# SHIP-MOTION PREDICTION: ALGORITHMS AND SIMULATION RESULTS

X. Zhao, R. Xu and C. Kwan
Intelligent Automation, Inc.
7519 Standish Place, Suite 200
Rockville, MD 20855, USA
ckwan@i-a-i.com

**Abstract**

Ship-motion prediction will be very useful for several naval operations such as aircraft landing, cargo transfer, off-loading of small boats, and ship "mating" between a big transport ship and some small ships. The prediction information will be extremely useful in sea states above 3. Five to ten seconds ship motion prediction can give the operator ample time to avoid serious collisions.

This paper summarizes the development of a high performance ship-motion prediction algorithm using Minor Component Analysis (MCA). Simulation results showed this method can predict ship motion a long time ahead with consistent accuracy. That is, the prediction error is almost the same for the 5 second and 20 second predictions. Other conventional algorithms like Neural Network (NN), Autoregressive method (AR), Wiener prediction were also studied for comparative purpose.

## 1. Introduction

There are many ship operations that require ship motion prediction [1]. First, the landing of aircraft in carriers and helicopter in a destroyer is an important application. In high sea states, the landing operation may be quite dangerous. Second, side by side cargo transfer is another application. One crane may off-load some dangerous ammunition from a big ship to a small ship. Again the operation can be quite dangerous in high sea states (above sea state 3). Third, in some naval operations, there are some "mating" operations between a large transport ship and some small ships. The large ship is floating far from shore and the small ships go back and forth between the large ship and the shore. During high sea states, it may be difficult for the small ships to go inside the large ship.

In all of the above applications, ship motion prediction will be very important because if one can look 5 to 10 seconds ahead of time, then one can time the landing to avoid a serious crash in the case of aircraft landing. For cargo transfer, the prediction information may prevent crash of cargo that may explode. Finally, a smooth "mating" between ships also requires some prediction information of ship motion.

In this paper, a prediction algorithm based on MCA was developed and implemented for ship-motion prediction. The algorithm takes past motion data for training, and predicts the data in the next 10 or 20 seconds based on the recent motion data. The algorithm can update its core from time to time to accommodate the sea state or ship speed change. To evaluate the performance of this predictor, numerical simulations were carried out and compared with conventional algorithms like NN, AR, and Wiener prediction.

## 2. MCA based prediction algorithm

MCA has similar mathematics as the Principal Component Analysis [2], except that MCA utilize the eigenvectors corresponding to the minor components. Specifically, the motion data was aligned into a sequence of vectors $X_i$, *i=1... N*. Then, the eigenvectors and eigenvalues of the following autocorrelation matrix was calculated

$$R = \sum_{i=1}^{N} X_i X_i^T \tag{1}$$

This will yield the *n* eigenvalues $\lambda_0$, ..., $\lambda_{n-1}$, and the associated eigenvectors $u_0$ ,..., $u_{n-1}$ . Now we choose $P$ eigenvectors corresponding to the $P$ smallest eigenvalues and form a matrix $B \in R^{P \times N}$ , which can in turn divides into 2 sub matrices, $B = [B_1, B_2]$, with $B_1 \in R^{p \times n_1}$ and $B_2 \in R^{p \times (N-n_1)}$ . Here $n_1$ is usually chosen to be larger than 2/3 of N. Thus we can rewrite each $X_i$ , *i=1,...,N.* as $X_i = \begin{bmatrix} X_{1i} \\ X_{2i} \end{bmatrix}$, where $X_{1i} \in R^{n_1}$ and $X_{2i} \in R^{N-n_1}$ .
According to MCA procedure, the following approximate equalities hold

$$B_1 X_{1i} + B_2 X_{2i} \approx 0 \tag{2}$$

$$X_{2i} \approx -(B_2^T B_2)^{-1} B_2^T B_1 X_{1i} \tag{3}$$

Thus Eq. (3) can be used to perform ship motion prediction. The dimension of $X_{2i}$ will be the length of the prediction window. $X_{1i}$ is a vector containing the past data.

## 3. Simulation Studies

### Ship motion prediction

The ship motion data was provided by JJMA, Inc. which has simulation software fully tested and evaluated by experienced Naval officers. We used the DDG-51 destroyer in all the data files. The ship is moving at a low speed (10-12 knots), heading 15 degrees against the wave direction; Ship motion status (surge, sway, heave, roll, pitch, yaw) were sampled at 8Hz rate at sea state 3 which has a wave height of about 10 ft.

To implement the above algorithm, the ship motion data were first down-sampled to 2 points per second. No information was lost because the ship moved rather slowly. The data were then separated into two segments: the first 5000 points of data were used for MCA training and the rest for testing. During the training, the size of the data window was 800 points, and each vector was obtained by shifting the window by 10 points. So altogether there were 420 vectors used for training. The minor components were selected in a sense that their energy added up to no more than 1.5% of the total energy. For the testing part, 750 past values were collected for predicting the next 50 points. So this is a 25 second prediction (longer prediction is also possible with longer training data and less accuracy). In the next several figures, the predicted and the real ship motions (surge, sway, heave, roll, pitch, and yaw) are plotted versus time. Although we predicted 50 points ahead of time, the plots only show the 40th point (20 seconds ahead) in the prediction window. The green lines are the real motions; the red ones are the predicted.

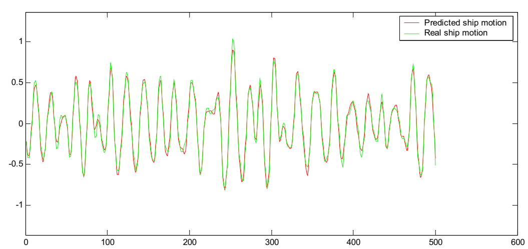


Fig. 1 MCA prediction of roll 20 seconds ahead

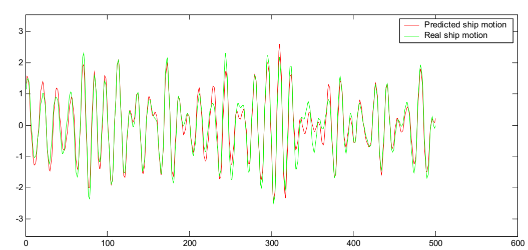


Fig. 2 MCA prediction of pitch 20 seconds ahead

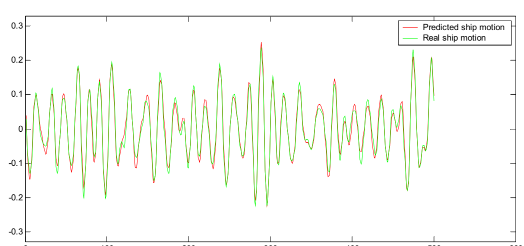


Fig. 3 MCA prediction of yaw 20 seconds ahead

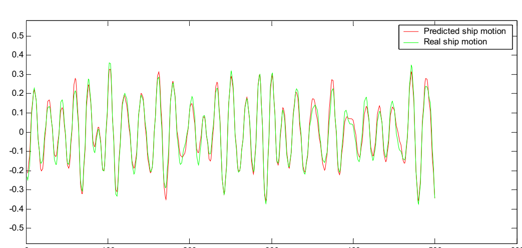


Fig. 4 MCA prediction of surge 20 seconds ahead

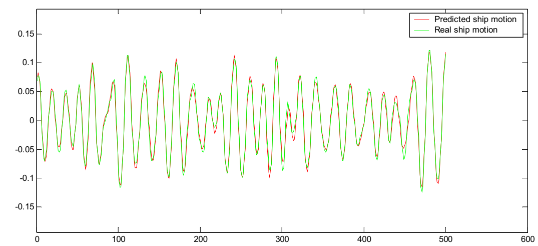


Fig. 5 MCA prediction of sway 20 seconds ahead

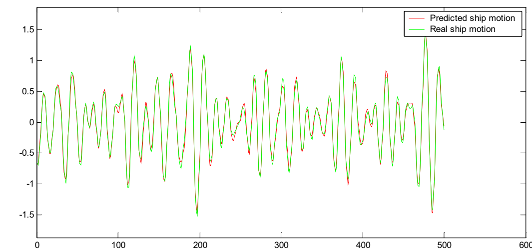


Fig. 6 MCA prediction of heave 20 seconds ahead

It is seen that the 20th second prediction data using MCA matches quite well with the real ship motion for all the six degree-of-freedom (DOF) in a decoupled manner [3]. It can predict the next 50 points simultaneously, which can be used to improve the prediction accuracy by using the mean value of several predictions for the same future point. And within the prediction window (in this case 50 points), the error keeps at the same level (Table 1), i.e., the prediction error for the last point is not necessary worse than the first point and the prediction error is small.

**Comparative studies**

Several other conventional prediction methods were utilized to compare the MCA predictor, e.g. NN, AR, and Wiener predictor. A brief description of those predictors is listed below.

*Neural network (NN)*

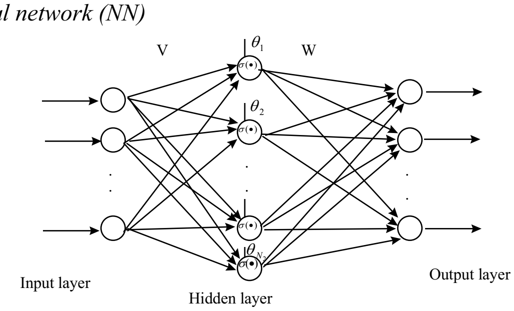


Fig. 7 Ship motion forecast using three-layer NN

Three-layer neural network shown in Fig. 7 was used for prediction. For a given input vector $x$ in $R^{N_1}$, the NN has an output given by

$$y_i = \sum_{j=1}^{N_2}\left[ w_{ij}\sigma\left[\sum_{k=1}^{N_1} v_{jk}x_k + \theta_{vj}\right] + \theta_{wi}\right], \qquad i = 1,\ldots, N_3 \quad (4)$$

with $\sigma(\bullet)$ the activation function such as a sigmoid function, $v_{jk}$ the first-to-second layer interconnection weights, and $w_{ij}$ the second-to-third layer interconnection weights. $\theta_{vm}, \theta_{wm}, m = 1, 2...$, are called the threshold offsets and the number of neurons in layer $l$ is $N_l$, with $N_2$ the number of hidden-layer neurons. In MATLAB code, function newff(IND,[100 M],{'tansig' 'purelin'},'trainrp') was used for training and testing.

*Autoregressive model (AR)*

Based on the past $n$ points in the ship motion data, an AR predictor predicts the next value of ship motion based on the past $n$ data sample

$$\hat{y}_k = a_1 y_{k-1} + a_2 y_{k-2} + ... + a_n y_{k-n} \tag{5}$$

where $y_{k-1}, y_{k-2}, ..., y_{k-n}$ are the past data samples. The key here is to obtain the AR coefficients $A(k) = [a_1(k) \quad a_2(k) \quad .... \quad a_n(k)]^T$ The on-line scheme to update A(k) is given by

$$A(k) = A(k-1) - ge(k)\begin{bmatrix} y_{k-1} \\ y_{k-2} \\ .... \\ y_{k-n} \end{bmatrix} \tag{6}$$

where $e(k) = y(k) - \hat{y}(k)$ is the prediction error and $g$ is gain factor which should be a small value. If g is too large, instability may occur. In our simulations, we set g = 0.005. To generate many future predictions, several AR predictors are needed with the first one performing a one-step prediction, the second one performing a two-step prediction, and so on and so forth

*Wiener Predictor*

An FIR Wiener predictor of order p-1 for multi-step linear prediction has the form

$$\hat{x}(n+\alpha) = \sum_{l=0}^{p-1} w(l) x(n-l) \tag{7}$$

where $\hat{x}(n+\alpha)$ is the predicted value of the $\alpha$ th point ahead and $x(n-l)$ for $l$ =0,1,...,p-1 are the past p data values, $w(l)$ are the coefficients of the prediction filter. The Wiener prediction problem requires that we find the filter coefficients that minimize the mean-square error

$$\xi = E\{|e(n)|^2\} = E\{|x(n+\alpha) - \hat{x}(n+\alpha)|^2\} \tag{8}$$

where $x(n+\alpha)$ is the true value of the $\alpha$ th data point ahead. The necessary and sufficient condition is the derivative of $\xi$ with respect to $w^*(k)$ be equal to zero for k=0, 1... p-1, thus we can form the following Wiener-Hopf equation:

$$\sum_{l=0}^{p-1} w(l) r_x(k-l) = r_x(k+\alpha); \quad k = 0, 1, ..., p-1 \tag{9}$$

Once the weights $w(l)$ are obtained, we can proceed to do the prediction based on Eq. (7).

In this study, we did three comparisons. First was the prediction accuracy. The Root-Mean-Square error was used to assess the performance of different methods based on using the same past data points to predict the same future values. Mathematically, RMS is defined as

$$RMS = \sqrt{\frac{\sum_{i=1}^{K} (x_i^T - x_i^P)}{K}} \tag{10}$$

where K is the size of the prediction window and $x_i^T$, $x_i^P$ are the true and predicted values of ship motion parameters. Second, we compared the computational complexity and/or computation time needed for each scheme. Third, the robustness of the two most accurate methods was compared. That is, the prediction performance with respect to noise and disturbances.

*a) Prediction accuracy comparison*

| Prediction method | Modal order (past data number) | RMS error for 0.5s ahead (m) | RMS error for 5s ahead (m) | RMS error for 20s ahead (m) |
|---|---|---|---|---|
| MCA | 40 points | 0.1507 | 0.4419 | Impossible |
| | 100 points | 0.1245 | 0.4678 | 0.7022 |
| | 800 points | 0.0466 | 0.0538 | 0.0540 |
| NN | 40 points | 0.1484 | 0.3303 | 1.2900 |
| | 100 points | 0.4437 | 0.5531 | 1.0095 |
| | 800 points | 0.5747 | 0.5287 | 0.4724 |
| AR | 40 points | 0.0576 | 0.9026 | 1.6180 |
| | 100 points | 0.1050 | 1.0726 | 1.7495 |
| | 800 points | 0.0453 | 0.1207 | 0.1412 |
| Wiener | 40 points | 0.0817 | 0.3843 | Impossible |
| | 100 points | 0.0604 | 0.3469 | 0.5792 |
| | 800 points | 0.0175 | 0.1248 | 0.2743 |

Table 1 Root-Mean-Square error comparison for the four prediction algorithms. The best performers are marked in pink and the runner-ups are in yellow.

Table 1 summarizes the initial results on the RMS errors of the four prediction methods, each having three model orders and predicting 0.5s, 5s and 20s motion ahead. It is seen that the MCA approach with model order of 800 gives the best performance for predicting ship motion 5s and 20s ahead of time. It also yields a good performance for predicting 0.5s ahead of time. The error within the prediction window (1 ~40 points) maintains at roughly the same low level. This is because MCA treat the whole prediction window as a "missing" part of the signal space, whereas the minor components act as noise space. The orthogonal relationship between noise space and signal space lead the prediction to be as accurate as possible for the whole window from a mean square point of view. The runner-up is the AR model with an order of 800, which predicts both short term and long term ship motion quite accurately. The 800th order Wiener filtering approach has the best accuracy in predicting 1 point ahead, but is weak in the long term prediction due to the loosened correlation.

*b) Computational complexity and/or computation time comparison*

Table 2 summarizes the comparison between various prediction algorithms from the computational complexity point of view. For each prediction, 10 points and 40 points (5 s and 20 s, respectively) of prediction were performed. The offline training time and the on-line prediction time (500 predictions were made) and computational complexity were compared with MATLAB program on a Pentium 1.6 GHz PC. It is seen that the training time for MCA is short; and it is also the fastest method among the four prediction methods. NN approach requires long training time. AR model is fast in short term predictions. However, both its training time and prediction time are proportional to the prediction data length.

Wiener filter approach does not require offline training, yet it is very slow in motion prediction, especially for long term.

| Prediction method (800 data points) | Prediction length | Offline training time | prediction time (500 predictions) |
|---|---|---|---|
| MCA | 10 points | 40 seconds | 0.09 seconds |
| | 40 points | 40 seconds | 0.09 seconds |
| NN | 10 points | 1735 seconds | 3.44 seconds |
| | 40 points | 3467 seconds | 3.68 seconds |
| AR | 10 points | 11.4 seconds | 1.6 seconds |
| | 40 points | 47 seconds | 6.4 seconds |
| Wiener | 10 points | Not required | 843.3 seconds |
| | 40 points | Not required | 1054 seconds |

Table 2 Comparison of computational time for the four prediction algorithms

*c) Robustness comparison*

Usually in the ship motion measurement, the sensors such as accelerometer and gyroscope data may be noisy due to sensor noise and external interference, etc. To test the robustness of the prediction methods, a zero-mean Gaussian random noise with various standard deviations was added to the ship motion data. Here we compare the AR and MCA approaches. Table 3 shows the RMS error for these two approaches when predicting the heave motion. Figure 8 shows the prediction results

| Noise level (standard deviation = peak amplitude percentage) | AR model (800 points) RMS error (m) | MCA model (800 points) RMS error (m) |
|---|---|---|
| 0% | 0.14 | 0.056 |
| 10% | 0.31 | 0.24 |
| 20% | 0.55 | 0.51 |

Table 3 Comparison of RMS error of AR and MCA model with the presence of Gaussian noise

It can be seen that both AR model and MCA degrade a bit with noise added. With 20% Gaussian noise added, the peak value prediction exhibits relative large error. However, they are quite robust in predicting the ship motion with noise level less than 10%.

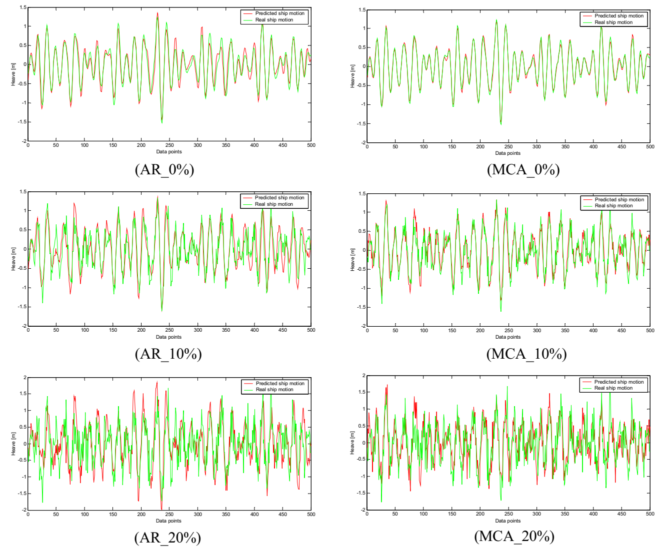


Fig. 8 Heave motion prediction (red) and actual value (green) at noisy environment. (AR_0%) means AR model prediction for 0% Gaussian noise, etc

### 4. Conclusions

A new algorithm based on MCA for ship motion prediction has been presented. Based on our evaluations with neural network, autoregressive and Wiener method, MCA based algorithm yields better performance and is also suitable for real-time implementation.

### Acknowledgment

This research was supported by the Office of Naval Research under contract N00014-02-M-0252.

### References

[1] C. Kwan et al., Phase 1 final report to the ONR, January, 2003.
[2] S. Haykin, *Neural Networks*, Prentice Hall, 1989.
[3] J. C. Chung, Z. Bien and Y. S. Kim, “A note on ship-motion prediction based on wave-excitation input estimation,” IEEE J. Oceanic Engineering, vol. 15, pp244 -250, 1990