Comparative analysis of algorithms for ship pitching forecasting

D Antonov

1PhD Student, JSC "Elektropribor", St.-Petersburg, Russia

E-mail: danila.v.antonov@mail.ru

Abstract. An overview and analysis of various ship motion prediction algorithms based on its description by random or quasideterministic processes are carried out. The testing of real data is conducted to estimate the performance of algorithms.

1. Introduction

To enhance the navigation safety during the ship motion, motion prediction is often required in the algorithms of various systems. This need occurs for example when handling large-sized fragile cargo in the open sea, working on oil and gas production platforms, aircraft deck or platform landing [1-3].

The applied algorithms can be divided depending on the model used for motion prediction. Most often a random process is used to synthesize a model, usually a second or third order Markov process. Also a quasideterministic process can be applied, usually represented as a time series, a set of sinusoids, or a polynomial model. Sometimes the process is presented in the form of a regular harmonic function of time to primarily estimate the algorithm performance or to conduct initial calculations.

To solve the prediction problem the signal should be presented in a stationary form, which is an important general characteristic of the process. As noted in [2], this assumption can be fulfilled over the intervals (20 - 40 min) [2].

Currently multiple studies on motion prediction algorithms are conducted, usually for the time interval from 5 to 10 seconds. In [3, 4] the authors present the ship motion prediction algorithm using the Kalman filter (KF) for the carrier landing systems. In [3] the KF is used only to estimate the motion angle while in [4] filtering of the extended (due to the ship characteristics) fourth-order state vector is presented. In [5-6] ship motion prediction algorithms using the autoregressive method are proposed. These algorithms estimate the coefficients of the autoregressive model. The coefficients are estimated based on a simpler presynthesized model in [5], and in [6], the Kalman filter is applied to adapt the coefficients and constantly update them during the prediction. Now the algorithms for predicting various time series based on artificial neural networks (ANN) – also used to predict the ship motion – have gained popularity [7-10, 17]. An important stage in prediction based on ANN is the process of network training. The authors in [7] use the ANN training method based on a genetic algorithm as one of the most accurate for finding the global minimum RMS error between the ANN and real values. The significant drawback of this algorithm is that it is highly resource-intensive, so it is proposed to use two synchronized computers to ensure its operation. To speed up the algorithm and reduce the convergence time in [8] an extreme learning machine algorithm is applied, whose strong points include much lower computing resource consumption and faster operation as compared to deep learning ANN with recurrent connections. However, in these networks the weights of hidden nodes should be initially accurately set, and they cannot be retrained. In [9] an algorithm based on support
vector machine is proposed, which is close to ANN algorithm. This algorithm is often applied for machine learning and features high speed and relatively high accuracy. In [10], the deep learning algorithm is employed (using a radial basis function), and the motion signal is divided into several harmonic components using wavelet transformation and their separate prediction with further composition.

This article is devoted to the research of the motion prediction algorithms and comparison of their accuracy. Three algorithms are presented in the paper: the first one based on a second-order Markov process model using a KF to estimate the motion prediction (KF-based algorithm), the second one based on autoregressive model, and the third one based on ANN with long-term short-term memory (LSTM). The algorithms have been tested on real data.

2. Description of Algorithms

2.1. The KF-based algorithm

This algorithm uses the model of ship motion (namely pitching) in the form of random process. Within the wave theory the motion \( u(t) \) can be also considered to be a fourth-order process, however for large ships the major motion characteristic is the natural frequency of the ship motion [2]. Therefore in this paper the pitch is described as a narrow-band second-order Markov process with a correlation function given by

\[
k(\tau) = \sigma^2 e^{-\alpha|\tau|} \left( \cos \beta \tau + \frac{\alpha}{\beta} \sin \beta \tau \right).
\]

where \( \sigma \) is the variance, \( \alpha \) is the damping coefficient and \( \beta \) is the frequency of correlation function variation.

In describing the ship pitch, parameters \( \alpha \) and \( \beta \) can be obtained from knowledge of the dominant pitch frequency \( \omega \) [2]:

\[
\beta = 0.82 \omega, \quad \alpha = 0.21 \beta.
\]

The dominant pitch frequency \( \omega \) can be determined using the spectral analysis of the signal recorded at the previous time interval. The process described by the correlation function (1) can be represented by a shaping filter for two-dimensional vector \( x(t) \), one of the components of which, for example the first one, is the pitch angle \( u(t) \), i.e. \( u(t) = x_1(t) \). In this case the fundamental matrix \( \phi(\tau) \) is defined as [11]:

\[
\phi(\tau) = e^{-\alpha|\tau|} \begin{bmatrix}
\cos \beta \tau - \frac{\alpha}{\beta} \sin \beta \tau & \frac{1}{\beta} \sin \beta \tau \\
\left(\frac{\alpha^2 + \beta^2}{\beta}\right) \sin \beta \tau & \cos \beta \tau + \frac{\alpha}{\beta} \sin \beta \tau
\end{bmatrix}.
\]

(3)

The measurements of the KF can be written as:

\[
y = u(t) + \varepsilon(t),
\]

where \( \varepsilon(t) \) is the white noise. The transition matrix of KF with account for (3) has the form
\[
F = \begin{bmatrix}
0 & 1 \\
-(\alpha^2 + \beta^2) & -2\alpha
\end{bmatrix},
\]

(5)

where \(\Delta t\) is the sampling time.

To estimate the pitch prediction \(\hat{x}(t+\tau)\) at time \(\tau\) the algorithm uses the following equation:

\[
\hat{x}^{KF}(t+\tau) = \phi(t+\tau)\hat{x}^{KF}(t)
\]

(6)

where \(t\) is the current time, \(\tau\) is the prediction time, \(\hat{x}^{KF}(t)\) is the pitch prediction estimate calculated using the known KF equations.

2.2. The autoregressive model based algorithm

An autoregressive model is a time series model in which the current time series values linearly depend on the previous values of the same series. The main pitch parameters are assumed unchanged during the prediction time.

When using autoregressive models the process can be described as [5]:

\[
u(t) = \sum_{i=0}^{p-1} a_i u(t-i\Delta t) + \varepsilon(t).
\]

(7)

where \(\Delta t\) is the sampling time, \(a_i\) is the model coefficients, \(p\) is the model order (\(|p| < 1\) is the stationarity condition), \(\varepsilon(t)\) is the white noise.

The estimates of coefficients \(a_i, i=1,p\) are calculated by approximating the pitch process over the interval \(n\Delta t\) using the least squares method, where \(n\) is the number of points in the sample. To synthesize the algorithm, matrix B of size \(p \times n\) is formed:

\[
\begin{bmatrix}
u(1) \\
u(2) \\
\vdots \\
u(p)
\end{bmatrix} = \begin{bmatrix}0 & u(0) & \cdots & u(p) & \cdots & u(n) \\0 & 0 & \cdots & u(p-1) & \cdots & u(n-1) \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & u(0) & \cdots & u(n-p)\end{bmatrix} \begin{bmatrix}a_1 \\a_2 \\
\vdots \\
a_p\end{bmatrix} = B \begin{bmatrix}a_1 \\a_2 \\
\vdots \\
a_p\end{bmatrix}.
\]

(8)

where \(u(0)\) is the pitch angle at the initial time, \(u(p)\) is the pitch angle at time \(p\), \(u(n)\) is the pitch angle at the last recorded time point.

The parameters \(a_i\) are determined by solving the system of equations (8).

An important characteristic of an autoregressive model is its order, which has a significant impact on the result. The order of the model was determined experimentally by comparing the approximation errors for models of different orders. The order \(p=10\) was chosen in the work. With smaller values the error grew significantly.

Once the coefficients are found, the prediction for the time interval \(\Delta t\) is calculated using the following ratio

\[
\hat{u}^{AR}(t+\Delta t) = \sum_{i=0}^{p-1} \hat{a}_i u \left(t - i\Delta t\right).
\]

(9)
where \( \hat{a}_i, i = 0, (p-1) \) are the calculated estimates of the coefficients; \( u(t - i \Delta t), i = 0, (p-1) \) are the pitch angles at the previous moments of time over interval \( p \Delta t \). To obtain a prediction for time interval \( 2 \Delta t \) according to this formula \( t \) should be substituted by \( t + \Delta t \), then we can write

\[
\hat{u}^{AR}(t + 2\Delta t) = \sum_{i=0}^{p-1} \hat{a}_i u(t + \Delta t - i \Delta t).
\]  

(10)

From relation (10) it follows that with \( i = 0 \) the pitch value is needed to be known at time \( u(t + \Delta t) \). This value is lacking during the prediction, so prediction (9) is substituted instead of \( u(t + \Delta t) \). The same is done when calculating the prediction for the time interval \( 2 \Delta t \) and, more generally, for the time interval \( \tau = l \Delta t \), where \( l \) is the number of points in the prediction sample. Thus, to determine the predicted pitch angle for time \( \tau \) sample \( u(t - (p - 1) \Delta t), u(t - (p - 2) \Delta t), \ldots, u(t) \) and the predicted pitch values at times exceeding \( t \) are calculated according to the described procedure.

The proposed method is easy to implement and can be retrained, which is a significant advantage during long-term operation when the model parameters may change due to changes in wind effects [2].

2.3. The long short-term memory neural network based algorithm

When using a neural network, the long short-term memory (LSTM) model is applied characterized by the ability to learn long-term processes [12]. The LSTM networks are widely used to estimate and predict dependencies presented in the form of time series. A specific feature of this network is the presence of an LSTM module that does not use the activation function inside its recurrent components (Fig. 1). Thus, the stored value does not blur in time and the gradient does not disappear when using the error back propagation in time when training the ANN.

![Figure 1. LSTM repeating cell [13], where \( \sigma \) is the filter, \( X_t \) is the network input, \( tanh \) is the hyperbolic tangent layer, \( h_t \) is the network output.](image)

The key components of an LSTM module are the cell state (horizontal top line across the top of the diagram) and various filters. Unlike a standard recurrent network, there are several filters in a cell which interact with the input data and with the state of the cell. Filters are a complex of a sigmoidal layer and multiplication operation. By the state of the cell we mean the network memory which transmits the relevant data along the entire chain of modules, it passes through the entire chain and can remain unchanged.

Thus, even information from earlier time steps can be used at later steps, neutralizing the effect of short-term memory. Cell filters shown in Fig. 1: the forgetting filter layer, which determines whether the information from the previous output is saved (returns 0 and 1), the input filter layer, which determines which values should be updated, and the hyperbolic tangent layer to obtain new values from -1 to 1.
The output will be based on the state of the cell, and some filters will be applied to it. First, a sigmoidal layer is applied, which decides what information from the cell state needs to be transmitted to the output. The cell state values are then passed through the hyperbolic tangent layer to output values in the -1 to 1 range, and are multiplied with the sigmoidal layer output values, allowing only the information required to be output.

When training the ANN, the previously recorded ship pitch angles are transmitted to the algorithm input. An important step in constructing a model is scaling the training data, which greatly improves the quality of training. Scaling is carried out in accordance with the ratio

\[ u_s(t) = \frac{u(t) - M_u}{D}, \]

where \( u_s(t) \) is the scaled value of the ship pitch angle at time \( t \), \( M_u \) is the mathematical expectation, and \( D \) is the variance of the training dataset.

The network model consists of one input layer, one output layer and 200 hidden layers, and is trained using the gradient descent method. The output parameters after the network training are the weight coefficients of the filters within the cells of ANN hidden layers; they determine the need to forget the cell state, the size of the vector of values to be replaced, the cell state after passing the input filter, and the state fed to the output. LSTM architecture does not use activation functions that determine the neuron output value depending on the result of the weighted sum of inputs and the threshold value within its recurrent components.

To calculate the prediction, time interval \( t_{fr} \) over which the network was trained is fixed, and the scaled pitch angle \( u(t_{fr}) \) recorded over this interval is fed to the network input, thereby partially updating the state of the cells in the network and calculating the prediction \( \hat{u}^{ANN}(t + \Delta t) \) for the first prediction interval \( \Delta t \). To predict the pitch angle for interval \( 2 \Delta t \), the resulting prediction is added to the signal \( u(t_{fr}) \) fed to the network input in the previous clock cycle. More generally, to forecast for the time interval \( \tau = l \Delta t \), where \( l \) is the number of points in the predicted sample, the sample with the predicted angles preceding \( \tau \) is fed as input signal \( u(t_{fr}) \). Then the cell states are partially updated at each interval in the network, so the prediction forecast \( \hat{u}^{ANN}(t + \tau) \) is formed.

From the above it follows that to generate a prediction using a recurrent neural network it is necessary to send the desired signal recorded at the previous clock cycles to the input of the pretrained network. The network updates its parameters step by step by processing the samples \( u(t_{fr}) \) and preserving the cell state obtained when processing the previous pitch angles. The final output of the model is the sample \( \hat{u}^{ANN}(t + \tau) \) for the interval \( \tau = l \Delta t \).

3. Simulation results

A prerecorded 2 Hz sample was used as the initial signal for the ship pitch. The stationarity of the process was verified by the autocorrelation function analysis.

To obtain the pitch parameters and use them in the KF algorithm, a sample of 500 values of the ship pitch angle was used, which was also applied in ANN training. The training sample for an autoregressive model was limited to 200 values that were used to estimate the coefficients.

The root mean square (RMS) reduced error in the predicted pitch angle \( \Delta d \) was calculated:

\[ \Delta d = \left| \frac{\hat{u}^i(t + \tau) - u(t + \tau)}{u_{max}} \right| \times 100, \]
where $\hat{u}_j(t + \tau), j = KF, AR, ANN$ is the predicted pitch angle at time $t + \tau$ obtained using various methods, $u(t + \tau)$ is the real pitch angle at time $t + \tau$, and $u_{\text{max}}$ is the maximum pitch angle over the studied time interval. The RMS values were calculated for different prediction time intervals by the method of statistical tests for 30 samples.

![Figure 2](image)

**Figure 2.** Reduced RMS error depending on the prediction time, where AR is the autoregressive model, KF is the Kalman filter algorithm, and ANN is the ANN algorithm.

The obtained results presented in Fig. 2 show a significant advantage of the ANN algorithm in prediction accuracy, however the training time was about 122 s (with gradient descent optimization, and 125 s, with adam method, extended stochastic gradient descent optimization algorithm [13]). The prediction calculation time in Matlab 2019 using an Intel Core i5-8265U 1.60 GHz processor was 0.6 s. The algorithm based on the autoregressive model showed the average accuracy in terms of the prediction RMS error.

RMS errors of the pitch prediction for three analyzed algorithms are presented in Table 1:

| Prediction model\Prediction time | $\tau=0.5$ s | $\tau=5$ s | $\tau=10$ s |
|----------------------------------|--------------|-------------|-------------|
| Kalman filter                    | 7%           | 36%         | >40%        |
| Autoregressive model             | 2%           | 13%         | >25%        |
| LSTM network                     | 7%           | 7%          | ~10%        |

**4. Conclusion**

Three algorithms for the ship motion prediction are compared: the Kalman filter, autoregressive algorithm, and artificial neural network with long-term short-term memory (LSTM). The performance of the algorithms is evaluated using simulation with real data. The ANN algorithm shows the best results in terms of prediction accuracy, but it is the most time consuming and computationally intensive. The gain in accuracy is due to the nonlinear and adaptive nature of this algorithm, which is actually implemented at the step of neural network training. Further it is planned to analyze the effectiveness of
identification and adaptation methods in the implementation of the Kalman filter described in [14, 15], with account for the work [4].

Acknowledgments
This work was financially supported by the Government of the Russian Federation (Grant 08-08) and by the Russian Foundation for Basic Research, project no. 18-08-01101A.

References
[1] Zhao X, Xu R and Kwan C 2004 Ship-motion prediction: algorithms and simulation results, Acoustics, Speech, and Signal Processing, 1988 International Conference on June
[2] Lukomskiy Y and Chugunov V 1988 Sistemy upravlenija morskimi podvizhnymi ob’ektami: Uchebnik (Marine mobile objects control systems: Manual) – L.: Sudostroenie (in Russian)
[3] Menahem M Sidar and Brian F Doolin 1983 On the Feasibility of Real-Time Prediction of Aircraft Carrier Motion at Sea, IEEE Transactions on automatic control
[4] Pelevin A 2016 Prognoz ugla naklona paluby korablja (Prediction of the ship deck tilt angle) Gyro and Nav. 4 122-132
[5] Wang Wei-chao, Qin Shi-qiao, Wu Wei and Zheng Jia-xing 2015 Prediction of ship pitch motion by dual autoregressive model 27th Chinese Control and Decision Conference (CCDC)
[6] Lin Zhuang, Yang Qiang, Guo Zhiqun and Li Jun 2011 An Improved Autoregressive Method with Kalman Filtering Theory for Vessel Motion Prediction International Journal of Intelligent Engineering and Systems
[7] Khan A, Bil Cees and Marion Kaye 2005 Ship motion prediction for launch and recovery of air vehicles 3 2795 - 2801
[8] Xianjun Liu, Wang Q, Huang Y, Song Q and Zhao L 2017 A prediction method for deck motion of aircraft carrier based on particle swarm optimization and kernel extreme learning machine Sensors and Materials 29 1291-1303
[9] Bo Zhou 2016 Continuous forecasting of ship sway based on LSSVM. 10.2991/icmmse-16.2016.16
[10] Yin Jianchuan, Perakis Anastassios and Wang Ning 2018 A real-time ship roll motion prediction using wavelet transform and variable RBF network Ocean Engineering 160
[11] Stepanov O 2017 Osnovy teorii otsenivaniya s prilozheniyami k zadacham obrabotki navigatsionnoi informatsii, tom 2: Vvedenie v teoriyu fil’tratsii (Fundamentals of Estimation Theory with Applications to Problems of Navigation Information Processing, vol. 2: Introduction to Filtering Theory) (St. Petersburg: GNTs RF TsNII “Elektropribor”) (in Russian)
[12] Sherstinsky A 2020 Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network (Physica D: Nonlinear Phenomena. Elsevier)
[13] Olah C Understanding LSTM Networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs
[14] Motorin A and Stepanov O 2015 IEEE Inter. Conf. on Multisens. Fusion and Integrat. for Intelleg. Sys. 2015 54–8
[15] Stepanov O and Motorin A 2019 Performance criteria for the identification of inertial sensor error models/Sensors. T. 19. 9
[16] Stepanov O and Motorin A 2020 Adaptivnye zadachi otsenivaniya i metody ih reshenija pri obrabotke navigatsionnoi informatsii. Trudy seminaru «Navigatsija i upravlenie dvizheniem», Samara
[17] Ruiyang Z, Konstantin N, Maria S and Andrey P 2020 Pitching Prediction Algorithm for Landing on Aircraft Carrier Deck (Saint Petersburg: 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), 1-3