in the input layer

The important aspect that should be considered with regards to Figure 4 is that the large fluctuations in the amplitude of the motion are well represented. The ANN was also able to predict the region of motion and 3 neurons in the hidden layer where the amplitude is not large which is equally important. For example, if one wanted to land a helicopter on a ship deck the pilot would be interested to know when the roll motion would have a large amplitude as this would make it unsafe to land, but equally important is knowing when the amplitude is low as this would mean that it would be safer to land. Therefore, as the motion of the ship as the low and high amplitude motion is predicted, even in predictions of up to 10 seconds, the ANN algorithm trained using a combination of the SVD and CG algorithms is very effective.

Conclusion

In this paper an artificial neural network based method utilizing a combination of the singular value decomposition and conjugate gradient algorithm for the prediction of the ship motion was presented. It was shown that the artificial neural networks were capable of learning the motion and producing accurate predictions for intervals up to 10 seconds. The most important outcome of the investigation was that the high and low amplitude motion was very well represented for all prediction interval lengths.

Acknowledgements: The first-named author would like to thank BAE Systems for providing funding and support for the project.

References

[1] Price, W. G.,Bishop, R. E. D.: Probabilistic theory of ship dynamics, Chapman and Hall Ltd, Salisbury, (1974)

[2] Yumori, I.: Real time prediction of ship response to ocean waves using time series analysis, OCEANS, 13, No. (1981), 1082{1089

[3] Cortes, N. B.: Predicting ahead on ship motions using Kalman filter implementation, RMIT University, Melbourne, (1999)

[4] Crump, M. R.: The Dynamic and Control of Catapult Launching Unmanned Air Vehichles from Moving Platforms, RMIT University, Melbourne, (2002)

[5] Sidar, M. M.,Doolin, B. F.: On the feasibility of real-time prediction of aircraft carrier motion at sea, IEEE Transactions on Automatic Control, 28(3), (1983), 350{356

[6] Suykens, J. A. K., Vandewalle, J. P. L., Moor, B. D.: Artificial neural networks for modeling and control of non-linear systems, Kluwer Academic Publishers, Dordrecht, (1996)

[7] Masters, T.: Practical neural network recipes in C++, Academic Press, Boston, (1993)

[8] Obitko, M.: Genetic algorithms, Prague, 1998, Viewed Access (1998), http://cs.felk.cvut.cz/»xobitko/ga/

[9] Rumelhart, D., McClelland, J., Group, P. R.: Parallel distributed processing, MIT Press, Cambridge, (1986)

[10] Press, W. H., Flannery, B. P., Vetterling, W. T., Teukolsky, S. A.: Numerical recipes in Fortran: The art of scientific computing, Cambridge University Press, New York, (1992) 9

[11] Polak, E.: Computational methods in optimisation, Academic Press, New York, (1971)

[12] Haupt, R. L., Haupt, S. E.: Practical Genetic Algorithms, John Wiley and Sons, (1998)

[13] Man, K. F., Tang, K. S., Kwong, S.: Genetic Algorithms, Springer Verlag, London, (1999)

[14] Keefer, T.: Glossary of Meteorology, Allen Press., Boston, 2005, Viewed Access (2005), URL: http://amsglossary.allenpress.com/glossary

[15] Polak, E.: Computational methods in optimisation, Academic Press, New York, (1971) 10

19th National Conference of the
Australian Society for Operations Research
Melbourne, Victoria
3 – 5 December, 2007

The conference provides a forum for optimisation researchers and operations research practitioners to exchange ideas and discuss their latest research and applications.

Full details will be available soon on the conference website which is being established. Further information can be obtained from A/Prof. Baikunth Nath (baikunth@unimelb.edu.au) and Prof. Santosh Kumar (skumar@csm.vu.edu.au).