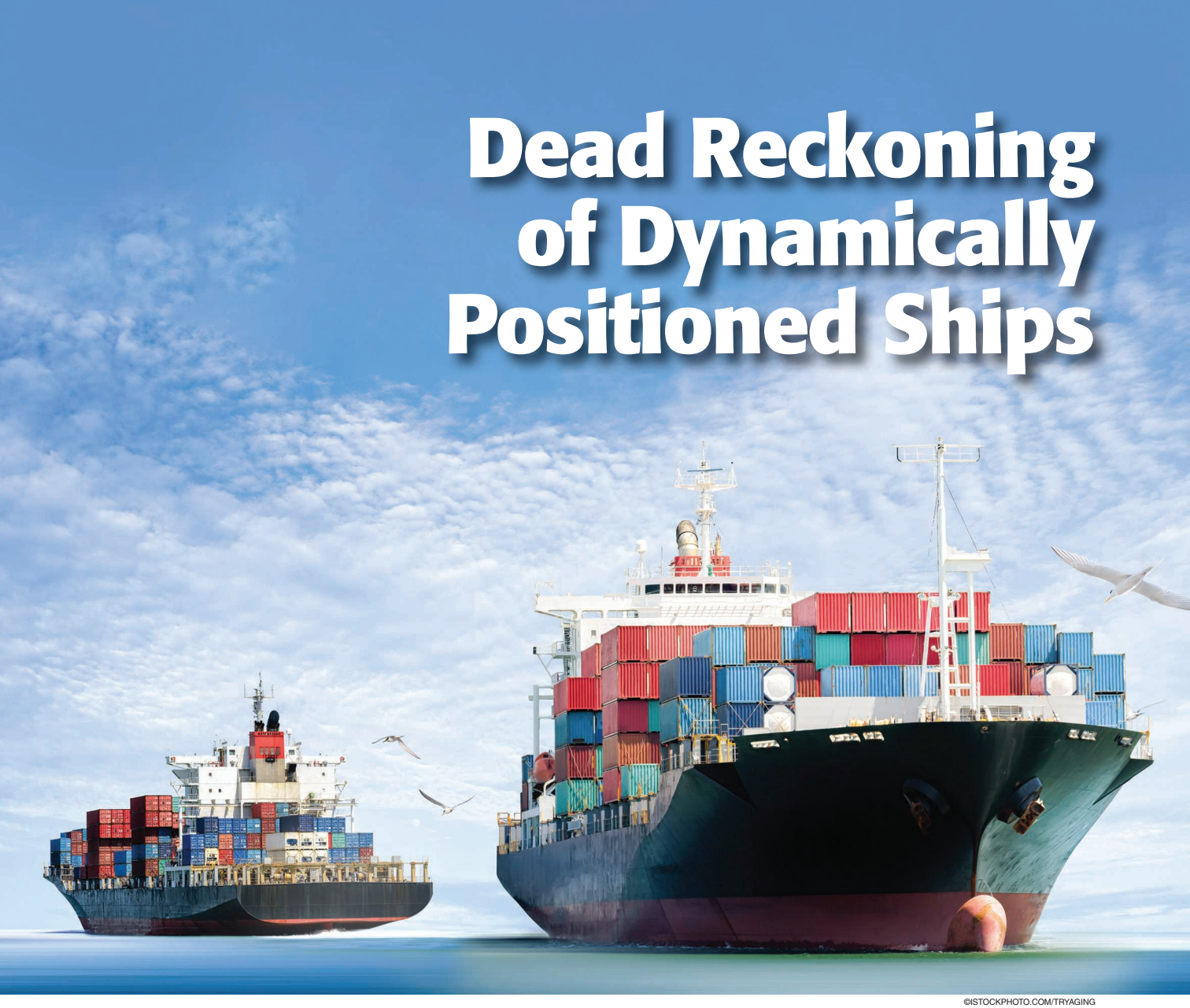
Dead Reckoning of Dynamically Positioned Ships



©ISTOCKPHOTO.COM/TRYAGING

Using an Efficient Recurrent Neural Network

By Robert Skulstad, Guoyuan Li, Thor I. Fossen, Bjørnar Vik, and Houxiang Zhang

When a ship experiences a loss of position reference systems, its navigation system typically enters a mode known as *dead reckoning* (*DR*) to maintain an estimate of its position. Commercial systems perform this task using a state estimator that includes mathematical model knowledge. Such a model is nontrivial to derive and needs tuning if the vessel's dynamic properties change. To this end, we propose using machine learning to estimate the horizontal velocity of the vessel without the help of position, velocity, or acceleration

Digital Object Identifier 10.1109/MRA.2019.2918125
Date of publication: 12 August 2019

sensors. A simulation study was conducted to demonstrate the ability to maintain position estimates during a Global Navigation Satellite System (GNSS) outage. Comparable performance is seen relative to the established Kalman-filter (KF) model-based approach.

Position Reference Failures

System failures that occur during the performance of operations at sea impose strict constraints on the ability of a ship to maintain position, which may have severe consequences. To mitigate the risk of failures, ships used in such operations have redundant systems. As marine operations grow increasingly autonomous and remotely operated [1], the importance of redundant systems to aid in controlling the vessel in case of failures increases. Failures that occur when a vessel is operating autonomously must be handled in a timely fashion by issuing a warning to a remote operator. When absolute position measurements are lost, a ship usually enters DR mode to provide estimates of the vessel position without the use of external signals. Various strategies exist to provide such estimates, but the general approach is to propagate the velocity and course of the vessel from a known position [2].

With regard to the position reference used by, for example, station-keeping motion controllers, various sensory platforms measuring the absolute or relative position may be applied, such as differential GNSS (dGNSS), radar, or hydroacoustic systems [3]. Although hydroacoustic inertial navigation systems (INSs) offer positioning solutions of quality similar to GNSS INSs, they rely on deployed seabed transponders [4]. Depending on the type of operation, this strategy might not be feasible. The most generic and widely available system is, therefore, dGNSS.

Because positioning systems ordinarily use signals from satellites to calculate position, there is a potential to experience a communication dropout between the remote operator and the autonomous vessel as well as a loss of GNSS position reference signals. In this case, the accuracy and long-term performance of the DR system become important for maintaining an accurate estimate of the vessel's current position.

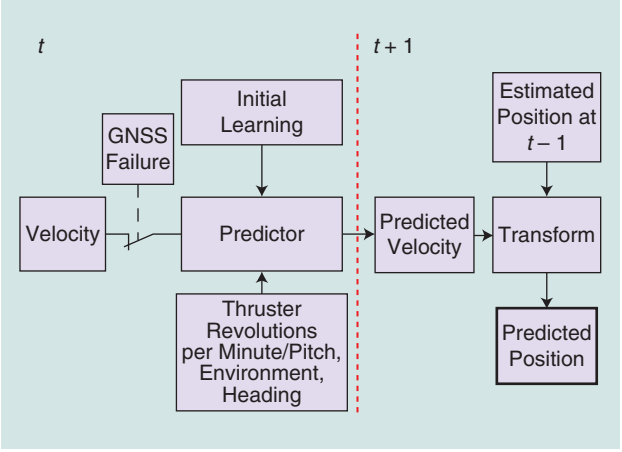


Figure 1. The proposed approach for performing DR using machine-learning methods.

In commercial navigation systems, the KF is often used to filter the wave-induced motion and provide estimates of vessel velocity [5], [6]. At a minimum, position and heading measurements are input to the estimator. These measurements are combined with the linearized vessel model to provide the state estimates. Wave filtering ensures that the oscillatory, wave-induced motion does not enter the controller of the vessel, which would cause increased fuel consumption and actuator wear [7]. The widespread use of KFs for state estimation and wave filtering makes them a natural choice as tools for providing DR position estimates as well.

Nonlinear observers that do not require knowledge of the vessel model have also been proposed for marine vessels operating in station-keeping mode, often referred to as *dynamic positioning* (*DP*). Bryne et al. performed wave filtering based on INS output [8], while Rogne et al. used inertial measurement unit (IMU) data for DR [9]. The IMU provides the raw angular velocity and specific force measurements; the INS integrates these measurements into a navigation solution in terms of position, velocity, and attitude.

In the event of a dGNSS position reference failure, the state estimator, assuming the KF is used, can make estimates based only on the vessel model [5]. The position reference failure may be caused by external factors, such as loss of a direct line of sight between satellite and receiver, intentional signal modification by a third party, or severe signal degradation due to noise incurred along the signal path [10]. Vessel model inaccuracies cause the position estimate to diverge from the real vessel position over time. If nonlinear observers and IMU data for DR are used, the integration of velocity and acceleration measurements that contain errors causes the estimated position to diverge from the true position. Typical IMU sensor error sources include bias, misalignment relative to the vessel frame axes, and temperature variation [11].

In this article, we present a data-based method for DR that involves modeling the horizontal velocity of the vessel in terms of inputs, such as thruster command/feedback values, thruster power consumption, measurable environmental states, and heading. A long short-term memory (LSTM) neural network was used for this purpose due to its ability to handle large time delays between input data and the resulting effect on the output data.

Because the heading of the vessel may be measured by an internal sensor system such as the compass, the availability of the heading measurement is independent of the GNSS system. Therefore, the change in position may be calculated from the heading angle, the predicted body-fixed velocity at the next time step, and the sampling time. Adding this to the position derived at the previous time step results in the estimated DR position at the next time step. Figure 1 shows a schematic view of the general units required. Under normal operation, the velocity unit provides target values for the supervised training. The target values have corresponding input vectors lagged by one

sample step and consist of variables related to the actuators of the vessel, wind conditions, and vessel heading. The initial learning block performs offline training based on the sampled targets and inputs. If a GNSS failure occurs, no targets are available, which precludes any further supervised training. At this point, the input vector is used to form one-step predictions of the vessel's longitudinal (surge) and lateral (sway) velocities. The proposed method has the advantage of not being dependent on a mathematical model of the vessel. Thus, it offers a more generic way of representing the velocity/position of a vessel due to force input by thrusters and other relevant and obtainable measurements.

In addition, automatic parameter adaptation can be performed based purely on sampled data. This may be relevant if, for example, the load distribution on the hull changes during operation. On the other hand, state estimators, such as the KF and nonlinear observers, allow for proof of stability and a more transparent input/output relationship. Figure 2 shows how the KF approach to DR may be performed. For both Figures 1 and 2, the vertical red line marks the demarkation between measured position signals and predicted position signals. A comparison in terms of position estimation performance was made between the two methods to gauge the feasibility of the LSTM model for DR.

Related Work

The DR mode is a position reference fallback system for marine surface vessels. Vessels operating beneath the ocean surface may apply DR positioning techniques as the primary system for determining position [12]. German et al. compared two methods of determining position for an autonomous underwater vehicle [13]. Internal sensors included a three-axis magnetic compass, a Doppler velocity log, and a depth sensor. The first method relied on an extended KF fusing global positioning system (GPS) data—transferred acoustically from an autonomous tender vessel—with the onboard sampled data. The second used only the internal sensors, which produced DR position solutions.

For DR of ocean surface vessels, Diamant and Jin used a three-axis accelerometer to provide the DR heading and position of a vessel [14]. They used machine learning to classify accelerometer data into bins of similar pitch angle and then projected these onto the local north–east horizontal plane. The projected accelerations were integrated to yield the estimated position and heading. The authors used only a three-axis accelerometer as sensor input for DR to avoid using measurements from a gyrocompass, which, according to them, may be unavailable or contain too much noise to be of use in estimating the attitude of the vessel.

Rogne et al. investigated the DR capabilities of an INS aided by dGNSS signals [9]. They applied two different low-cost IMUs, providing accelerometer, compass, and angular velocity measurements. Two different nonlinear observers were compared, using no information about the vessel model, on a test set sampled on a vessel performing a DP operation in the North Sea. The authors found that the top performer had a position error of about 100 m after 10 min of dGNSS outage.

DR has been used in other domains as well, such as for automobiles and in aerospace applications. When comparing seagoing vessels with airplanes, it is clear that there is a large difference in dynamic properties and the severity of wind impact on the frame of the respective objects. This is especially true for unmanned aerial vehicles (UAVs) due to their small size. Mokhtarzadeh and Gebre-Egziabher performed a study of cooperative navigation for UAVs [15]. Several UAVs connected in a network shared navigational information during a 5-min GPS outage to reduce the position error drift rate of a DR-based navigation filter. The authors opted to use an integration of airspeed measurements instead of the more traditional INS sensors to avoid the double integration necessary to determine position from the acceleration estimated by the INS. An additional advantage to this approach is the separation of the DR operation from the attitude and heading reference system. Instead of using an airspeed sensor, Fusini et al. used a downward-looking camera and a machine-vision system to provide the velocity of the UAV [16]. The acquired velocity was input to both a nonlinear observer and an exogenous KF for performing DR, and a bounded error rate was achieved during experimental real-system testing.

In the event of a dGNSS position reference failure, the state estimator, assuming the KF is used, can make estimates based only on the vessel model.

Land vehicles usually follow predefined tracks, often in areas that are not conducive to robust GNSS signal reception. To produce continuous in-car navigation services, DR/INS systems, digital maps, and mathematical models of

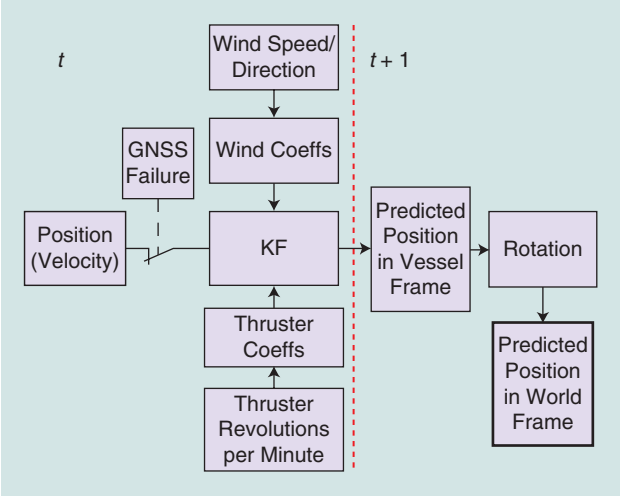


Figure 2. The approach for performing DR using the KF. coeffs: coefficients.

the vehicle typically complement the GNSS measurements. Skog and Händel provide an overview of such systems and the methods used for fusing both external (e.g., GNSS) data and internal sensor data (e.g., odometer, gyroscopes, and accelerometers) [17].

Abbott and Powell provided a study of the error contribution of various sensors for an in-car navigation system [18]. They applied sensitivity analysis to gauge the performance of a KF-sensor fusion algorithm against a reference system. Their findings suggest that the use of differential GPS (dGPS) offered improved calibration of the internal sensors, resulting in significant reduction of error drift during a satellite system outage. Therefore, relatively inexpensive internal sensors combined with dGPS could provide sufficiently accurate DR systems. Extending the flexibility of the KF for combining data from several sensors at various sampling rates, Barrios et al. introduced a dynamic state noise covariance matrix [19]. The purpose of this dynamic matrix is to reflect the state uncertainty more accurately when sensors drop out for any length of time.

Like Rogne et al. [9], Ahmed and Tahir [20] recognize that high-performance IMU units contribute significantly to overall system cost. That motivated the use of a low-cost micro electromechanical system IMU unit containing a triaxial gyroscope and an accelerometer to accurately determine the attitude of a car. The authors estimated the vehicle acceleration using the kinematic vehicle model and the known norm of the gravity. In addition to providing accurate attitude estimates, the ability to separate the gravity-induced acceleration components from the overall acceleration measurement proved beneficial to DR performance.

Table 1. The parameters used for the additive noise elements of the position and heading measurements.

	GM σ	T_c	White Noise σ	Bias μ
Position	0.1 m	240 s	0.2 m	[–0.2, 0.2] m
Heading	0.1°	60 s	0.1°	[–0.1, 0.1]°

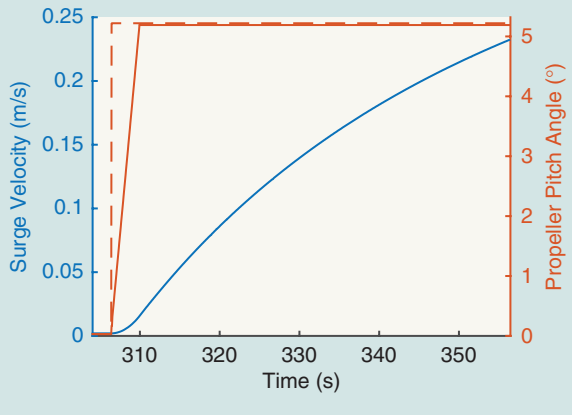


Figure 3. The delayed response of the two variables, surge velocity (solid blue line) and actual thruster pitch angle (solid red line), as a reaction to a step increase in the commanded thruster pitch angle (dashed red line).

Methodology

In this section, we introduce the measured signals, delays present in the actuators of the simulated vessel, and LSTM network model. Methods of limiting the input data dimension and selection of LSTM hyperparameters are also considered.

Measurement Noise

Noise was added to the following measured states:

- *position*: the position measurements given in the north-east-down (NED) frame
- *heading*: rotation about the z-axis of the vessel
- *velocity*: the linear velocity given in the NED frame.

Therefore, the position and heading measurements, as seen by consumers of the sensor data, are a sum of the true value sampled from the simulator, white noise, a bias, and a Gauss-Markov (GM) process. The discretized GM process is

$$x[k+1] = \exp\left(-\frac{\Delta t}{T_c}\right)x[k] + \sigma w[k], \tag{1}$$

where k is the discrete time variable, Δt is the sampling interval, T_c is the correlation time, and w is the Gaussian white noise with a standard deviation of σ. The addition of noise terms to form the expression for the position and heading with noise [21] is as follows:

$$\boldsymbol{p}[k] = \boldsymbol{p}_{\text{true}}[k] + \boldsymbol{x}_p[k] + \boldsymbol{\sigma}_p \boldsymbol{w}_1[k] + \boldsymbol{\mu}_p, \tag{2}$$

where $\boldsymbol{p}$ is a two-dimensional column vector containing the north and east position with additive noise, $\boldsymbol{p}_{\text{true}}$ is the noiseless north–east position, $\boldsymbol{x}_p$ holds the corresponding GM processes for the two components, $\boldsymbol{\sigma}_p$ is a diagonal matrix containing standard deviations of added white noise ($\boldsymbol{w}_1$), and $\boldsymbol{\mu}_p$ holds the position bias:

$$\psi[k] = \psi_{\text{true}}[k] + x_\psi[k] + \sigma_\psi w_2[k] + \mu_\psi. \tag{3}$$

Noise added to the heading signal is described in (3), where ψ is the heading angle containing noise, ψ_{true} is the noiseless heading angle, x_ψ is the GM process related to the heading angle, σ_ψ is the standard deviation of the Gaussian white noise w_2, and μ_ψ is the heading angle bias. Table 1 shows the parameters used in simulating the position and heading states with noise. The angular/linear velocity received only a constant bias and white noise [22].

Time Delay

Delays in time between a change in thruster command (input) and the given response in velocity (output) are present in the sampled time series. They are caused by both the linear/rotational inertia of the vessel and the rotational inertia of the various thruster systems. Figure 3

shows the surge velocity response due to a step increase of 5° in the commanded pitch angle of the two main thrusters. At a pitch angle of 5°, the thrusters output about 5% of the maximum thrust force. We see that the rate of change of the thruster itself is limited to 1.4°/s, such that it takes approximately 3.5 s to reach 5°. Furthermore, the time spent to reach a surge velocity of 63% of the steady-state value of 0.37 m/s is 50 s.

During normal DP operation, there will be no step function inputs because the controller reaches a relatively fixed command vector to compensate for the external disturbances. However, perturbations in thruster commands occur due to imperfect wave filtering, causing set-point changes in the range of –0.5 to 0.5°. To ensure that the input vector to the machine-learning algorithms contains information of the most significant transient effects due to changes in thruster commands, we include 10 s of history data for each input variable. Similar delays are seen for the tunnel thrusters. Delays also exist between the vessel velocity and changes in wind velocity and direction.

Position Estimation Concept

Two networks predict the horizontal velocity components of the vessel: one predicts the surge velocity and one the sway velocity. This makes it possible to provide a custom network in terms of input pattern for each of the velocities expressed relative to the horizontal axes of the vessel frame of reference. After an initial network learning phase, the proposed approach does not rely on samples produced by a GNSS. Inputs to the networks are therefore available up to and including the discrete step k. This enables a prediction of the velocities at the subsequent time step, $k+1$. To get from a predicted velocity to a predicted traveled distance in the NED frame, the predicted velocity is multiplied by the sampling time and rotated according to the heading angle. At this point, the traveled distance due to the predicted velocity, Δp in Figure 4, is added to the previously estimated position. The propagation of position is given by

$$\hat{\boldsymbol{p}}[k+1] = \hat{\boldsymbol{p}}[k] + \boldsymbol{R}(\psi)\,\hat{\boldsymbol{\nu}}[k+1]\,\Delta t, \tag{4}$$

where $\hat{\boldsymbol{p}}$ is the estimated north–east position of the vessel in the NED frame, $\hat{\boldsymbol{\nu}}$ is the predicted velocity vector relative to the vessel-frame coordinate system, and $\boldsymbol{R}(\psi)$ is the square rotation matrix that transforms the predicted velocities to NED-frame velocities. $\hat{\boldsymbol{\nu}}$ contains the surge and sway velocity of the vessel [variables ($\hat{\nu}_{\text{lon}}$, $\hat{\nu}_{\text{lat}}$) of Figure 4]. k is the discrete step index with a step interval of $\Delta t = t[k] - t[k-1]$. A visual representation of the process is given in Figure 4.

At time $t[k]$, the horizontal position is measured using the signal received from GNSS satellites. At the next time step, $t[k+1]$, the receiver on the vessel fails to produce the position of the vessel via GNSS signals due to one of the aforementioned reasons for GNSS unavailability. At this point, the DR algorithm is activated and provides an estimate of the vessel position through the prediction of the surge ($\hat{\boldsymbol{\nu}}_{\text{lon}}$) and sway ($\hat{\boldsymbol{\nu}}_{\text{lat}}$) velocities seen in Figure 4. Together, they make up the velocity vector $\hat{\boldsymbol{\nu}}[k+1]$ of (4). By design, the method proposed in this article, the LSTM recurrent neural network, receives only input variables that contain information about external disturbances, the heading angle, and the control intention of the vessel. Measurable external disturbances include the wind velocity and wind direction for the system used in this article. Although systems exist for measuring and estimating wave spectrum parameters in the vicinity of the vessel [23], [24] as well as for measuring the velocity and direction of the ocean current affecting the hull [25], we limit the environmental sensory equipment to sensors that are currently available in the system. A key assumption at this stage is that the velocities relative to the vessel frame are available without bias. If the velocity targets used for training the machine-learning methods contain biases, the error rates during DR increase significantly.

LSTM

An LSTM network was used to model how the velocity of the vessel relates to the inputs. LSTM networks differ from feedforward networks in that they have weight connections between all nodes that are not input nodes [26]. To avoid the problem of vanishing/exploding gradients for back-propagation-through-time learning, Hochreiter and Schmidhuber devised a unit called a *memory cell* [27], which contains a constant error carousel (CEC) unit that aims to keep the error flow constant through a unity self-connection. A linear activation is used in the CEC. In addition to the CEC, the memory cell contains two multiplicative gate units that control the access of input signals and output signals to the CEC. Because LSTM networks are particularly well suited for learning the relationship between events separated by a long time delay, we include this network in our analysis. Due to the large inertia

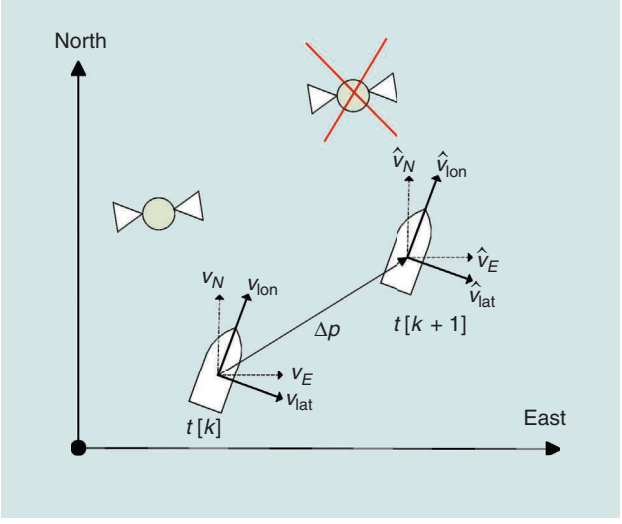


Figure 4. A switch from normal operation [$t(k)$] to loss of the GNSS, requiring a DR system to estimate the position at the next step without an absolute position measurement.

values of both the vessel and the various actuator systems, there may be delays between such events as inputting a command and a significant position change. See the "Time Delay" section for more information about time lag. Functions in the MATLAB neural network toolbox were used for training and prediction with the LSTM network.

As sensors output measurements of various physical quantities, they operate in different value ranges. To have each measured variable contribute equally as part of the input vector, all data should be normalized. To scale both the variation and the absolute value of each variable in the data set, we use the mean/standard deviation approach to normalization according to

$$\boldsymbol{x}' = (\boldsymbol{x} - \bar{\boldsymbol{x}})/\mathrm{std}(\boldsymbol{x}), \tag{5}$$

where $\boldsymbol{x}$ is the N-sample by M-variable training data set, $\bar{\boldsymbol{x}}$ is the mean value of each variable, std($\boldsymbol{x}$) represents the standard deviation of the variables, and $\boldsymbol{x}'$ is the normalized data. All of the signals used in this article have a bounded range, meaning that, given a representative set of training data, the range of the test data does not differ significantly.

Input Selection

By limiting the number of input variables to those that hold a certain level of information about the output states, the network's ability to generalize increases, and its complexity is reduced. Mutual information (MI) is applied in this article to facilitate the dimension reduction of the input vectors used by the machine-learning models. This operation is known as *input selection* and is performed prior to generating or updating the actual predictive network. MI provides a measure of the reduction of uncertainty about a variable x given a variable y [28]. It is defined by

Table 2. The normalized average MI value of input variables relative to output variables.

Input Variable	Description	Surge Velocity	Sway Velocity
1	Heading angle	0.0	0.08
2	Wind angle	1.00	0.69
3	Wind velocity	0.67	0.69
4	Bow thruster power	–	1.00
5	Bow thruster cmd	–	0.02
6	Bow thruster act	–	0.04
7	Stern thruster power	–	0.73
8	Stern thruster cmd	–	0.00
9	Stern thruster act	–	0.01
10	Main thruster power	0.49	–
11	Main thruster cmd	0.37	–
12	Main thruster act	0.44	–

act: feedback; cmd: command.

$$I[\boldsymbol{x}, \boldsymbol{y}] = -\iint p(\boldsymbol{x}, \boldsymbol{y}) \ln\left(\frac{p(\boldsymbol{x})p(\boldsymbol{y})}{p(\boldsymbol{x}, \boldsymbol{y})}\right) d\boldsymbol{x}d\boldsymbol{y}, \tag{6}$$

where $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$ are the distributions of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, and $p(\boldsymbol{x}, \boldsymbol{y})$ is the joint distribution between the two sets. Thus, if the evaluation of $I[\boldsymbol{x}, \boldsymbol{y}_1]$ results in a larger numerical value compared to the evaluation of $I[\boldsymbol{x}, \boldsymbol{y}_2]$, variable $\boldsymbol{y}_1$ contains more information than variable $\boldsymbol{y}_2$ about variable $\boldsymbol{x}$. Estimators are employed for practical implementations of MI, and its use within the domain of time-series regression is documented in [29] and [30]. In this article, we calculate MI using the MATLAB functions presented in [31].

Input Structure

The vessel has six thrusters: two bow tunnel thrusters, two stern tunnel thrusters, and two main thrusters with rudders. In this article, the vessel performs station keeping using one proportional integral derivative (PID) regulator per degree of freedom (DoF), which precedes a basic thrust-allocation unit that applies the unconstrained generalized inverse method for distributing motion-controller force requests. To simplify the allocation problem, the rudder angles of the two main thrusters were fixed. A further simplification was performed to decouple the effect of the main thrusters on the rotation of the vessel. For all simulations in this article, the main thrusters are operated in unison, such that they affect only the motion of the vessel along its longitudinal axis.

By intuition, we select inputs to represent the velocity of the vessel in its forward and sideways axes individually. The forward/surge speed varies depending on the inertia, thruster force, and environmental force applied along that axis. Thus, measurements of the main thrusters (fixed along the forward axis) are included, along with the wind direction and velocity and heading angle. Lacking a mathematical model of the effect of the thruster commands and wind magnitude and direction, we aim to derive this from the measurements.

We take a similar approach in selecting the input variables for the velocity in the sway direction, selecting measurements from both a forward- and a stern-mounted thruster as well as the heading and wind measurements. The partitioning of the variables in an input pattern is shown by

$$\begin{aligned} \boldsymbol{z}_k = [&x_1[k] + x_1[k-d] + \ \ldots + x_1[k-(n-1)\,d], \ldots \\ &x_2[k] + x_2[k-d] + \ldots \ + x_2[k-(n-1)\,d], \ldots \\ &x_m[k] + x_m[k-d] + \ \ldots + x_m[k-(n-1)\,d]], \end{aligned} \tag{7}$$

where $\boldsymbol{z}$ marks the total 1D input pattern, k is the discrete sample step, x is the measured input variable, d is the delay in number of steps, n is the number of delayed samples of a variable to include, and m indicates the type of input variable. See the first column of Table 2 for a list of input variables used in the two separate input patterns, which correspond to the variable m.

Optimizing Network Structure

Depending on parameters such as neuron number, layer depth, size of training data set, and so on, the evaluation of a single instantiated neural network may be quite costly in terms of computation time. The approach of Snoek et al., termed *Bayesian optimization*, provides efficient hyperparameter optimization, thereby lowering the overall cost of producing an efficient model configuration [32]. In this article, the optimization of the LSTM network (see the "LSTM" section) was focused on the number of LSTM blocks in a single layer as well as the learning rate, the two most important parameters according to [33]. A range of 10 to 200 was selected for the number of blocks, and a range of 10^{-5} to 10^{-1} was selected for the learning rate. The cost function returned the mean square error (MSE) of the validation samples (10% of the total number of samples used for training), which provided a means of quantifying the expected performance of the network. Together with the input selection stage, the number of parameters in need of tuning has now been limited to those required for setting the threshold for the input selection and the upper/lower values of the range in which hyperparameter optimization is performed.

Simulation Results

We propose using a data-based model, described in the "LSTM" section, to model the relationship between various inputs and the predicted linear vessel frame-relative velocities at the next time step. To assess the performance and the validity of this method, we compare it to two other models:

- *KF*: Linearized equations of motion are obtained for the vessel by rotating the position measurements to a vessel-parallel coordinate system at each time step. This facilitates the use of a linear KF observer model for the DP test case described in this article [34].
- *Single-layer feedforward neural network (SLFN)*: This represents the most basic structure among neural networks used for regression.

In the case of the KF, we coast through the outage using the thruster command, wind velocity, and wind angle as inputs. These measurements are fed to the mathematical model of the vessel. The individually learned predictive models of the two machine-learning methods replace the explicit vessel model. The machine-learning DR methods do not use the vessel model or sensors to measure the displacement of the vessel.

Vessel and Environment Description

All experiments were conducted in a commercial simulator, developed by the Norwegian company Offshore Simulator Centre AS. It features a simulated environment in which a user may manipulate the wind, waves, and ocean current to mimic real-life conditions and offers a library of virtual vessels to choose from. For these experiments, a multipurpose offshore vessel was selected. Table 3 provides its main dimensions, and Figure 5 shows a view of the simulated environment with the selected vessel engaged in a DP operation close to a static rig. For the specific simulation study performed in this article, varying environmental parameters were applied. The direction of the environmental disturbances is incremented at intervals of 30° from 0° to 360°, relative to the vessel frame. At each fixed direction, a set of wind and wave magnitudes was applied consecutively, causing increasingly severe weather conditions. Table 4 shows the wind and wave magnitudes for each of the distinct conditions faced by the vessel at the directions previously specified. A specific weather condition is determined by the direction of the wind and waves along with their respective magnitudes. In this test set, each weather condition has a duration of 14 min; the first 7 min involve a change of both wind and wave magnitude from the previous weather condition. If all conditions have been run for a single direction, this period involves a linear transition from one weather direction to the next one. The entire simulation test set spans approximately 15 h of vessel maneuvering. The actual run time is reduced by running the simulation five times faster than the real time.

A 3-DoF DP controller is applied to perform station keeping. The controller applies a single PID controller in each DoF, and the output of the motion controller connects to a basic generalized inverse-control allocator for distribution of the generalized force vector into individual thruster commands. Figure 6 shows how the true position compares to the position with measurement noise added (see the "Measurement Noise" section). The latter is the raw position output by the dGNSS system when it is operating normally. The noiseless position signal is not used for any purpose other than visualization.

KF Parameters

A KF was implemented for comparison to a conventional DR method. It requires model-dependent matrices in addition to tuning parameters. We list the applied tuning parameters along with the matrices describing the mass and damping of the simulated vessel in the following:

Table 3. The dimensions of the simulated vessel.

Description	Value
Length between perpendiculars (L_{pp})	82.7 m
Breadth	23.1 m
Displacement	10,180 × 10^3 kg

Table 4. The parameters of the sea states simulated at each discrete weather direction.

Significant Wave Height (Hs)	Wind Velocity
1 m	2 m/s
2 m	4 m/s
3 m	7 m/s
4 m	11 m/s

$$M = \begin{bmatrix} 1.02e7 & 0 & 0 \\ 0 & 1.02e7 & 8.44e6 \\ 0 & 8.44e6 & 5.80e9 \end{bmatrix} \tag{8}$$

and

$$D = \begin{bmatrix} 300000 & 0 & 0 \\ 0 & 550000 & 600000 \\ 0 & 600000 & 1.38e8 \end{bmatrix}. \tag{9}$$

Furthermore, the two tunable matrices of the KF, the $\boldsymbol{R}$ and $\boldsymbol{Q}$ matrices, had the following numerical values. The values in $\boldsymbol{R}$ were determined using a data set sampled while the vessel was unaffected by environmental disturbances, while the general rules given in [6] were used for tuning the $\boldsymbol{Q}$ matrix:

$$\begin{aligned} \boldsymbol{R} &= \text{diag}([0.7, 0.7, 0.2]) \\ \boldsymbol{q}_1 &= \text{diag}([0.1, 0.1, 0.1]) \\ \boldsymbol{q}_2 &= \text{diag}([1e6, 1e6, 1e6]) \\ \boldsymbol{q}_3 &= 0.1 \times \boldsymbol{R} \end{aligned} \tag{10}$$

and

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{q}_2 & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{q}_3 \end{bmatrix}. \tag{11}$$

In terms of objective, the implementation of the KF used in this article differs from the other methods. The KF aims to produce a position estimate that reflects no influence caused by zero-mean oscillatory wave forces. Thus, under normal operation, we would expect to see a smooth trajectory following the mean of the measured position and heading. The other two methods aim to copy the exact behavior of each time series, resulting in a more erratic trajectory during normal operation due to both measurement noise and wave-induced motion.

Case Study 1: Input Selection

Reducing the input dimension of the network has positive effects on computation time as well as network interpretability and generalization ability. It is key to retain sections of the overall input pattern that contain useful information, which sets the stage for the MI method described in the "Input Selection" section. MI allows for a ranking of input variable importance relative to an output variable. Therefore, input variables that offer a low relative MI value were deselected at this stage. Table 2 shows the 0–1-normalized MI of the two target variables, surge velocity and sway velocity.

As shown in the "Description" column of Table 2, *power* indicates the consumed power in watts of the specific thruster, *cmd* indicates the command sent to the thruster (either a blade pitch angle or an angular velocity value), and *act* indicates the feedback value measured at the thruster. Given the results in Table 2 and a threshold value of 0.4, we see that the reduced input pattern of the network predicting the surge velocity consists of input variables 2, 3, 10, and 12.

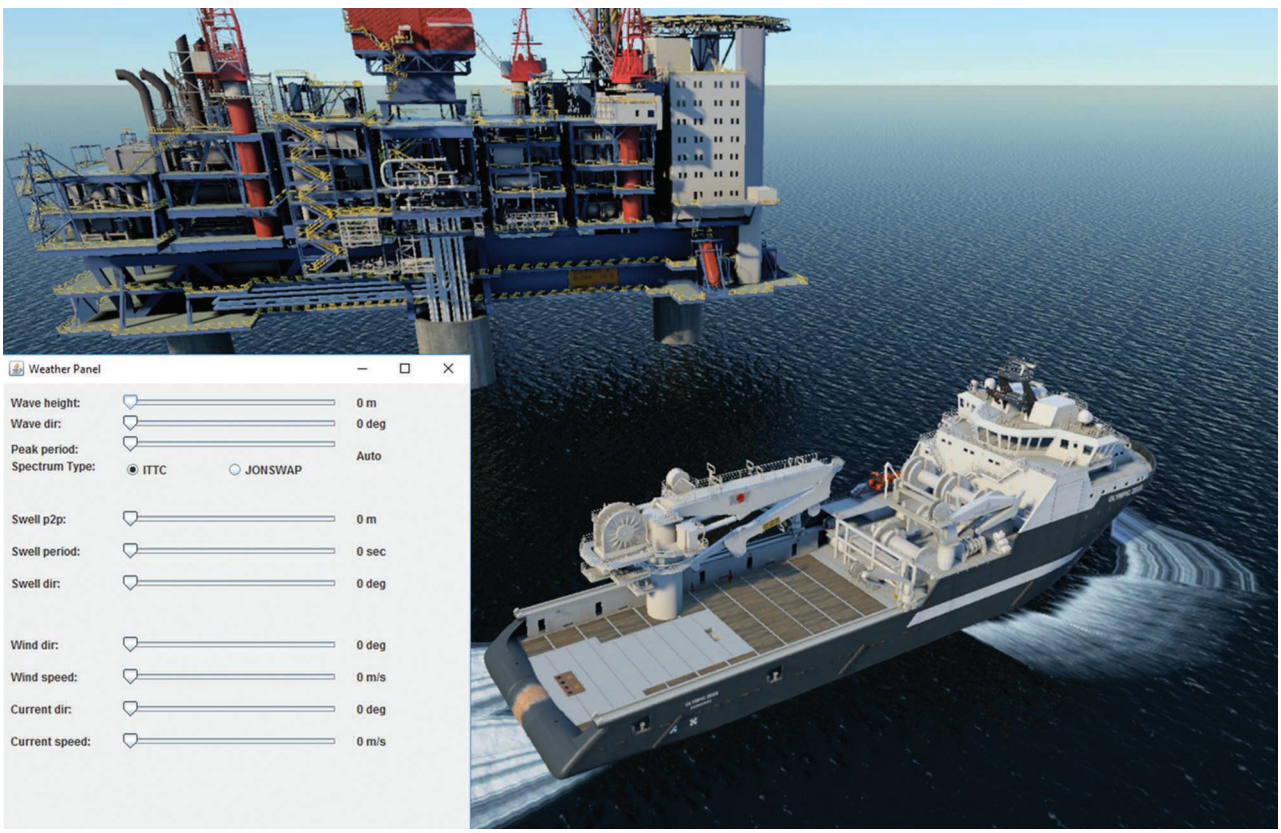


Figure 5. A screenshot showing the simulated environment and the panel for applying environmental disturbances. (Source: Offshore Simulator Centre; used with permission.)

For the network predicting the sway velocity, the variables are 2, 3, 4, and 7. The input patterns are thereby reduced to 66% (surge velocity) and 44% (sway velocity) of the original input length. The data set used for training contains 10^4 samples spaced by 1 s. Over the course of about 2.5 h of simulation time, 12 randomly chosen weather conditions were run. Wave heights and wind velocities were chosen within the ranges given in Table 4.

A comparison of the performance in terms of estimated position, relative to the sampled true position, is shown in Figure 7, which displays the mean error with and without MI over a 1-min DR period for all weather conditions in the test set within which the vessel was able to keep the desired position. The deselected weather conditions are highlighted in case study 2. As noted in the "Vessel and Environment Description" section, each individual weather condition lasts for 14 min; 1 min toward the end of each weather condition was applied for the DR tests. Using the complete input vector for both the surge velocity and sway velocity estimators results in an increase in position error. Figures 8 and 9 show

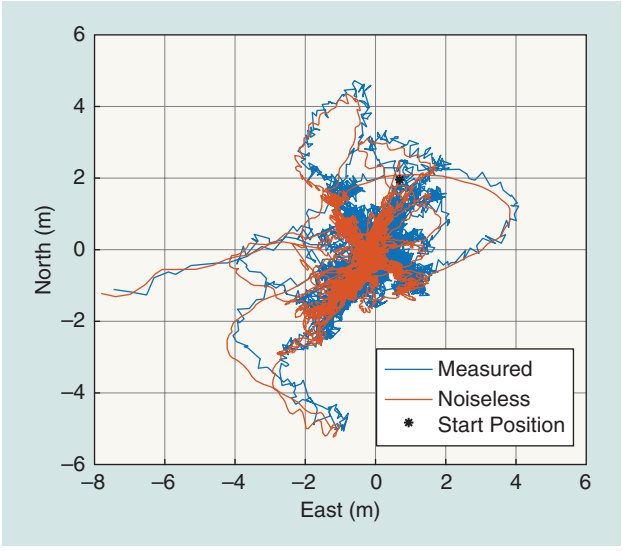


Figure 6. A visual representation of the vessel position for a part of case study 2. The noiseless position measurement (red line) is included only to provide a reference to the raw dGNSS position output (blue line).

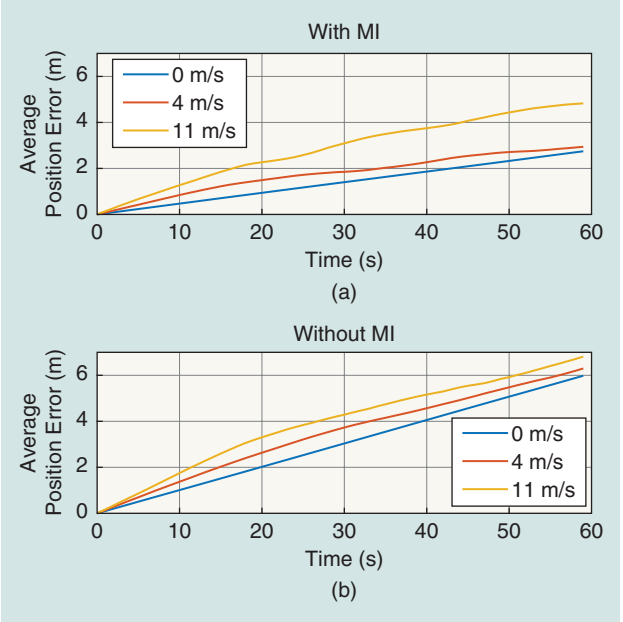


Figure 7. The mean DR error of the LSTM method for all wind directions at wind velocities of 0, 4, and 11 m/s. (a) A subset of the elements in the complete input vector was extracted using MI and used as input to the LSTM model. (b) All entries in the complete input vector were used as input to the LSTM model.

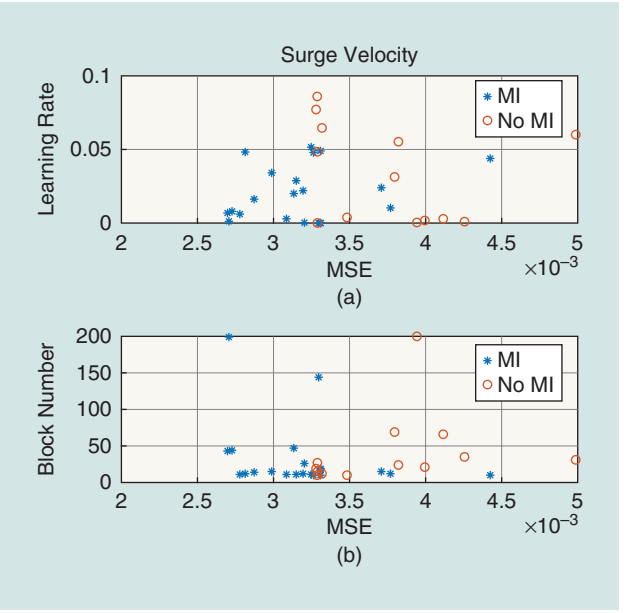


Figure 8. The results of running the LSTM hyperparameter optimization function on the two parameters, (a) learning rate and (b) block number, for the surge velocity estimation model.

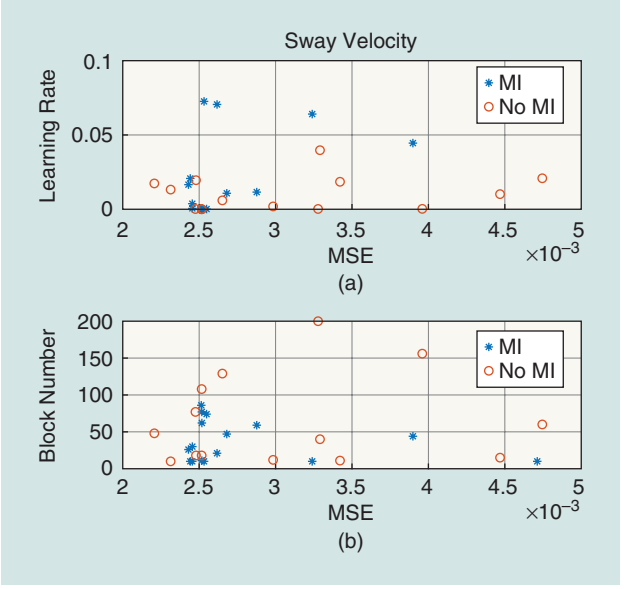


Figure 9. The results of running the LSTM hyperparameter optimization function on the two parameters, (a) learning rate and (b) block number, for the sway velocity estimation model.

the result of running the optimization function to determine optimized hyperparameters for the LSTM estimators. For the reduced-input estimators of surge velocity and sway velocity, the following hyperparameter pairs were selected based on the lowest observed MSE value:

- surge velocity: block number = 43, learning rate = 0.0070
- sway velocity: block number = 26, learning rate = 0.0165.

Case Study 2: Impact of the Environmental Variables

In this section, we look at how the LSTM, SLFN, and KF perform over a wide operational range. As described in the previous section, input selection is applied, resulting in the use of variables (2, 3, 10, 12) to predict future surge velocity and variables (2, 3, 4, 7) to predict future sway velocity (see Table 2). Figure 6 shows the position of the vessel throughout the test set, with and without measurement noise. Similar to the previous section, we use the position without measurement noise as reference. To evaluate the models, we view the mean distance error observed during a 1-min period of each weather condition. Each period of evaluation that requires DR starts 3 min after the transition into the new weather condition has finished. This allows the control algorithm time to adapt to the current environmental forces. Figure 10(a) shows how the distance error propagates, without any GNSS input, for the LSTM approach. The distance

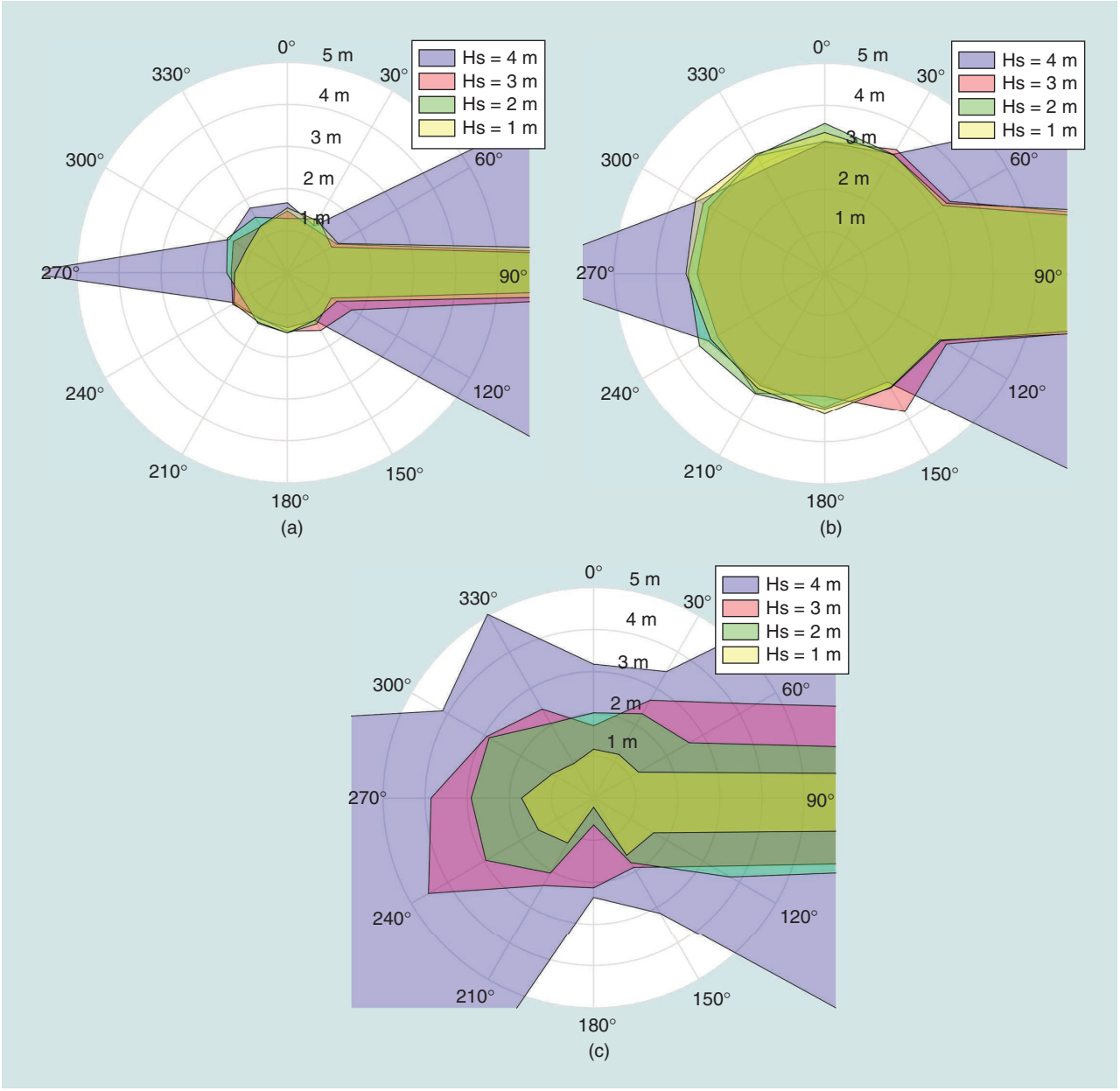


Figure 10. The results of mean position estimation error given in the horizontal plane for the (a) LSTM, (b) SLFN, and (c) KF model in case study 2. Each data point shows the mean position estimation error during a 1-min DR period. Hs denotes the significant wave height in meters.

from the origin of the figure to each discrete weather direction is determined by (12), which gives the mean position estimation error:

$$\bar{e}_{\text{dist}}[k] = \frac{1}{N}\sum_{k=1}^{N}\sqrt{(\hat{p}_n[k]-p_n[k])^2+(\hat{p}_e[k]-p_e[k])^2}. \tag{12}$$

In (12), (p_n, p_e) is the measured horizontal plane position given in the NED frame, $(\hat{p}_n, \hat{p}_e)$ is the corresponding DR position, and k signifies the discrete step. Figure 10(b) and (c) shows the results of running the same test with the remaining two methods, that is, the SLFN and the KF.

Considering the area covered by the polygons in Figure 10(a), the mean error is roughly similar, irrespective of weather direction and wave heights of fewer than 3 m. Similar properties are seen for the SLFN method [see Figure 10(b)]. The optimization scheme used for the LSTM method was also applied for the SLFN method. This yielded an optimized hidden neuron number of 93 for the sway velocity estimator and 55 for the surge velocity estimator. For both estimators, the optimization procedure favored a sigmoid activation function. An increased DR error may be seen for both methods at weather directions of 270°, 120°, 90°, and 60°. At these directions and a wave height of 4 m, the tunnel thrusters are unable to produce sufficient thrust to withstand the environmental forces acting on the vessel. This caused saturation of thruster commands and divergence from the desired position. When a given set of thruster commands no longer causes vessel motion similar to that experienced in the training set (e.g., when the environmental forces outweigh the control forces and cause thruster saturation), the output of the estimators diverges from the true vessel velocity. The most severe effects of the saturation are seen at a direction of 120° and a 4-m wave height. The vessel is unable to recover the desired position in a timely fashion, causing further estimation error for all simulated conditions at the subsequent weather direction of 90°.

Discussion

The input variables related to thruster command, thruster operating point, and power do not directly give information about the motion of the vessel. However, they indirectly contain information about how the vessel moves. A thruster command, executed over a given time interval, induces forces on the vessel, causing a change in linear/angular speed. The consumed power fluctuates due to both the thruster command and the velocity of the vessel relative to the surrounding water. Accounting for lags (see the "Time Delay" section), one may obtain knowledge of how the vessel moves by viewing thruster data. This is one of the advantages of using a data-based model: it learns such connections. To make the task of the machine-learning methods easier and more effective, input selection picks the most relevant input variables for our problem and also mitigates the curse of dimensionality, which is an issue for high-dimensional input patterns in regression problems [35]. The number of samples necessary to approximate a function to a certain degree of smoothness grows exponentially with the input dimension.

In this study, we performed input selection on the basis of the mean MI (see the "Input Selection" section) for an input variable containing lags according to, for example, variable x_1 of (7). This allows for an uninterrupted representation of the selected variable. Another strategy would be to select the entries of the total input pattern (see the "Input Structure" section) that have an MI value greater than some threshold, which does not leave the intervariable spacing intact but ensures that all entries in the selected pattern have a given MI content relative to the target variable.

The results produced in case study 1 show the increased performance gained by selecting input variables that provide a certain amount of information about the output variable, omitting the remainder of the original input variables. Viewing the optimization results in Figures 8 and 9, we see that only the surge velocity estimator benefits from applying MI, at least in terms of the MSE derived from a validation set consisting of 10% of the samples in the training data set. This amounts to approximately 1,000 samples. Although the sway velocity estimator displays a slight decrease in performance when applying the reduced input pattern, the overall effect of MI is positive. As the input selection processes of the two estimators are separate, one may choose to implement one or both of the reduced input patterns to maximize the expected DR performance. Figure 7 shows how the estimated position, using input vectors selected by MI, diverges more slowly than when the original input pattern is applied during a GNSS dropout. As MI was shown to aid the LSTM model (see the "LSTM" section) in terms of reducing the position estimation error, it was applied to both machine-learning models for the second case study, shown in the "Case Study 2: Impact of the Environmental Variables" section. Of the two, the LSTM performed better, with a mean distance error of fewer than 2 m for wave heights below 3 m. The measurements of thruster-related states (power consumption, set point, and feedback) were assumed to be noise free.

Reducing the input dimension of the network has positive effects on computation time as well as network interpretability and generalization ability.

The KF, described in the "KF Parameters" section, has similar performance relative to the LSTM for wave heights of 1 m. When wave heights of 2, 3, and 4 m affect the vessel, the LSTM provides consistent DR position estimates, whereas the KF error increases. The KF error increase is due, in part, to the linear relationship between a thruster command and the

resulting force output of a thruster assumed in a regular KF. Due to the lack of measurements to facilitate a corrector function, the DR position is driven solely by the vessel model and the thruster-command input vector. Similar to the LSTM model, the SLFN model displays consistent DR position estimates, although at a larger magnitude. When wave heights exceed 3 m, the SLFN outperforms the KF. Because the implemented KF requires a significant number of parameters to be set, an optimization scheme to derive optimized KF parameters might offer a more balanced comparison among the three methods.

When wave heights of 2, 3, and 4 m affect the vessel, the LSTM provides consistent DR position estimates, whereas the KF error increases.

Although the KF requires no initialization process, it does require a mathematical model of the vessel. Machine-learning models create an equivalent model based on data. That is why, from a cold start, machine-learning algorithms require a certain amount of time to construct and train the estimator. During this time, the DR functionality is unavailable. Although this is inconvenient, it may be remedied by performing the initialization process at regular intervals (or continuously) to have a DR model that is current with respect to the state of the vessel. Thereby, the model can seamlessly provide position estimates to a vessel operator or to the vessel's underlying automatic control system during a position reference system outage. This requires either maintaining a window of the most recent samples to perform batch training or feeding each individual sample to an online training algorithm for each of the two machine-learning methods. This is particularly important because we assume constant mean environmental forces during the DR process. If an old model is used, it may not reflect the characteristics of the current environmental state. In our approach, the training set consisted of 12 random weather conditions that were assumed to be representative of the complete set of possible weather conditions. We acknowledge that the relative performance of the methods proposed in this study is significantly affected by the value of the parameters of each method.

Conclusions

Through simulation studies, we compared the proposed LSTM neural network method with a conventional KF and an SLFN model. When no position or velocity measurements are available, the three methods utilize their own established model together with related model inputs. For the KF, these are given by the vessel model as thruster commands and forces due to wind. For the machine-learning methods, however, an initial input pattern was selected; then, input selection reduced this to a vector composed of approximately two thirds of the most relevant entries of the original input vector. This offered an improvement in terms of position estimation performance.

Findings suggest that the models created by machine-learning methods offer comparable performance in terms of position error drift, without requiring any vessel-dependent parameters. This shows that the dynamics of the vessel may be modeled without the development and tuning of a mathematical model. However, machine-learning methods offer no guarantees of convergence, being inherently black box. Therefore, future research should investigate in more detail how to establish a measure of confidence in the behavior of the machine-learning methods.

Acknowledgments

This work was supported in part by the project SFI Marine Operations in Virtual Environments and Digital Twins for Vessel Life Cycle Service (projects 237929 and 280703). Thor I. Fossen was partially funded by the Norwegian Research Council (Centre for Autonomous Marine Operations and Systems) at the Norwegian University of Science and Technology (grant 223254). We thank the staff of the Offshore Simulator Centre AS for their assistance related to interfacing with the simulator.

References

[1] M. Laurinen, "Remote and autonomous ships: The next steps," London, U.K.: Advanced Autonomous Waterborne Applications, Tech. Rep., 2016. [Online]. Available: https://www.rolls-royce.com/~/media/Files/R/Rolls-Royce/documents/customers/marine/ship-intel/aawa-whitepaper-210616.pdf

[2] J. Bhatti and T. E. Humphreys, "Hostile control of ships via false GPS signals: Demonstration and detection," *Navigation*, vol. 64, no. 1, pp. 51–66, 2017. doi: 10.1002/navi.183.

[3] A. J. Sørensen, "A survey of dynamic positioning control systems," *Annu. Rev. Control*, vol. 35, no. 1, pp. 123–136, 2011. doi: 10.1016/j.arcontrol.2011.03.008.

[4] M. Carter, "Moving towards a standardized interface for acoustic inertial reference systems," in *Proc. Dynamic Positioning Conf.*, 2014. [Online]. Available: https://dynamic-positioning.com/proceedings/dp2014/Sensors_carter.pdf

[5] O. Cadet, "Introduction to Kalman filter and its use in dynamic positioning systems," in *Proc. Dynamic Positioning Conf.*, 2003. [Online]. Available: https://dynamic-positioning.com/proceedings/dp2003/design_cadet.pdf

[6] T. I. Fossen and T. Perez, "Kalman filtering for positioning and heading control of ships and offshore rigs: Estimating the effects of waves, wind, and current," *IEEE Control Syst.*, vol. 29, no. 6, pp. 32–46, 2009.

[7] J. G. Balchen, N. A. Jenssen, E. Mathisen, and S. Sælid, "A dynamic positioning system based on Kalman filtering and optimal control," *Modeling Identification Control*, vol. 1, no. 3, pp. 135–163, 1980. doi: 10.4173/mic.1980.3.1.

[8] T. H. Bryne, T. I. Fossen, and T. A. Johansen, "Design of inertial navigation systems for marine craft with AdaptiveWave filtering aided by

triple-redundant sensor packages," *Int. J. Adaptive Control Signal Process.*, vol. 31, no. 4, pp. 522–544, 2015.
[9] R. H. Rogne, T. H. Bryne, T. I. Fossen, and T. A. Johansen, "MEMS-based inertial navigation on dynamically positioned ships: Dead reckoning," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 139–146, 2016. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S2405896316319218 doi: 10.1016/j.ifacol.2016.10.334.
[10] D. Russell, "Integrating GNSS and INS to provide reliable positioning," in *Proc. Dynamic Positioning Conf.*, 2012. [Online]. Available: https://dynamic-positioning.com/proceedings/dp2012/sensors1_russell.pdf
[11] B. Vik, A. Shiriaev, and T. I. Fossen, "Nonlinear observer design for integration of {DGPS} and {INS}," *New Directions Nonlinear Observer Design*, vol. 244, pp. 135–159, 1999. doi: 0.1007/BFb0109925.
[12] Z. Chu, D. Zhu, B. Sun, J. Nie, and D. Xue, "Design of a dead reckoning based motion control system for small autonomous underwater vehicle," in *Proc. 2015 IEEE 28th Canadian Conf. Electrical and Computer Engineering (CCECE)*, 2015, pp. 728–733.
[13] C. R. German et al., "A long term vision for long-range ship-free deep ocean operations: Persistent presence through coordination of Autonomous Surface Vehicles and Autonomous Underwater Vehicles," in *Proc. 2012 IEEE/OES Autonomous Underwater Vehicles*, 2012, pp. 1–7.
[14] R. Diamant and Y. Jin, "A machine learning approach for dead-reckoning navigation at sea using a single accelerometer," *IEEE J. Ocean. Eng.*, vol. 39, no. 4, pp. 672–684, 2014. doi: 10.1109/JOE.2013.2279421.
[15] H. Mokhtarzadeh and D. Gebre-Egziabher, "Performance of networked dead reckoning navigation system," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 5, pp. 2539–2553, 2016. doi: 10.1109/TAES.2016.150180.
[16] L. Fusini, T. A. Johansen, and T. I. Fossen, "Dead reckoning of a fixed-wing UAV with inertial navigation aided by optical flow," in *Proc. 2017 Int. Conf. Unmanned Aircraft Systems*, 2017, pp. 1250–1259.
[17] I. Skog and P. Händel, "In-car positioning and navigation technologies: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 4–21, 2009. doi: 10.1109/TITS.2008.2011712.
[18] E. Abbott and D. Powell, "Land-vehicle navigation using GPS," *Proc. IEEE*, vol. 87, no. 1, pp. 145–162, 1999. doi: 10.1109/5.736347.
[19] C. Barrios, Y. Motai, and D. Huston, "Intelligent forecasting using dead reckoning with dynamic errors," *IEEE Trans. Ind. Informat.*, vol. 12, no. 6, pp. 2217–2227, 2016. doi: 10.1109/TII.2015.2514086.
[20] H. Ahmed and M. Tahir, "Accurate attitude estimation of a moving land vehicle using low-cost MEMS IMU sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1723–1739, 2017. doi: 10.1109/TITS.2016.2627536.
[21] T. H. Bryne, "Nonlinear observer design for aided inertial navigation of ships," Ph.D. dissertation, Dept. Eng. Cybernetics, Norwegian Univ. Science and Technology, Trondheim, 2017.
[22] R. Mahony, T. Hamel, P. Morin, and E. Malis, "Nonlinear complementary filters on the special linear group," *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1557–1573, 2008. doi: 10.1080/00207179.2012.693951.
[23] L. P. Perera and C. G. Soares, "Weather routing and safe ship handling in the future of shipping," *Ocean Eng.*, vol. 130, pp. 684–695, Jan. 2017. doi: 10.1016/j.oceaneng.2016.09.007.
[24] U. D. Nielsen and J. J. Jensen, "A novel approach for navigational guidance of ships using onboard monitoring systems," *Ocean Eng.*, vol. 38, no. 2–3, pp. 444–455, 2011. doi: 10.1016/j.oceaneng.2010.11.024.
[25] E. M. Bitner-Gregersen et al., "Recent developments of ocean environmental description with focus on uncertainties," *Ocean Eng.*, vol. 86, pp. 26–46, Aug. 2014. doi: 10.1016/j.oceaneng.2014.03.002.
[26] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015. doi: 10.1016/j.neunet.2014.09.003.
[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[28] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer-Verlag, 2006.
[29] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse, "Methodology for long-term prediction of time series," *Neurocomputing*, vol. 70, no. 16–18, pp. 2861–2869, 2007. doi: 10.1016/j.neucom.2006.06.015.
[30] T. Trappenberg, J. Ouyang, and A. Back, "Input variable selection: Mutual information and linear mixing measures," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 37–46, 2006. doi: 10.1109/TKDE.2006.11.
[31] T. Suzuki, M. Sugiyama, T. Kanamori, and J. Sese, "Mutual information estimation reveals global associations between stimuli and biological processes," *BMC Bioinformatics*, vol. 10, no. 10, (Suppl. 1), p. S52, 2009.
[32] J. Snoek, H. Larochelle, and R. Adams, "Practical Bayesian optimization of machine learning algorithms" in *Proc. Neural Information Processing Systems Conf.*, pp. 2951–2959, 2012. [Online]. Available: https://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf
[33] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey Klaus," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, 2017.
[34] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Hoboken, NJ: Wiley, 2011.
[35] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," *Lecture Notes Comput. Sci.*, vol. 3512, pp. 758–770, June 2005. doi: 10.1007/11494669_93.

Robert Skulstad, Mechatronics Lab, Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Aalesund. Email: robert.skulstad@ntnu.no.

Guoyuan Li, Mechatronics Lab, Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Aalesund. Email: guoyuan.li@ntnu.no.

Thor I. Fossen, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim. Email: thor.fossen@ntnu.no.

Bjørnar Vik, Kongsberg Maritime, Aalesund, Norway. Email: bjornar.vik@km.kongsberg.com.

Houxiang Zhang, Mechatronics Lab, Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Aalesund. Email: hozh@ntnu.no.