Integrated GNSS/IMU Hub Motion Estimator for Offshore Wind Turbine Blade Installation

Zhengru Ren[a,b,c,*], Roger Skjetne[a,b,c], Zhiyu Jiang[d], Zhen Gao[a,b,c], Amrit Shankar Verma[a,c]

[a]*Centre for Research-based Innovation on Marine Operations (SFI MOVE), Norwegian University of Science and Technology (NTNU)*
[b]*Centre for Autonomous Marine Operations and Systems (AMOS), NTNU*
[c]*Department of Marine Technology, NTNU, NO-7491 Trondheim, Norway*
[d]*Department of Engineering Sciences, University of Agder, 4879 Grimstad, Norway.*

Abstract

Offshore wind turbines (OWTs) have become increasingly popular for their ability to harvest clean offshore wind energy. Bottom-fixed foundations are the most used foundation type. Because of its large diameter, the foundation is sensitive to wave loads. For typical manually assisted blade-mating operations, the decision to perform the mating operation is based on the relative distance and velocity between the blade root center and the hub, and in accordance with the weather window. Hence, monitoring the hub real-time position and velocity is necessary, whether the blade installation is conducted manually or automatically. In this study, we design a hub motion estimation algorithm for the OWT with a bottom-fixed foundation using sensor fusion of a global navigation satellite system (GNSS) and an inertial measurement unit (IMU). Two schemes are proposed based on a moving horizon estimator, a multirate Kalman filter, an online smoother, and a predictor. The moving horizon estimator mitigates the slow GNSS sampling rate relative to the hub dynamics. The multirate Kalman filter estimates the position, velocity, and accelerometer bias with a constant GNSS measurement delay. The online smoothing algorithm filters the delayed estimated trajectory to remove sudden step changes. The predictor compensates the delayed estimate, resulting in real-time monitoring. HAWC2 and MATLAB are used to verify the performance of the estimation algorithms, showing that a sufficiently accurate real-time position and velocity estimate with a high sampling rate is achieved. A sensitivity study compares the accuracy of different algorithms applied in various conditions. By combining both proposed algorithms, a sufficiently accurate estimation can be achieved for a wider scope of practical applications.

Keywords: Offshore wind turbine, offshore installation, bottom-fixed wind turbine, GPS, accelerometer, multirate Kalman filter, sensor fusion, real-time monitoring.

1. Introduction

Offshore wind turbines (OWTs) have become increasingly popular because of their high energy production and quality. Though sensitive to wave loads, bottom-fixed monopile foundations are the most common type, because of their simple shape, their large diameter, and a structural design driven mainly by fatigue loads [1]. Several publications focus on passive and active approaches to reduce the fatigue damages by, e.g., adding a tuned mass damper [2, 3], controlling the blade pitch for operating OWTs [4], and using a ball vibration absorber [5].

Nonetheless, the price of the electricity from offshore wind energy was three times higher than that from onshore wind energy in 2016 [6]. Large expenses for installation and maintenance are primary reasons for the high price. Two important approaches to reduce the cost of OWT installation are to improve the installation efficiency and to decrease the offshore waiting time between successive operations. Fifteen million euros is currently the extra cost of waiting for lower wind speed windows for the construction of a middle-sized wind farm in Europe [7]. Hence, several studies have been conducted with an emphasis on installation in high wind scenarios, thereby improving the installation efficiency. Studies on the critical allowable weather conditions for wind turbine installations are found in the literature [8–10]. In other studies [11], a closed-loop controller has been proposed to stabilize a suspended single blade. To achieve an effective mating performance, the blade root should follow the hub motion to reduce their relative displacement and relative velocity, consequently reducing the resulting impact force. Therefore, a method to monitor the hub displacement in real time is acutely necessary. The advantages and disadvantages of more than 20 commercial wind turbine condition monitoring systems, as well as their main functions, have been compared in the literature [12]. On-site acceleration data are recorded by the wind turbine condition monitoring systems; however, hub displacement monitoring is lacking [13].

To estimate a position trajectory, the most used strategies are the direct measurement from a global navigation satellite system (GNSS) and the double integration of the acceleration measurement, after a coordinate transformation, using an inertial measurement unit (IMU) [14, 15]. However, neither method is sufficient to handle the positioning problem for a dynamic system with high frequency. In this study, a global positioning system (GPS) is used as a representation of the GNSS, illustrating that vital shortages exist in the direct GPS measurement. A GPS receiver operates at a low sampling rate with inferior accuracy and with a time delay from the communication and computation. For a slow dynamic system, the low sampling rate and the time delay effects are insignificant, e.g., in the case of a dynamic positioning system of an offshore vessel. However, the wave-induced motion on the wind turbine foundation is not a slow dynamic system [16]. The delay of the GPS receiver can be several hundred

---

*zhengru.ren@ntnu.no

milliseconds, such that the intersample motion becomes significant. Therefore, a motion estimator must consider the time delay. The challenge of using measurements from an IMU is that the bias in the acceleration measurement results in a drift when integrated twice. The bias results from assembly flaws and the ambient conditions, such as temperature and humidity [17].

Integration of displacement and acceleration using a navigation filter has been widely applied in various areas. Normally, the estimation approaches are categorized into extended stochastic linear estimation techniques, deterministic complementary filter and nonlinear observer design techniques, and combinations of these techniques [18]. Linearized Kalman filters, such as the extended Kalman filter (EKF) and the unscented Kalman filter, are widely used in practice, even though global stability is typically not guaranteed [19, 20]. Nonlinear observers with stronger stability properties, however, are case-specific and lack generic design methodologies. An exogenous Kalman filter is a combination of a linearized Kalman filter and a nonlinear observer, which ensures both global stability and low variance in the estimates [21]. Examples of the application of these approaches are building deformation and earthquake monitoring [22–24], pedestrian navigation systems [25], dead reckoning [26], dynamic positioning systems [15, 27], and aerial applications [28, 29].

In this study, two hub motion estimation algorithms based on a GNSS/IMU fusion are customized to monitor the hub's real-time displacement and velocity for an OWT with a bottom-fixed monopile foundation. Combining both algorithms ensures sufficient accurate estimations for most practical applications.

This paper is structured as follows. In Section 2, the problem statement is proposed with foundation modeling and a motivational example. The characteristics of the hub motion are analyzed, and the complexity of the estimation scheme is reduced to ensure operational feasibility. In Section 3, the estimation model is proposed. In Section 4, two algorithms are developed based on a moving horizon estimator (MHE), a multirate Kalman filter (MKF), an online smoother, and a predictor. The algorithms monitor the OWT hub displacement and velocity in real time. In Section 5, HAWC2 and MATLAB are used to verify the performance of the monitoring algorithms. Sensitivity studies are conducted to evaluate the estimation performance. The conclusions are presented in Section 6.

Notation: Scalars, vectors, and matrices are expressed with normal letters, bold lowercase letters, and bold uppercase letters, respectively. The hat notation, $\hat{x}$, denotes the estimate of a state x. $\boldsymbol{I}$ and $\boldsymbol{0}$ are a 2×2 identity matrix and a zero matrix, i.e., $\boldsymbol{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\boldsymbol{0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$.

2. Problem formulation

An OWT with a monopile foundation without blades is simulated in HAWC2 to illustrate the problem. HAWC2 is an aeroelastic code designed to calculate the wind turbine response in the time domain [30]. In this study, the north-east-down coordinate system $\{n\}$ and body-fixed reference frame $\{b\}$ are adopted

and assumed inertial (see figures 1–2). Both of the frames follow the right-hand rule. The origin $\{n\}$ is located at the projection on the free water surface by the center of the cross-section of the substructure at rest. The x-axis points to the north, the y-axis points the east, and z-axis points downward. The origin of $\{b\}$ is placed at the location of the IMU, with the x^b- and z^b-axes are pointing to the hub and downward, respectively. The relative roll, pitch, and yaw orientation of $\{b\}$ in $\{n\}$ are denoted as ϕ_h, θ_h, and ψ_h, respectively.

2.1. Modeling of the monopile foundation and support structures

From low fidelity to high fidelity, the three most used monopile models are the apparent fixity model, the coupled spring model, and the distributed spring model. Differences among the models are mainly from the modeling of the soil-pile interaction below the mudline. In this study, the distributed spring model is adopted [31, 32]. The p-y model [33] is used in the simulation. The soil, modeled as elastic springs, is layered with different effective weights and angles of internal frictions. As defined by the American Petroleum Institute [34], the soil resistance is a function of the pile displacement at a given point along the pile. Each layer is modeled as an uncoupled nonlinear spring with a layer-specific stiffness as shown in Figure 1. The soil parameters can be found in Table 1. The soil has higher initial modulus of subgrade reaction with increasing depth. Timoshenko beams are used to model the pile, the transaction piece, the tower, and the hub [35–37].

Table 1: Soil profile [38, 39].

Depth under mudline (m)	Effective weight (kN/m^3)	Angle of internal friction (deg)
0–5	10.0	33.0
5–14	10.0	35.0
14–36	10.0	38.5

In the following, the environmental parameters are the mean wind speed U_w, the turbulence intensity TI, the wind turbulence seed S_{TI}, the main wind direction β_{wind}, the significant wave height H_s, the wave period T_p, the wave seed S_{wave}, and the wave heading β_{wave}. Normally, wave conditions for offshore operations are selected as $H_s \leq 3$ m and 4 s $\leq T_p \leq 10$ s. The IEC Class C wind turbulence model is chosen [40]. Structure parameters could be identified by various optimization approaches [41–43].

2.2. Illustrating example

An OWT with a 66-meter monopile foundation has been hammered into the seafloor to a depth of 36 m. The water depth is 30 m. All support structures have been assembled except the blades. The damping ratio of the foundation is tuned to be 1%. The hub is rotated to a fixed yaw $\psi_h = 0°$, ready for further installation operations. The foundation structure is exposed to a JONSWAP
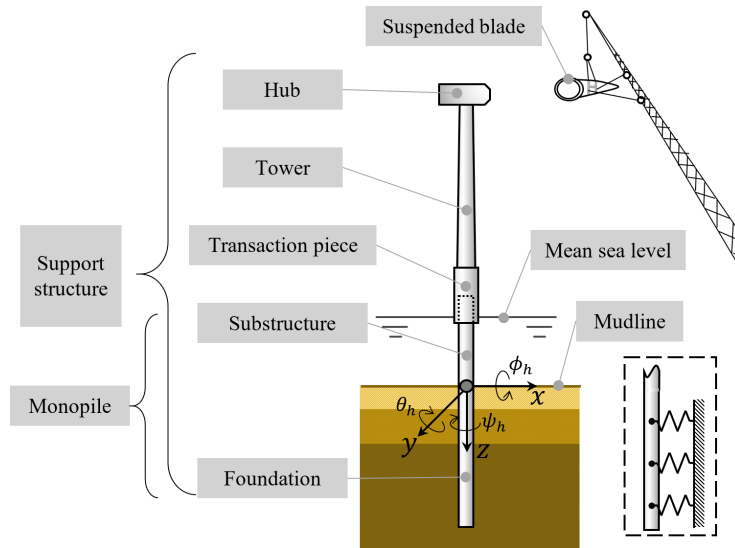
4

Figure 1: The single blade installation scenario with monopile foundation, tower, pre-assembled hub, suspended blade, and the distributed spring model.
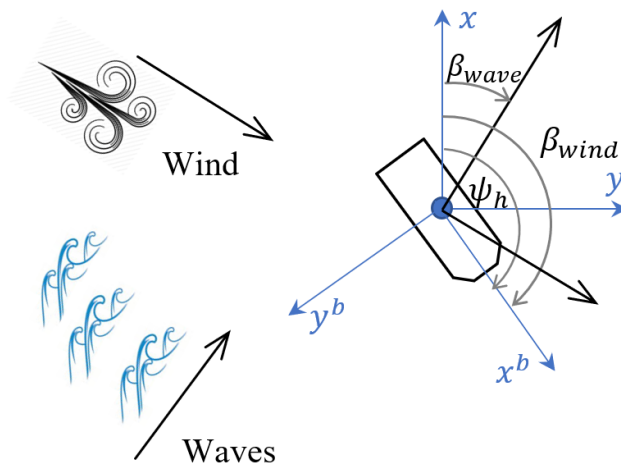

Figure 2: Proposed model in the horizontal plane.

Table 2: OWT structural and environmental parameters.

Parameter	Unit	Value
Sub-structure length	m	36
Foundation length	m	30
Tower length	m	77.6
Monopile diameter	m	6
Wave spectrum type	-	JONSWAP
Water depth	m	30
Significant wave height	m	2.87
Wave period	s	6.03
Monopile mass	tonnes	571.0
Tower mass	tonnes	347.5
Nacelle mass	tonnes	240.0

wave spectrum. The structural and environmental parameters are tabulated in Table 2. The waves spread to 30 deg north of east. The measurements are ideal, without sensor noise, bias, or delay. The simulation is conducted for 2000 s. The time domain simulation results of the hub displacement and its fast Fourier transform (FFT) results are presented in figures 3–4. Since no blade is considered in the proposed simulation, the main responses of the monopile and tower mainly result from the wave loads. The resonant responses at the first eigenmode at 3–4 seconds due to wave loads are dominated. Furthermore, the relative velocity between the hub and blade root is important when dealing with the final connection of the blade to hub. The velocity of the hub might be large due to the high resonant frequency.

The influence of the installed blades to the first fore-aft natural frequencies is presented in Table 3. As the total mass on top of the tower increases when more blades are installed, it slightly decreases the natural frequencies of the fore-aft modes. However, for the studied 5-MW wind turbine, the influences on the natural frequencies are very small, since the blades are lightweight (17 tonnes each) compared with total mass without blade (1158.5 tonnes), which includes the nacelle (240 tonnes), the tower (347.5 tonnes) and the monopile (571 tonnes).

Table 3: First fore-aft natural frequencies with different number of installed blades.

Number of installed blades	First fore-aft natural frequency (Hz)
0	0.258
1	0.249
2	0.239

Since the hub displacement spectra agree well with narrow-band spectra, a second-order wave-induced motion spectrum is used to fit the spectra, with the

following form:

$$S(f) = \frac{af}{f^2 + bf + c},$$ (1)

where a, b, and c are the coefficients. The derivative is

$$\frac{dS(f)}{df} = \frac{-a(f^2 - c)}{(f^2 + bf + c)^2},$$ (2)

which implies a maximum value $S(f^*) = \frac{a}{2\sqrt{c}+b}$ at $f^* = \sqrt{c}$. In Section 5.3, the time series generated by Eq. (1) are used to test the estimation performance. Moving the FFT results in Figure 4 leftward and rightward on the x-axis, the coefficients of the fitted resulting spectra are tabulated in Table 4. The oscillation amplitude, spectrum bandwidth, and structural first fore-aft natural frequency are determined by a, b, and c, respectively.

Table 4: Coefficients that determine the fitted spectra of the hub horizontal motion at different natural periods.

Natural period (s)	a	b	c
3	1.37e-06	-0.67	0.11
4	1.83e-06	-0.50	0.062
5	2.32e-06	-0.40	0.040
6	2.81e-06	-0.33	0.028
7	3.23e-06	-0.28	0.020
8	3.67e-06	-0.25	0.016

The key observations from the simulation results are as follows:

- The displacement of the hub is limited to a range in which the GPS noise is not negligible. Therefore, sensor noise from a GPS receiver of poor quality will be a problem.

- It is noted that the motion in the z-axis is small; hence, only the displacement in the horizontal plane, i.e., the x- and y-directions, are of interest.

- From the eigenvalue analysis results, the first fore-aft natural period of the structure is approximately 4 s. Consequently, the monopile wave-induced motion is not a slow dynamic process. Thus, the GPS update frequency may be too slow to capture the motion well. This situation means that the intersample motion between the GPS updates must be considered.

In summary, neither a GPS nor an accelerometer is independently able to measure the OWT hub displacement adequately. The GPS noise, GPS time delay, GPS low sampling rate, accelerometer noise, and accelerometer bias are significant factors that influence the hub displacement in a real-time motion estimator.
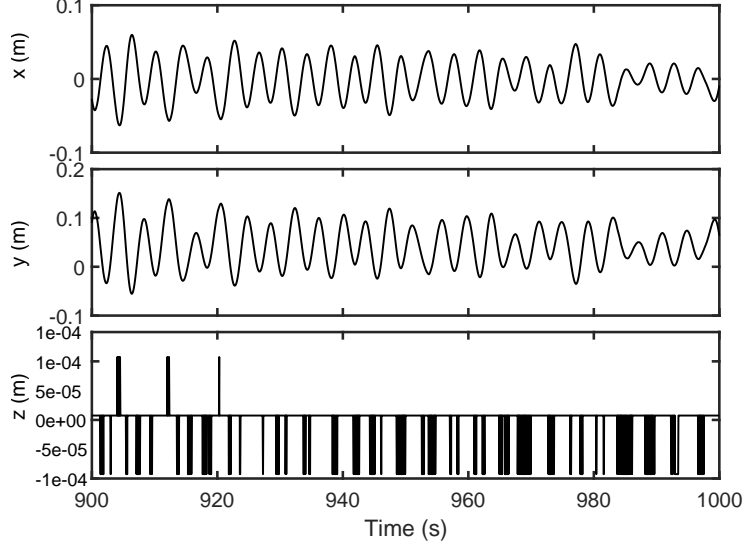
Figure 3: The history of the hub position for the proposed OWT w.r.t. the equilibrium, $H_s = 2$ m, $T_p = 8$ s, $\beta_{wave} = 30°$, $S_{wave} = 75$, $U_w = 8$ m/s, $TI = 0.146$, and $S_{TI} = 94$.

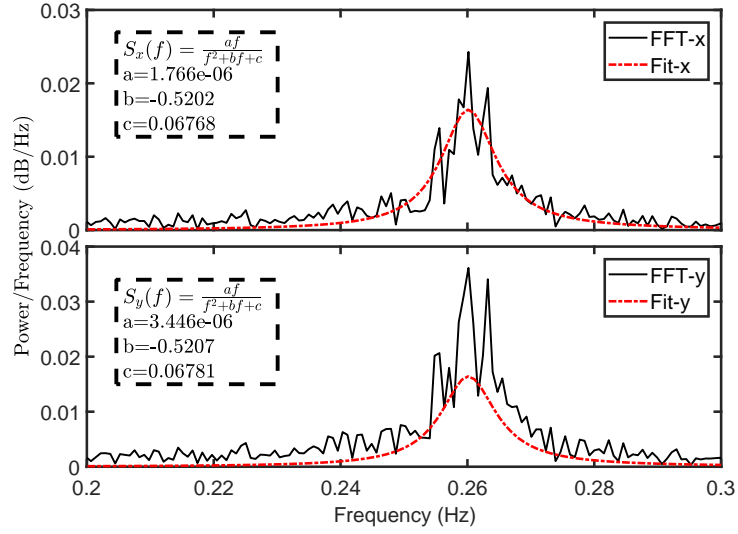

Figure 4: The hub displacement spectra of measurement from Figure 3 after an FFT, $H_s = 2$ m, $T_p = 8$ s, $\beta_{wave} = 30°$, $S_{wave} = 75$, $U_w = 8$ m/s, $TI = 0.146$, and $S_{TI} = 94$.

8

2.3. Problem statement

The objective of this study is to develop an integrated motion estimator using sensor fusion of the GPS and IMU measurements. It should provide real-time estimates of the position, velocity, and acceleration of an oscillating OWT hub at the mating height. For a single blade installation operation, the relative distance between the hub and the blade root affects the successful mating rate, and the relative velocity influences the impact force [10]. Therefore, real-time hub position and velocity estimates with a sufficient accuracy and high sampling rate are necessary.

We assume the availability of a DGPS receiver and an IMU that are assembled as a sensor suite near the hub attachment; the sample time steps for them are T_g and T_a, respectively, with $T_g \gg T_a$. The GPS operates at a lower sampling rate with a delay τ_d. It is assumed that the time delay of the GPS signal varies as a function of the number of connected satellites, which is a known constant during a given period. Define an integer $N_d \simeq \frac{\tau_d}{T_a}$. The sampling interval of the GPS is assumed to be an integer multiple of the IMU sampling interval, i.e., $T_g/T_a = M$ where $M \geq 2$ is an integer. The IMU provides a real-time noisy and biased specific force measurement at a high sampling rate [15, 17]. The IMU data are received in a sensor-fixed reference frame at a fixed location in $\{b\}$. Zero-order hold (ZOH) sampling is used to simulate the discrete measurements, i.e., each measurement remains constant until the next sampling instant. The assumptions are as follows:

1. The hub has been rotated to a fixed known yaw ψ_h and ψ_h is measured by a compass, where the wave- and wind-induced motions have a negligible influence on ψ_h.
2. The variation in rotation of the IMU about the x- and y-axes caused by the monopile motion is negligible, i.e., $\phi_h \approx 0$ and $\theta_h \approx 0$.
3. The gravity vector is assumed constant in the NED coordinates.
4. The NED frame is assumed inertial, which implies the effect of the rotation of the earth is negligible.
5. The uncertain sensor parameters, i.e., unknown sensor bias and time delay, are constant during a given short period.
6. The GPS bias is negligible.

The first fore-aft natural frequency of the hub and the first-order natural frequency of the pendulum motion of the blade are close to 0.26 Hz and 0.09 Hz, respectively. To accurately replicate both the shape and amplitude of a periodic motion, a fast sampling is essential. Hence, 10 Hz is a suitable sampling frequency. Moreover, since the diameter of the guide pins are only a few centimeters, a centimeter level accuracy positioning is necessary.

The IMU specific force measurement is transferred to $\boldsymbol{f}_{imu}$ in $\{b\}$ by a constant coordinate transformation. A simplified horizontal-plane measurement model is given by

$$\boldsymbol{f}_{imu}(t) = \boldsymbol{R}^\top (\dot{\boldsymbol{v}}(t) - \boldsymbol{g}(t)) + \boldsymbol{b}_a^b(t) + \boldsymbol{w}_a^b(t), \tag{3}$$

$$\boldsymbol{p}_{gps}(t) = \boldsymbol{p}(t - \tau_d) + \boldsymbol{v}_g(t), \tag{4}$$

where $\boldsymbol{f}_{imu}$ and $\boldsymbol{p}_{gps} \in \mathbb{R}^2$ are the specific force and position measurements from the IMU in $\{b\}$ and the GPS in $\{n\}$, respectively, $\boldsymbol{p} \in \mathbb{R}^2$ is the real GPS receiver position, $\boldsymbol{v} \in \mathbb{R}^2$ is the velocity in $\{b\}$, $\dot{\boldsymbol{v}}$ is the acceleration at the IMU mounting location, $\boldsymbol{g} \in \mathbb{R}^2$ is the acceleration caused by the gravity in $\{n\}$, $\boldsymbol{b}_a^b \in \mathbb{R}^2$ is the bias in the specific force measurements, $\boldsymbol{w}_a^b$ and $\boldsymbol{v}_g \in \mathbb{R}^2$ denote the zero-mean measurement noise from the IMU and GPS, t refers to the time, τ_d is the delayed time caused by the GPS position estimate and the data communication time, and the rotation matrix is

$$\boldsymbol{R} = \begin{bmatrix} \cos\psi_h & -\sin\psi_h \\ \sin\psi_h & \cos\psi_h \end{bmatrix}, \tag{5}$$

where ψ_h is received from compass measurements.

The motions on the top of a floating wind turbine, e.g., a floating spar, is easier to estimate, since the wave-frequency and low-frequency motions are dominate, instead of the resonant frequency of the tower vibration. Therefore, the motion estimation for a bottom-fixed wind turbine is more critical.

3. Observer design

The state estimate problem is modeled as a Wiener process. Because of assumptions 1 and 4, the terms $\boldsymbol{R}$ and $\boldsymbol{g}$ are assumed to be known constants.

We propose a hub motion estimator that consists of a multirate Kalman filter to estimate the state trajectories with optimal filtering of the measurements, a smoothing filter for the step changes that occur due to the slower GNSS updates, and a state predictor to compensate for the delays of the GNSS and the smoothing filter. To ensure good filtering of the IMU measurements, we define the acceleration as a state driven by white noise in the process model used in the MKF, as shown in model (6). In this way, the issue of the acceleration estimate being a direct feedthrough of the IMU measurements is avoided. In the moving horizon estimator and predictor function, however, it is convenient to define the IMU acceleration measurement as an input driving the position/velocity dynamics. Accordingly, we use a reduced model with only position and velocity as states for the predictor, as shown in model (8).

3.1. Process model for the Kalman filter

One approach is to set acceleration as a state, and the estimate can output filtered acceleration. Hence, the state-space representation is then given by

$$\begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \\ \dot{\boldsymbol{x}}_3 \\ \dot{\boldsymbol{x}}_4 \end{bmatrix} (t - \tau_d) = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{bmatrix} (t - \tau_d) + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{w}_b(t - \tau_d), \tag{6a}$$

$$\begin{bmatrix} \boldsymbol{p}_{gps}(t) \\ \boldsymbol{f}_{imu}(t - \tau_d) \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}^\top & -\boldsymbol{R}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{bmatrix} (t - \tau_d) + \begin{bmatrix} \boldsymbol{v}_g \\ \boldsymbol{w}_a \end{bmatrix} (t), \tag{6b}$$

10

where $\boldsymbol{x}_1 = [x, y]^\top$ denotes the position vector indicating the location of the sensors at the hub, $\boldsymbol{x}_2 = [v_x, v_y]^\top = \dot{\boldsymbol{x}}_1$ is the velocity vector in $\{n\}$, $\boldsymbol{x}_3 = [a_x, a_y]^\top = \dot{\boldsymbol{x}}_2$ is the acceleration vector in $\{n\}$, $\boldsymbol{x}_4 = -\boldsymbol{b}_a + \boldsymbol{g} = [b_{ax}, b_{ay}]^\top$ with $\boldsymbol{b}_a = \boldsymbol{R}\boldsymbol{b}_a^b$ refers to the combined bias and gravity vector, $\boldsymbol{w}_b \sim (0, \boldsymbol{Q}_b)$ with $\boldsymbol{Q}_b \in \mathbb{R}^{2\times2}$, $\boldsymbol{w}_a = \boldsymbol{R}\boldsymbol{w}_a^b \sim (0, \boldsymbol{Q}_a)$, and $\boldsymbol{v}_g \sim (0, \boldsymbol{Q}_g)$ with $\boldsymbol{Q}_g \in \mathbb{R}^{2\times2}$. The diagonal elements of $\boldsymbol{Q}_g \in \mathbb{R}^{2\times2}$ and $\boldsymbol{Q}_a \in \mathbb{R}^{2\times2}$ represent the covariances of the GPS noise and accelerometer noise, respectively. The vector form is

$$\Sigma_1 : \begin{cases} \dot{\boldsymbol{x}}(t - \tau_d) = \boldsymbol{A}\boldsymbol{x}(t - \tau_d) + \boldsymbol{E}\boldsymbol{w}(t - \tau_d), & \text{(7a)} \\ \boldsymbol{z} = \boldsymbol{H}\boldsymbol{x}(t - \tau_d) + \boldsymbol{v}_s(t), & \text{(7b)} \end{cases}$$

where $\boldsymbol{x} = [\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, \boldsymbol{x}_3^\top, \boldsymbol{x}_4^\top]^\top \in \mathbb{R}^8$, $\boldsymbol{z}(t) = [\boldsymbol{p}_{gps}^\top(t), \boldsymbol{f}_{imu}^\top(t - \tau_d)]^\top \in \mathbb{R}^4$, $\boldsymbol{A} \in \mathbb{R}^{8\times8}$, $\boldsymbol{H} \in \mathbb{R}^{4\times8}$, and $\boldsymbol{w} = \boldsymbol{w}_b \in \mathbb{R}^2 \sim (0, \boldsymbol{Q})$ and $\boldsymbol{v}_s = [\boldsymbol{v}_g^\top, \boldsymbol{w}_a^\top]^\top \in \mathbb{R}^4 \sim (0, \boldsymbol{\Omega})$ with diagonal matrices $\boldsymbol{Q} = \boldsymbol{Q}_b$ and $\boldsymbol{\Omega} = \text{diag}(\boldsymbol{Q}_g, \boldsymbol{Q}_a)$. The trends illustrated by Eq. (6) have been proven [44].

3.2. Process model for the predictor and MHE

Another approach is to let the acceleration be an input that drives the position/velocity dynamics. The aim of this model is to design the MHE. Similar to Σ_1, the model is then given by

$$\begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \\ \dot{\boldsymbol{x}}_4 \end{bmatrix} (t - \tau_d) = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_4 \end{bmatrix} (t - \tau_d) + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{R}\boldsymbol{f}_{imu}(t - \tau_d) \quad \text{(8a)}$$

$$+ \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_a \\ \boldsymbol{w}_b \end{bmatrix} (t - \tau_d),$$

$$\boldsymbol{z}(t) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_4 \end{bmatrix} (t - \tau_d) + \boldsymbol{v}_g(t), \quad \text{(8b)}$$

where $\boldsymbol{x}_1 = [x, y]^\top$, $\boldsymbol{x}_2 = [v_x, v_y]^\top = \dot{\boldsymbol{x}}_1$, $\boldsymbol{x}_4 = -\boldsymbol{b}_a + \boldsymbol{R}\boldsymbol{g} = [b_{ax}, b_{ay}]^\top$, and $\boldsymbol{z}(t) = \boldsymbol{p}_{gps}(t)$. The vector form is

$$\Sigma_2 : \begin{cases} \dot{\boldsymbol{x}}(t - \tau_d) = \boldsymbol{A}\boldsymbol{x}(t - \tau_d) + \boldsymbol{B}\boldsymbol{u}(t - \tau_d) + \boldsymbol{E}\boldsymbol{w}(t - \tau_d), & \text{(9a)} \\ \boldsymbol{z}(t) = \boldsymbol{H}\boldsymbol{x}(t - \tau_d) + \boldsymbol{v}_s(t), & \text{(9b)} \end{cases}$$

where $\boldsymbol{x} = [\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, \boldsymbol{x}_3^\top]^\top \in \mathbb{R}^6$, $\boldsymbol{u} = \boldsymbol{R}\boldsymbol{f}_{imu} \in \mathbb{R}^2$, $\boldsymbol{A} \in \mathbb{R}^{6\times6}$, $\boldsymbol{B} \in \mathbb{R}^{6\times2}$, $\boldsymbol{H} \in \mathbb{R}^{2\times6}$, $\boldsymbol{w} = [\boldsymbol{w}_a^\top, \boldsymbol{w}_b^\top]^\top \in \mathbb{R}^4 \sim (0, \boldsymbol{Q})$ with $\boldsymbol{Q} = \text{diag}(\boldsymbol{Q}_a, \boldsymbol{Q}_b)$, and $\boldsymbol{v}_s = \boldsymbol{v}_g \in \mathbb{R}^2 \sim (0, \boldsymbol{\Omega})$ with $\boldsymbol{\Omega} = \boldsymbol{Q}_g$.

3.3. Discrete system

Using a shift register to receive $\boldsymbol{f}_{imu}(t)$, the effects of time delay are canceled. Discretizing the continuous system (7) or (9) and ignoring the system

11

disturbance and measurement noise yields

$$\boldsymbol{x}_{k+1}(k - N_d) = \begin{cases} \boldsymbol{A}_d \boldsymbol{x}_k(k - N_d) & (\Sigma_1) \\ \boldsymbol{A}_d \boldsymbol{x}_k(k - N_d) + \boldsymbol{B}_d \boldsymbol{u}(k - N_d) & (\Sigma_2) \end{cases}, \tag{10a}$$

$$\boldsymbol{z}_k(k - N_d) = \boldsymbol{H} \boldsymbol{x}_k(k - N_d), \tag{10b}$$

where the subscript k denotes the time instance $t_k = kT_s$, where T_s is the sampling interval. In this paper, $T_s = T_a$. The discrete matrices are

$$\boldsymbol{A}_d = e^{\boldsymbol{A}T_a} = \begin{cases} \boldsymbol{I}_{8\times8} + \boldsymbol{A}T_a, & (\Sigma_1) \\ \boldsymbol{I}_{6\times6} + \boldsymbol{A}T_a, & (\Sigma_2) \end{cases} \tag{10c}$$

$$\boldsymbol{B}_d = T_a \boldsymbol{B} + \frac{T_a^2}{2} \boldsymbol{A}\boldsymbol{B}, \tag{10d}$$

$$\boldsymbol{Q}_d = \int_0^{T_a} e^{\boldsymbol{A}\tau} \boldsymbol{Q} e^{\boldsymbol{A}^\top \tau} d\tau = \boldsymbol{Q}T_a + \boldsymbol{Q}\boldsymbol{A}T_a^2 + \boldsymbol{A}\boldsymbol{Q}\boldsymbol{A}^\top \frac{T_a^3}{3}, \tag{10e}$$

$$\boldsymbol{\Omega}_d = \boldsymbol{\Omega}/T_a. \tag{10f}$$

4. State estimation and prediction algorithms

A state estimation algorithm that compensates the GNSS delay and provides real-time monitoring of the hub position, velocity, and acceleration is required. We propose two algorithms in this section. The hub motion estimator 1 (HME1) has a low complexity because it does not include a moving horizon estimator and a nonlinear program solver. Based on HME1, the hub motion estimator 2 (HME2) is developed to improve the prediction performance for the system with longer natural period by introducing an MHE.

4.1. Algorithm 1 - HME1

The workflow diagram is presented in Figure 6, and its steps are tabulated in Algorithm 1. Only Σ_1 in (7) is necessary, and the observer runs at a high frequency with an interval T_a. For every time step, the shift register #1 stores the real-time IMU specific force measurement. The MKF estimates the states, based on the delayed GPS measurement and the delayed acceleration measurement. In a GPS sampling interval, the position measurements inputs to the MKF once. Shift register #4 stores the data $\hat{\boldsymbol{x}}(t - \tau_d)$ and the covariance matrices. Then the stored data are supplied to the online smoother. The online Rauch, Tung, and Striebel (RTS) smoother conducts backward smoothing [45]. The additional delay caused by the smoother is τ_s so that the estimate becomes $\hat{\boldsymbol{x}}(t - \tau_d - \tau_s)$. A predictor is used to bring forward the estimates to the present time t. Then, the process moves on to the next step, and the above procedures are repeated. See Algorithm 1.
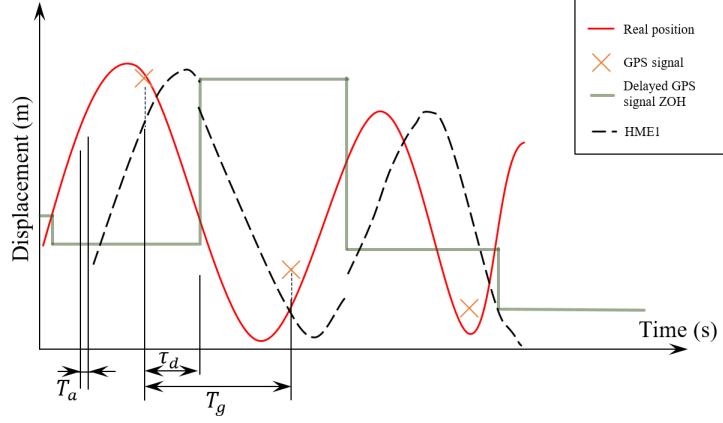
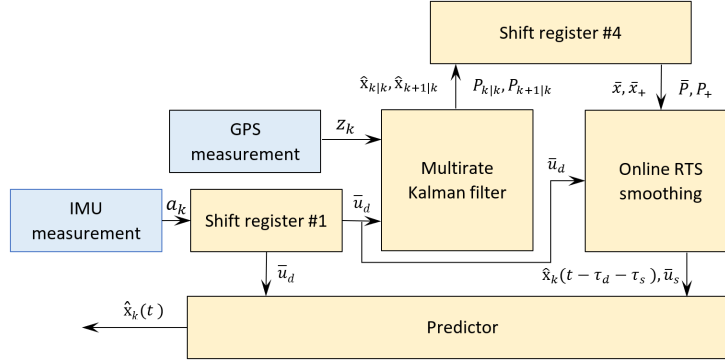Figure 5: Sketch of how the HME1 works.

Figure 6: Workflow diagram of the proposed algorithm (HME1).

Algorithm 1 HME1.

Monopile hub position estimation$(\boldsymbol{z}_k, \boldsymbol{a}_k, t, T_g, N_d, N_s)$

Initialize: $\bar{\boldsymbol{u}}_d = \boldsymbol{0}_{2 \times N_d}, \bar{\boldsymbol{u}}_s = \boldsymbol{0}_{2 \times N_s}, \bar{\boldsymbol{x}} = \boldsymbol{0}_{9 \times N_s}, \bar{\boldsymbol{x}}_+ = \boldsymbol{0}_{9 \times N_s}, \bar{\boldsymbol{P}} = \boldsymbol{0}_{9 \times 9 \times N_s}, \bar{\boldsymbol{P}}_+ = \boldsymbol{0}_{9 \times 9 \times N_s}$

1: **while** k **do**
2:     $\bar{\boldsymbol{u}}_d = $ Shift register #1$(\boldsymbol{a}_k)$
3:     $[\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}, \hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{P}_{k+1|k}] = $Multirate Kalman filter$(\boldsymbol{z}_k, \bar{\boldsymbol{u}}_d, t, T_g)$
4:     $[\bar{\boldsymbol{x}}, \bar{\boldsymbol{P}}, \bar{\boldsymbol{x}}_+, \bar{\boldsymbol{P}}_+] = $Shift register #2$(\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}, \hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{P}_{k+1|k})$
5:     $[\hat{\boldsymbol{x}}_k(t - \tau_d - \tau_s), \bar{\boldsymbol{u}}_s] = $Online RTS smoothing$(\bar{\boldsymbol{x}}, \bar{\boldsymbol{P}}, \bar{\boldsymbol{x}}_+, \bar{\boldsymbol{P}}_+, \bar{\boldsymbol{u}}_d)$
6:     $\hat{\boldsymbol{x}}_k(t) = $Predictor$(\hat{\boldsymbol{x}}_k(t - \tau_d - \tau_s), \bar{\boldsymbol{u}}_d, \bar{\boldsymbol{u}}_s, N_d, N_s)$
7:     $k = k + 1$
8: **end while**
    return $\hat{\boldsymbol{x}}_k(t)$

13

4.2. Algorithm 2 - HME2

Based on the periodic motion of the hub, it is possible to predict the hub motion at a time instant in the near future based on past observations. Moreover, the predictor in HME1 is a simple deterministic open-loop simulation that does not take into account the stochastic properties of the measurements. Hence, to further improve the estimation performance, a more optimal HME2 is proposed. The differences between HME1 and HME2 are illustrated in Figure 7. In Algorithm 2, we introduce an MHE. This gives optimal predictions to the states during the gap between two GPS updates, i.e., the MHE resamples the GPS sampling interval with a number of optimal position predictions according to past measurements and the process model Σ_2. This approach is equivalent to increasing the position sampling rate. Consequently, the MKF position estimation is corrected at a higher frequency, which helps overcome large drifts accumulated during a GPS sampling interval.

It is noted that the fidelity of the predictions, according to the discrete GPS signal sequences, depends on the first fore-aft natural frequency of the hub. Similar to the aliasing effect, limited GPS sampling points from the continuous narrow-band hub position history implies lost information regarding the dynamics and a less accurate prediction. Therefore, HME2 is more suitable for oscillations with a longer natural period.

The workflow diagram and process are presented in Figure 8 and Algorithm 2. The calculation interval of the MHE solver is T_m, i.e., $T_a < T_m < T_g$. Shift register #2 updates every T_g and outputs stored data with delayed IMU specific force and GPS measurements every T_m. Together with the initial value provided by shift register #3, $\boldsymbol{x}_{m0,k}$, the MHE solver calculates the delayed states $\boldsymbol{x}_m(t - \tau_d)$ using nonlinear programming. The yellow blocks (multirate Kalman filter, smoother, and predictor) remain the same as those used in HME1 by taking the position $\hat{\boldsymbol{z}}_k$ from $\boldsymbol{x}_m(t - \tau_d)$, instead of taking $\boldsymbol{z}_k$ directly from the GPS measurements.
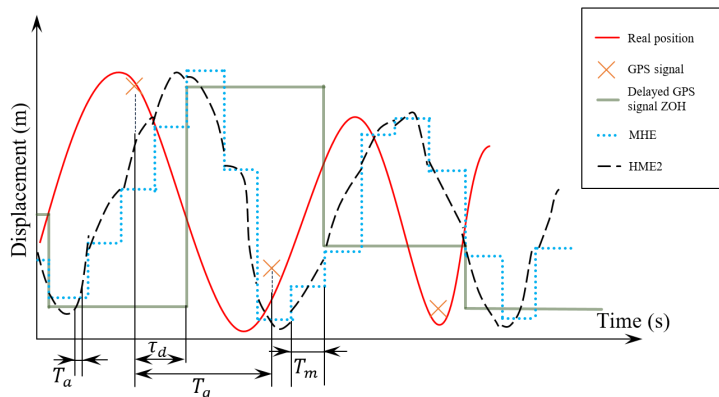

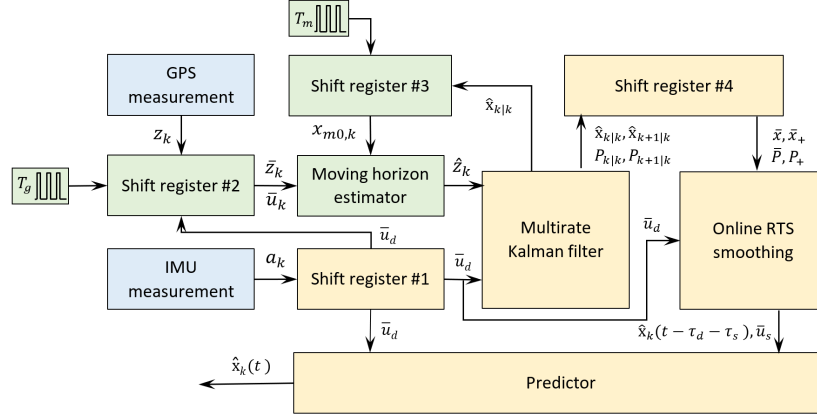Figure 7: Sketch of how the HME2 works with MHE.

14

Figure 8: Workflow diagram of the proposed algorithm with moving horizon estimator (HME2).

Algorithm 2 HME2.

Monopile hub position estimation$(\boldsymbol{z}_k, \boldsymbol{a}_k, t, T_g, T_m, N_d, N_s, N_m)$

Initialize: $\bar{\boldsymbol{u}}_d = \boldsymbol{0}_{2\times N_d}, \bar{\boldsymbol{u}}_s = \boldsymbol{0}_{2\times N_s}, \bar{\boldsymbol{x}}_m = \boldsymbol{0}_{2\times N_m}, \bar{\boldsymbol{x}} = \boldsymbol{0}_{9\times N_s}, \bar{\boldsymbol{x}}_+ = \boldsymbol{0}_{9\times N_s}, \bar{\boldsymbol{P}} = \boldsymbol{0}_{9\times 9\times N_s}, \bar{\boldsymbol{P}}_+ = \boldsymbol{0}_{9\times 9\times N_s}$

1: **while** k **do**
2:   $\bar{\boldsymbol{u}}_d = $ Shift register #1$(\boldsymbol{a}_k)$
3:   $[\bar{\boldsymbol{u}}_k, \bar{\boldsymbol{z}}_k] = $ Shift register #2$(\bar{\boldsymbol{u}}_d, z_k)$
4:   $\boldsymbol{x}_{m0,k} = $ Shift register #3$(\boldsymbol{x}_{k|k})$
5:   $\hat{\boldsymbol{z}}_k = $ Moving horizon estimator$(\bar{\boldsymbol{u}}_k, \bar{\boldsymbol{z}}_k, \boldsymbol{x}_{m0,k})$
6:   $[\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}, \hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{P}_{k+1|k}] = $ Multirate Kalman filter$(\hat{\boldsymbol{z}}_k, \bar{\boldsymbol{u}}_d, t, T_m)$
7:   $[\bar{\boldsymbol{x}}, \bar{\boldsymbol{P}}, \bar{\boldsymbol{x}}_+, \bar{\boldsymbol{P}}_+] = $ Shift register #4$(\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}, \hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{P}_{k+1|k})$
8:   $[\hat{\boldsymbol{x}}_k(t - \tau_d - \tau_s), \bar{\boldsymbol{u}}_s] = $ Online RTS smoothing$(\bar{\boldsymbol{x}}, \bar{\boldsymbol{P}}, \bar{\boldsymbol{x}}_+, \bar{\boldsymbol{P}}_+, \bar{\boldsymbol{u}}_d)$
9:   $\hat{\boldsymbol{x}}_k(t) = $ Predictor$(\hat{\boldsymbol{x}}_k(t - \tau_d - \tau_s), \bar{\boldsymbol{u}}_d, \bar{\boldsymbol{u}}_s, N_d, N_s)$
10:   $k = k + 1$
11: **end while**
  return $\hat{\boldsymbol{x}}_k(t)$

4.3. Implementation

4.3.1. Delayed IMU measurement and stored signals

To input the delayed IMU measurement, Algorithm 3 is applied to shift the measurement N_d steps. Furthermore, the IMU specific force measurement during the delay is stored and input to the predictor in Algorithm 9.

Algorithm 3 Shift register #1.

Shift register $\#1(\boldsymbol{a}_k)$
1: $\bar{\boldsymbol{u}}_d = [\boldsymbol{u}_d(:, 2 : N_d), \boldsymbol{a}_k]$
   return $\bar{\boldsymbol{u}}_d$

4.3.2. Moving horizon estimator

The GPS and IMU do not sample synchronously. The aim of the MHE is to fill the gap between two subsequent GPS updates with reasonable state predictions. For a discrete dynamics model Σ_2, the current state $(\boldsymbol{x}_k)$ depends on the sequence of past inputs $\boldsymbol{U}_k = \mathrm{col}(\boldsymbol{u}_{k-N} \cdots \boldsymbol{u}_{k-N})$ and its initial value $\boldsymbol{x}_{k-N}$. The measurement $\boldsymbol{z}_k$ is a mapping of the states $\boldsymbol{x}_k$. Therefore, the states $\boldsymbol{X}_k = \mathrm{col}(\boldsymbol{x}_{k-N} \cdots \boldsymbol{x}_{k-N})$ are estimated with a sequence of measurements $\boldsymbol{Z}_k = \mathrm{col}(\boldsymbol{z}_{k-N} \cdots \boldsymbol{z}_{k-N})$ by minimizing the error of the measurement error and the disturbance. The states at the last step can be treated as filtered signals. Then, the position estimation from the MHE inputs to the MKF. First, two shift registers are required to provide the historic measurements and inputs, as well as the initial value for the MHE solver; see algorithms 4–5.

Algorithm 4 Shift register #2.

Shift register $\#2(\bar{\boldsymbol{u}}_d, z_k)$
1: $\boldsymbol{u}_k = \bar{\boldsymbol{u}}_d(:, 1)$
2: $\bar{\boldsymbol{u}}_k = [\boldsymbol{u}_d(:, 2 : N_d), \boldsymbol{u}_k]$
3: $\bar{\boldsymbol{z}}_k = [\boldsymbol{u}_d(:, 2 : N_d), \boldsymbol{z}_k]$
   return $\bar{\boldsymbol{u}}_k,\ \bar{\boldsymbol{z}}_k$

Algorithm 5 Shift register #3.

Shift register $\#3(\boldsymbol{x}_{k|k})$
1: $\bar{\boldsymbol{x}}_m = [\boldsymbol{u}_m(:, 2 : N_m), \boldsymbol{x}_{k|k}]$
2: $\boldsymbol{x}_{m0,k} = \bar{\boldsymbol{x}}_m(:, 1)$
   return $\boldsymbol{x}_{m0,k}$

$\boldsymbol{Z}$ and $\boldsymbol{U}$ update every T_g, and the MHE solves the nonlinear program with an interval T_m. At all steps except the MHE updating time, i.e., $\mathrm{rem}(t, T_m) = 0$, the MHE should predict the current state based on past measurements and

inputs. The cost function is given by

$$\min_{S,Q} \sum_{i=k-N_m+1}^{k-1} \|\boldsymbol{w}_i\|_{\mathcal{Q}} + \sum_{i=k-N_m}^{k-1} +\|\boldsymbol{v}_i\|_{\mathcal{R}} + \|\boldsymbol{v}_k\|_{\mathcal{R}_f}, \tag{11a}$$

$$\text{s.t.} \quad \boldsymbol{w}_i = \boldsymbol{x}_{i+1} - (\boldsymbol{A}_d\boldsymbol{x}_i + \boldsymbol{B}_d\boldsymbol{u}_i), \qquad i = k - N_m + 1, \cdots, k-1, \tag{11b}$$

$$\boldsymbol{v}_i = \boldsymbol{z}_i - \boldsymbol{C}_d\boldsymbol{x}_i, \qquad i = k - N_m, \cdots, k-1, \tag{11c}$$

$$\boldsymbol{w}_k = \boldsymbol{x}_k - (\boldsymbol{A}_{df}\boldsymbol{x}_{k-1} + \boldsymbol{B}_{df}\boldsymbol{u}_{k-1}), \tag{11d}$$

$$\boldsymbol{v}_k = \boldsymbol{z}_k - \boldsymbol{C}_d\boldsymbol{x}_k, \tag{11e}$$

$$t_f = T_g + \text{rem}(t, T_g) \tag{11f}$$

$$\boldsymbol{A}_{df} = \boldsymbol{I}_{6\times6} + t_f\boldsymbol{A}, \tag{11g}$$

$$\boldsymbol{B}_{df} = t_f\boldsymbol{B} + t_f^2/2\boldsymbol{AB}, \tag{11h}$$

$$\mathcal{R}_f = \begin{cases} \mathcal{R} > 0, & \text{if } t_f = 0, \\ \boldsymbol{0}, & \text{otherwise}, \end{cases} \tag{11i}$$

where $\text{rem}(a,b)$ is the remainder after division of a by b, $\mathcal{Q}$ and $\mathcal{R}$ are positive-definite diagonal weighting matrices, and the weighted Euclidean norm is expressed as $\|\boldsymbol{w}_i\|_Q^2 = \boldsymbol{w}_i^\top \mathcal{Q} \boldsymbol{w}_i$.

4.3.3. Multirate Kalman filter

A Kalman filter is used to filter the noisy measurements in the state equation Σ_1. Because the sampling frequency in the GPS and the IMU are different, the MKF is modified from the classic Kalman filter, as illustrated in Figure 5. The MKF works at the same frequency as the IMU. When the GPS signal is available, i.e., at $t/T_g =$ integer steps, the GPS signal corrects the estimate. Otherwise, at a moment when the GPS measurement is not available, i.e., at $t/T_g \neq$ integer, the part of the Kalman gain matrix $\boldsymbol{K}$ according to the position measurements is set to zero, and the estimate is updated with merely the acceleration measurements as a state in Eq. (7) or as an input in Eq. (9). The error grows until it is corrected at the next GPS update. The summarized algorithm is tabulated in Algorithm 6. At this stage, the MKF calculates the delayed estimate $t - \tau_d$.

4.3.4. Smoothing

The MKF leads to discontinuous points when $t/T_g =$ integer. Therefore, smoothing is conducted to smooth the estimate along the time history. There are three categories of smoothing: fixed-interval smoothing, fixed-point smoothing, and fixed-lag smoothing.

A widely used offline fixed-interval smoothing method is the RTS smoother, also referred to as the two-pass smoother [45]. A forward pass is the Kalman filter, which computes the filtered quantities, $\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}, \hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{P}_{k+1|k}$, and stores them in the memory. The backward pass smooths the estimate. However, the RTS smoother is an offline algorithm that process the stored estimate data from the Kalman filter. At the end of an estimation on a given dataset, the

Algorithm 6 Multirate Kalman filter.

Multirate Kalman filter$(\hat{\boldsymbol{z}}_k, \bar{\boldsymbol{u}}_d, t, T)$

1: **Initialize**: $\hat{\boldsymbol{x}}_{0|0}$, $\boldsymbol{P}_{0|0} = \boldsymbol{I}_{6\times6}$
2: $\boldsymbol{u}_k = \bar{\boldsymbol{u}}_d(:,1)$
3: $\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{A}_d\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{B}_d\boldsymbol{u}_k$
4: $\boldsymbol{P}_{k+1|k} = \boldsymbol{A}_d\boldsymbol{P}_{k|k}\boldsymbol{A}_d^\top + \boldsymbol{Q}_d$
5: $\boldsymbol{K}(k+1) = \boldsymbol{P}_{k+1|k}\boldsymbol{H}^\top[\boldsymbol{H}\boldsymbol{P}_{k+1|k}\boldsymbol{H}^\top + \boldsymbol{\Omega}_d]^{-1}$
6: **if** $t/T \neq$ integer **then**
7: $\quad K(1:2,:) = \boldsymbol{0}_{2\times4}$
8: **end if**
9: $\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{K}(k+1)[\hat{\boldsymbol{z}}_{k+1} - \boldsymbol{H}\hat{\boldsymbol{x}}_{k+1|k}]$
10: $\boldsymbol{P}_{k+1|k+1} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_{k+1|k}$
    return $\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}, \hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{P}_{k+1|k}$,

---

RTS smoother normally runs once. For a real-time application, however, the algorithm is insufficient. Thus, to ensure an online application, a window function is employed to accomplish the backward pass for each step, where N_s is the length of the moving window. The procedure is summarized in Algorithm 7 and 8. Due to the additional phase shift of the online smoothing, the estimate is now further delayed by $t - \tau_d - \tau_s$.

Algorithm 7 Shift register #4.

Shift register#4 $(\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}, \hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{P}_{k+1|k})$

1: $\bar{\boldsymbol{x}} = [\bar{\boldsymbol{x}}_{k|k}(:, 2:N_s), \hat{\boldsymbol{x}}_{k|k}]$
2: $\bar{\boldsymbol{P}} = [\bar{\boldsymbol{P}}_{k|k}(:,:, 2:N_s), \boldsymbol{P}_{k|k}]$
3: $\bar{\boldsymbol{x}}_+ = [\bar{\boldsymbol{x}}_{k+1|k}(:, 2:N_s), \hat{\boldsymbol{x}}_{k+1|k}]$
4: $\bar{\boldsymbol{P}}_+ = [\bar{\boldsymbol{P}}_{k+1|k}(:,:, 2:N_s), \boldsymbol{P}_{k+1|k}]$
    return $\bar{\boldsymbol{x}}, \bar{\boldsymbol{P}}, \bar{\boldsymbol{x}}_+, \bar{\boldsymbol{P}}_+$

4.3.5. Predictor

To move the estimated states forward to the present time t, a predictor is proposed. The time delay $\tau_d + \tau_s$ is still small; thus, the integration drift caused by the accelerometer bias is not significant until the next GPS correction. During the time delay, all acceleration inputs to Eq. (10a) are known. Therefore, the present estimate is achieved by updating Eq. (10a) for $N_d + N_s$ steps with the input vectors $\bar{\boldsymbol{u}}_d$ and $\bar{\boldsymbol{u}}_s$. The process is illustrated in Algorithm 9.

5. Verification of the proposed algorithms

5.1. Simulation overview

The simulation is conducted with HAWC2 and MATLAB [46]. The model mentioned in Section 2.2 is adopted in the following simulations. The monopile

---

Algorithm 8 Online RTS smoothing

Online RTS smoothing($\bar{\boldsymbol{x}}, \bar{\boldsymbol{P}}, \bar{\boldsymbol{x}}_+, \bar{\boldsymbol{P}}_+, \boldsymbol{u}_k$)

Initialize: $\hat{\boldsymbol{x}}_{N_s|N_s} = \bar{\boldsymbol{x}}(:, N_s)$, $\boldsymbol{P}_{N_s|N_s} = \bar{\boldsymbol{x}}(:, :, N_s)$

1: $\bar{\boldsymbol{u}}_s = [\bar{\boldsymbol{u}}_s(:, 2 : N_s), \bar{\boldsymbol{u}}_d(:, 1)]$

2: **if** $k \leq N_s$ **then**

3: $\quad \hat{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}(:, 1)$

4: **else**

5: $\quad$ **for** $j = N_s - 1 : -1 : 0$ **do**

6: $\quad\quad \hat{\boldsymbol{x}}_{j|j} = \bar{\boldsymbol{x}}(:, j), \hat{\boldsymbol{x}}_{j+1|j} = \bar{\boldsymbol{x}}_+(:, j), \boldsymbol{P}_{j|j} = \bar{\boldsymbol{P}}(:, :, j), \boldsymbol{P}_{j+1|j} = \bar{\boldsymbol{P}}_+(:, :, j)$

7: $\quad\quad \boldsymbol{A}(j) = \boldsymbol{P}_{j|j} \boldsymbol{A}_d^\top \boldsymbol{P}_{j+1|j}^{-1}$

8: $\quad\quad \hat{\boldsymbol{x}}_{j|N_s} = \hat{\boldsymbol{x}}_{j|j} + \boldsymbol{A}(j)[\hat{\boldsymbol{x}}_{j+1|N_s} - \hat{\boldsymbol{x}}_{j+1|k}]$

9: $\quad\quad \boldsymbol{P}_{j|N_s} = P_{k|k} + \boldsymbol{A}(k)[\boldsymbol{P}_{k+1|N_s} - \boldsymbol{P}_{k+1|k}]\boldsymbol{A}^\top(k)$

10: $\quad$ **end for**

11: $\quad \hat{\boldsymbol{x}}_k(t - \tau_d - \tau_s) = \hat{\boldsymbol{x}}_{0|N_s}$

12: **end if**

return $\hat{\boldsymbol{x}}_k(t - \tau_d - \tau_s), \bar{\boldsymbol{u}}_s$

Algorithm 9 Predictor

Online RTS smoothing($\hat{\boldsymbol{x}}_k(t - \tau_d - \tau_s), \bar{\boldsymbol{u}}_d, \bar{\boldsymbol{u}}_s, N_d, N_s$)

1: **for** $i = 1 : N_d - 1$ **do**

2: $\quad \hat{\boldsymbol{x}}_k = \boldsymbol{A}_d \hat{\boldsymbol{x}}_k + \boldsymbol{B}_d \bar{\boldsymbol{u}}_d(:, i)$

3: **end for**

4: **for** $i = 1 : N_s$ **do**

5: $\quad \hat{\boldsymbol{x}}_k = \boldsymbol{A}_d \hat{\boldsymbol{x}}_k + \boldsymbol{B}_d \bar{\boldsymbol{u}}_s(:, i)$

6: **end for**

return $\hat{\boldsymbol{x}}_k(t)$

---

is exposed to irregular waves and a turbulent wind field. The environmental parameters are listed in Table 2. HAWC2 outputs high-frequency position, velocity, and acceleration at the hub, which are considered as the real values in the simulations. After adding zero-mean white noise processes and a slow-varying bias, and resampling with specific sample rates with delays, the resulting position and acceleration signals are used as measurements from GNSS and IMU. The important sensor parameters are listed in Table 5. The time step T_a is selected as 10 ms. The biases on the IMU specific force measurements are simulated as Wiener processes.

To evaluate of the performance of the state estimator, the mean square error (MSE) is introduced as criterion as follows:

$$MSE := \frac{1}{N} \sum_{k=1}^{N} (x_k - \hat{x}_k)^2, \tag{12}$$

where N is the total number of steps used in the estimate.

Table 5: Sensor parameters in the simulations.

Parameter	Value	Unit
GPS accuracy	± 0.03	m
GPS sampling rate	1	Hz
GPS delay	0.2	s
Accelerometer accuracy	± 0.15	m/s^2
Accelerometer sampling rate	100	Hz

5.2. Simulation results

The simulation results are presented in figures 9–14. The double integration of the acceleration measurement drifts off significantly due to the IMU bias, as expected. The performance of HME1 and HME2 are similar. Both HME1 and HME2 provide a high-frequency real-time motion estimation. Although the GPS sampling rate of 1 Hz is quite low, the MKF accurately estimates the position and velocity of the hub, including the sensor bias. The MKF suppresses the GPS measurement noise, and the estimate approaches the real state. The online smoother smooths the trajectory in the case of sudden changes which may cause high derivatives. The predictor corrects the delay between the smoothed estimate and the real values. The proposed GNSS/IMU integration scheme greatly improves the GPS measurement in terms of both the accuracy and sampling rate. As shown in figures 11–13, the velocity and acceleration estimates converge to the real values. The error of the velocity estimate is within 0.05 m/s; see Figure 12.

From the accumulated absolute errors (Figure 9 and Figure 11), HME1 has a better position estimation, whereas the velocity and bias estimates from HME2 are slightly better. In the case study, the monopile natural period is 3.8 seconds,

for which the GPS sampling interval is too large. Hence, the moving horizon estimator struggles to predict precisely according to the sampled data. In summary, in this scenario, HME1 is preferred because of its lower complexity and sufficient accuracy.
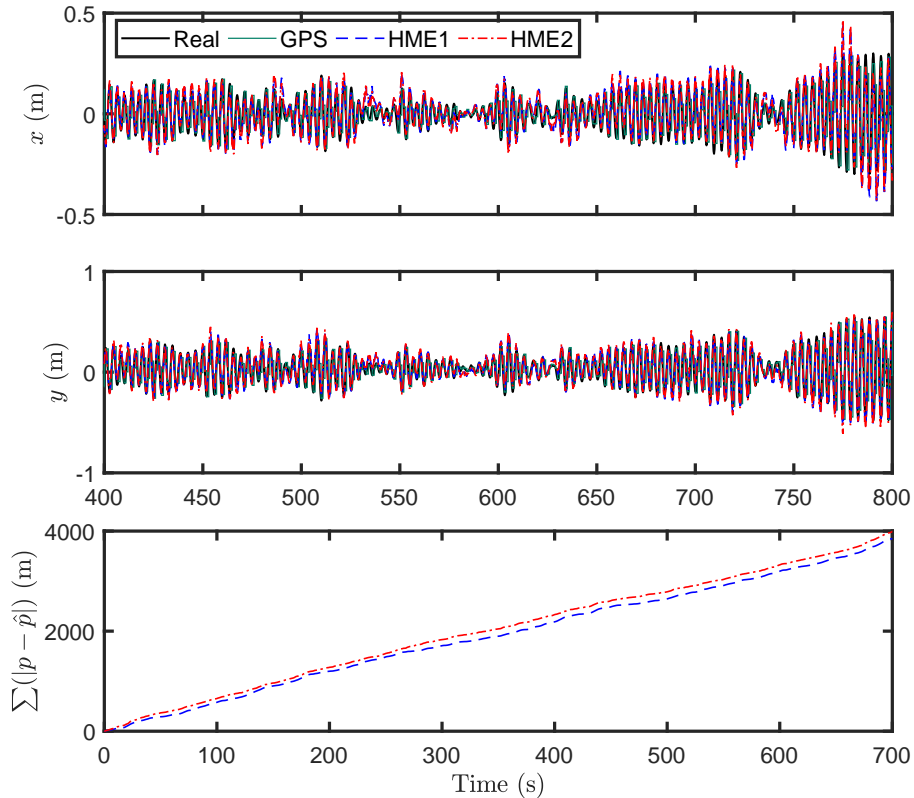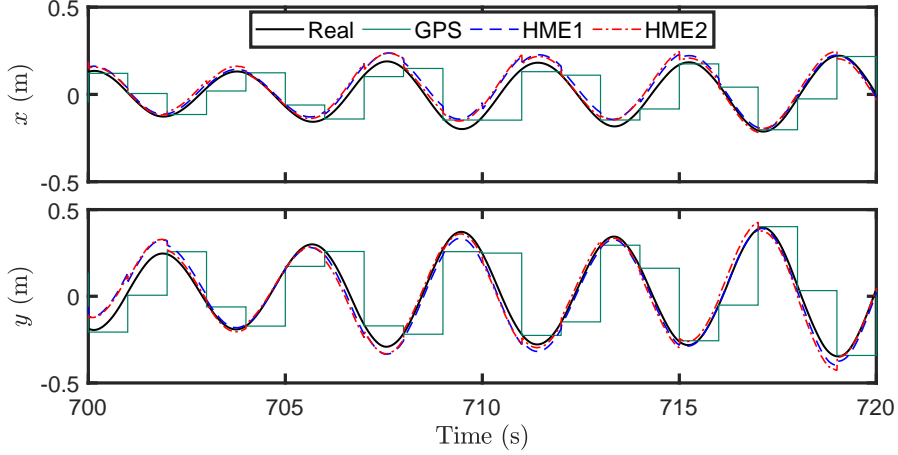

Figure 9: Position estimate, $H_s = 2$ m, $T_p = 6$ s, $\beta_{wave} = 30°$, $S_{wave} = 10$, $U_w = 8$ m/s, $TI = 0.146$, and $S_{TI} = 596$.

5.3. Sensitivity study on the oscillation amplitude

In Section 5.2, the simulation results show that the estimation performance of HME2 is not improved by MHE. In this section, a sensitivity study is conducted to show how the environmental conditions influence the performance of HME1 and HME2. A series of simulations are conducted. The time series generated by the fitted spectra (Eq. 1) in Table 4 are selected as standard measurements and inputs with an oscillation amplitude less than 1 m. The generated time series is multiplied by a coefficient $N_a \in \{0.05, 0.1, 0.2, 0.5, 0.7, 1, 2, 3, 7, 10\}$ to simulate the oscillations with distinct amplitudes. All the other parameters, e.g., the GPS and IMU measurement noises, remain the same in all the simulations. The incoming direction of the wave is $30°$, i.e., $\beta_{wave} = 30°$. To analyze

Figure 10: Position estimate, zoomed in, $H_s = 2$ m, $T_p = 6$ s, $\beta_{wave} = 30°$, $S_{wave} = 10$, $U_w = 8$ m/s, $TI = 0.146$, and $S_{TI} = 596$.

the performance of the HME1 and HME2 algorithms, we also implement an EKF and an MKF for comparison. The EKF is implemented based on the plant model (6), where the innovation from the GPS signal is set to zero when the GPS signal is not available. Similarly, the MKF is implemented based on the delayed GPS signals.

Taking the natural periods 4, 6, and 8 seconds as examples, the influence of the oscillation amplitudes on the various estimation schemes are shown in figures 15–17. It is easy to notice that the MSEs of both position and velocity estimates increase with the motion amplitude. The EKF shows the best performance when the oscillation amplitude is less than 0.1 m, i.e., when the motion estimator is not necessary. However, the EKF fails to accurately monitor the motion when $N_a \geq 0.1$. Use of only the MKF (Algorithm 6) without shift register #1 (Algorithm 3) yields medium performance results as compared with HME1 and HME2. Because of the GPS delay, there is always a delay between its estimate and the real value. With the correction by the predictor, the delay increases when the MKF updates with the delayed IMU specific force measurement from shift register #1. Both HME1 and HME2 estimate the velocity well.

For an OWT with a first fore-aft natural period 4 s, HME2 estimates the position better when the oscillation amplitude is less than 0.3 m, whereas HME1 shows better performance when the amplitude is larger than 0.3 m; see Figure15. For an OWT with a 6-second natural period, the performance boundary is at an amplitude of 1 m; see Figure 16. Normally, the hub motion amplitude is located in the yellow envelope. Hence, both the HME1 and HME2 will improve the estimation performance over a single MKF.

Applying HME2 when the oscillation amplitude is less than the specific amplitude boundary, and adopting HME1 for the cases with oscillation amplitudes
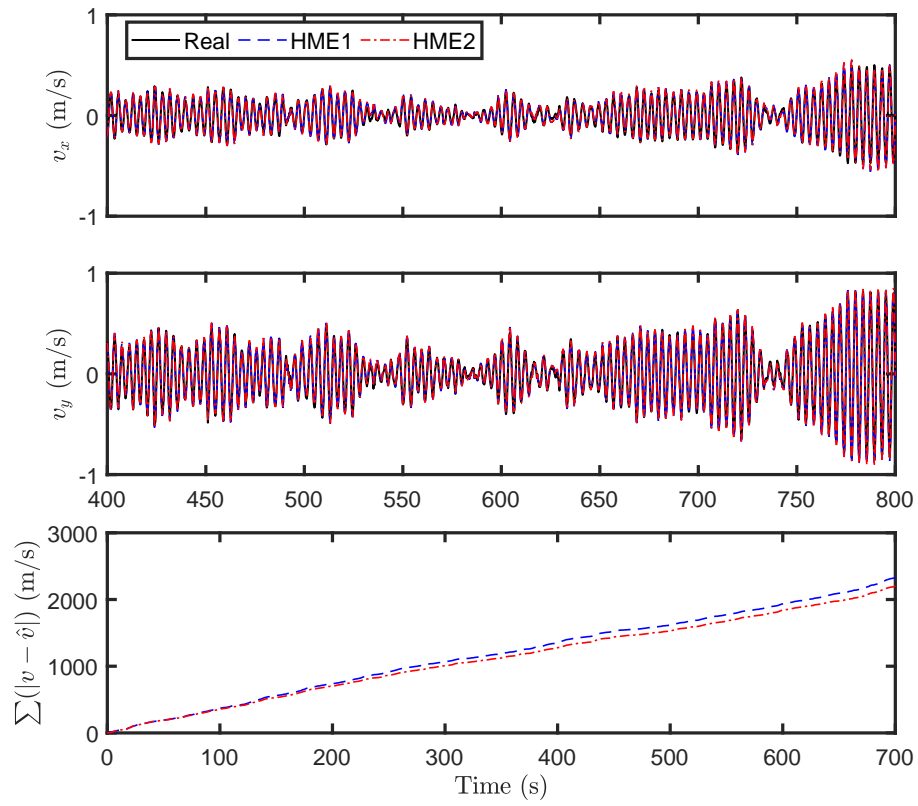
Figure 11: Velocity estimate, $H_s = 2$ m, $T_p = 6$ s, $\beta_{wave} = 30°$, $S_{wave} = 10$, $U_w = 8$ m/s, $TI = 0.146$, and $S_{TI} = 596$.
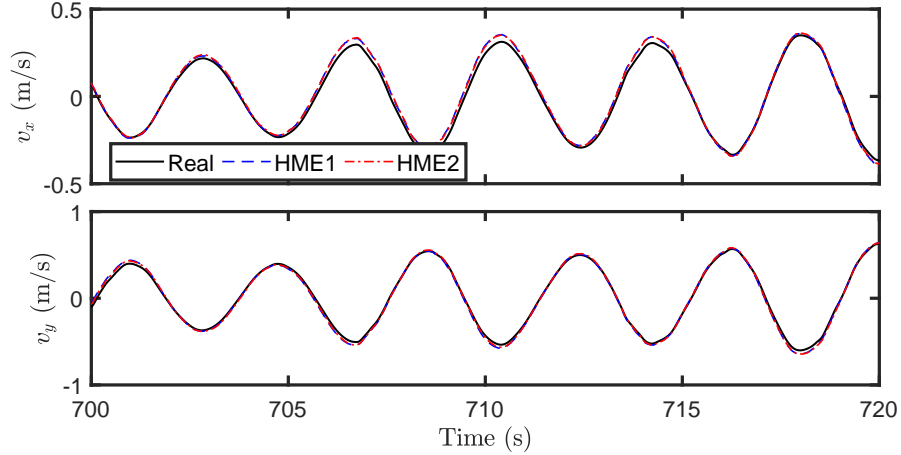
Figure 12: Velocity estimate, zoomed in, $H_s = 2$ m, $T_p = 6$ s, $\beta_{wave} = 30°$, $S_{wave} = 10$, $U_w = 8$ m/s, $TI = 0.146$, and $S_{TI} = 596$.

larger than the boundary, appear to achieve the best estimation performance.

5.4. Sensitivity study on the natural period and oscillation amplitude

Allowing the natural period to be from 3 to 8 seconds, the results of the best algorithms are illustrated in Figure 18, using the criterion of minimum MSE. The yellow area shows the normal envelope of oscillation amplitude and natural period for an OWT. The detailed MSE results in three example conditions are presented in tables 6–9. HME1 provides the best position estimate at upper left corner in Figure 18, while HME2 is the best choice, shown in the lower right corner. MKF is the best scheme to use at two points ([0.5 m, 4 s] and [0.5 m, 5 s]), where the performance of MKF, HME1, and HME2 are very similar; see Figure 15 The velocity estimate by HME2 is slightly better than the one from HME1. The performance of HME2 improves with an increasing natural period of the hub and with a decreasing oscillation amplitude. For the conditions in tables 6–8, HME2 improves the estimate by approximately 20–30%. However, the estimate from HME1 is more accurate when the natural period is small; see Table 9.

Because of the low sampling rate of the GPS and the relatively high hub oscillation frequency, the moving horizon estimator struggles to accurately estimate using the sampled GPS signal and specific force measurements when the natural period of the OWT is low. HME2 greatly improves the estimate in the low-amplitude and long-natural-period region, whereas HME1 performs better in the region with higher motion amplitudes. Therefore, the suitable scheme can be decided by the oscillation amplitude and frequency.
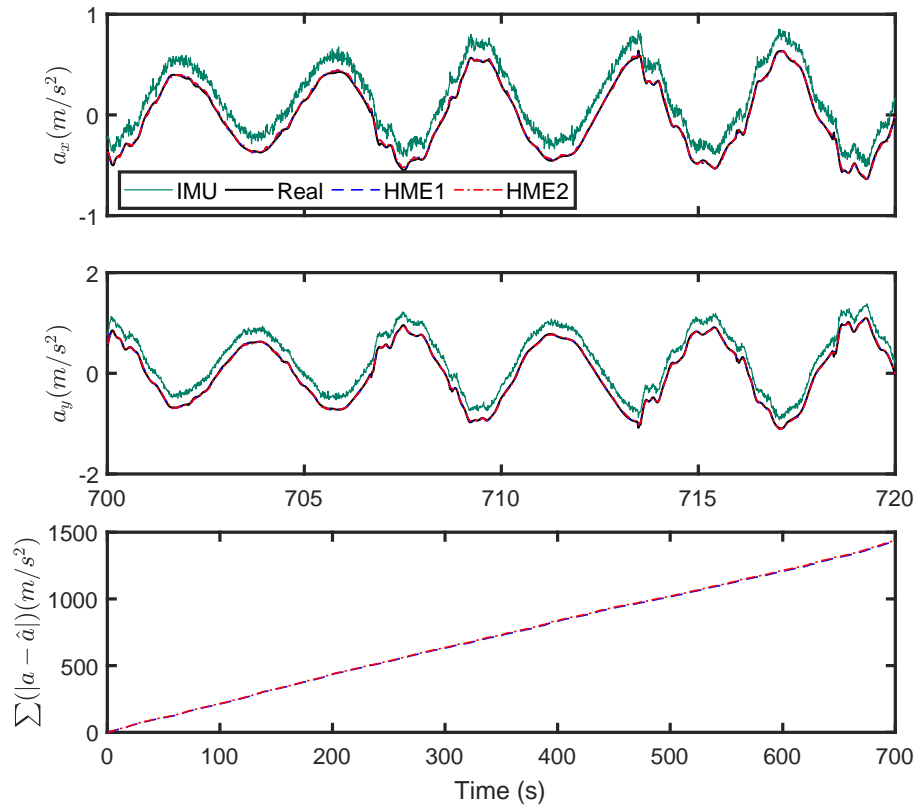
24

Figure 13: Acceleration estimate, $H_s = 2$ m, $T_p = 6$ s, $\beta_{wave} = 30°$, $TI = 0.146$, $S_{wave} = 10$, $U_w = 8$ m/s, and $S_{TI} = 596$.
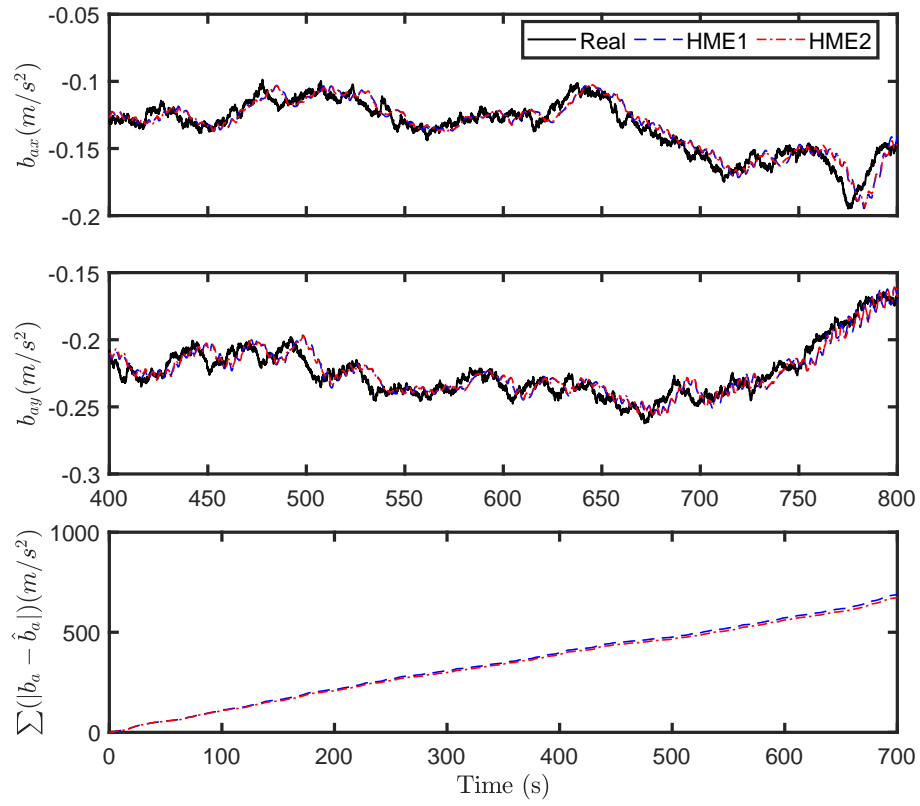
Figure 14: Accelerometer bias estimate, $H_s = 2$ m, $T_p = 6$ s, $\beta_{wave} = 30°$, $TI = 0.146$, $S_{wave} = 10$, $U_w = 8$ m/s, and $S_{TI} = 596$.
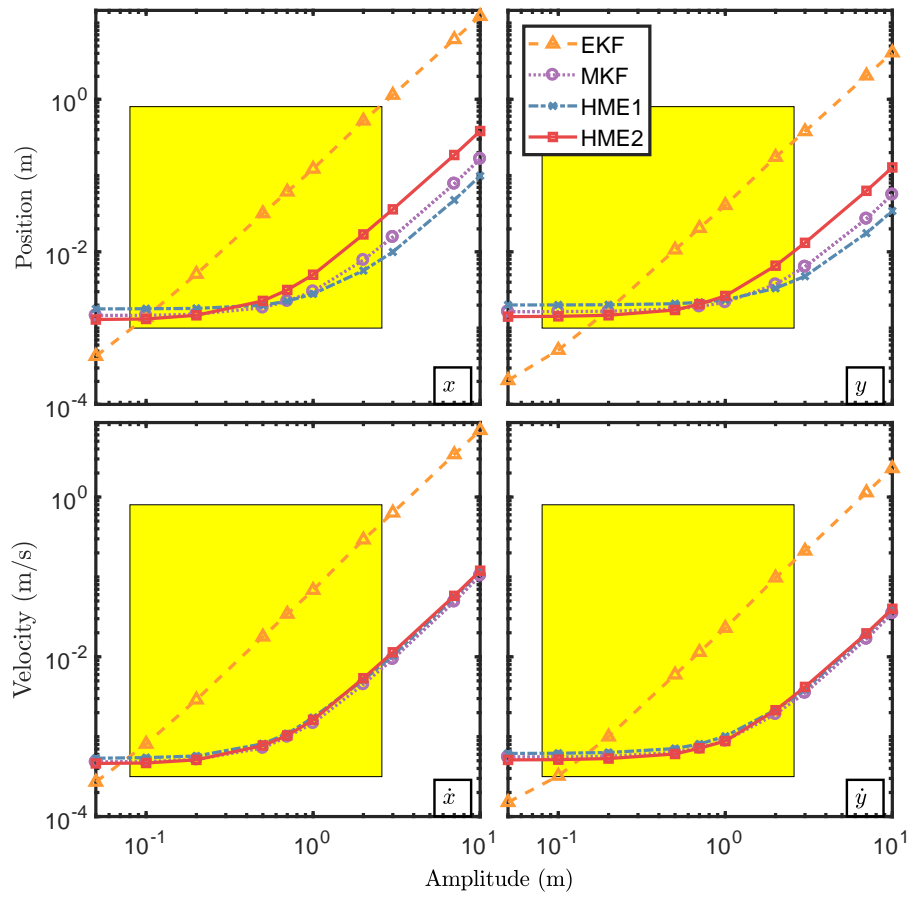
Figure 15: MSE of hub motion estimates using different methods in various hub motion amplitudes, natural period 4 seconds.
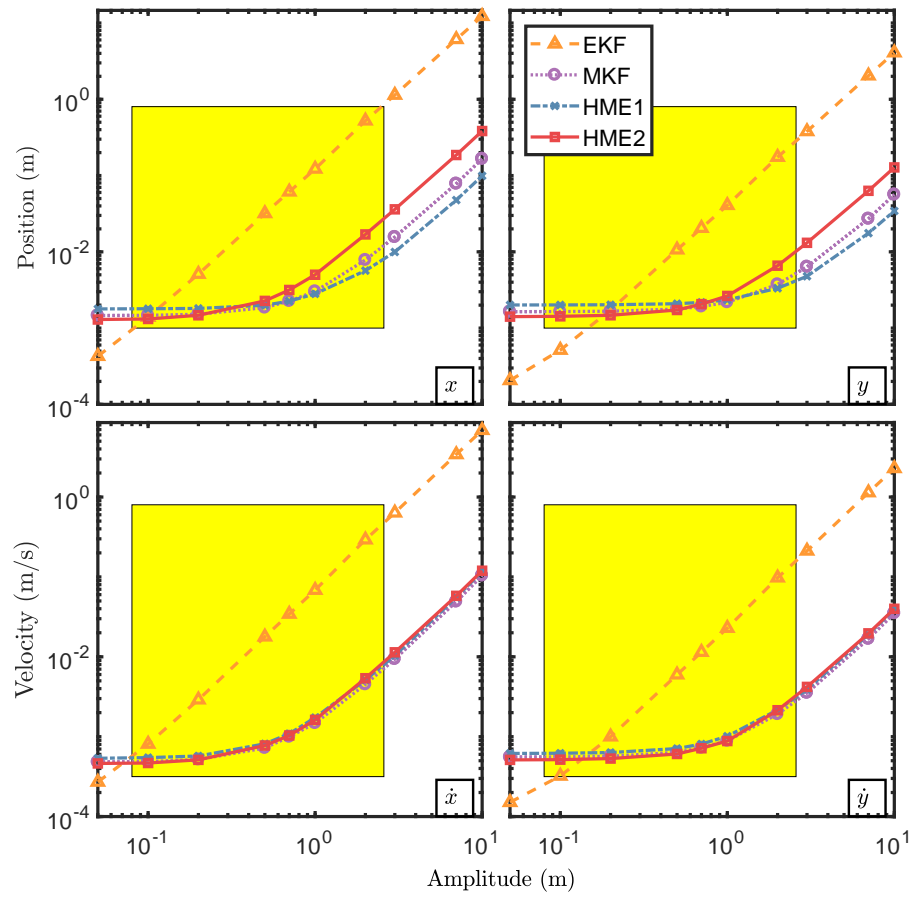
Figure 16: MSE of hub motion estimates using different methods in various hub motion amplitudes, natural period 6 seconds.
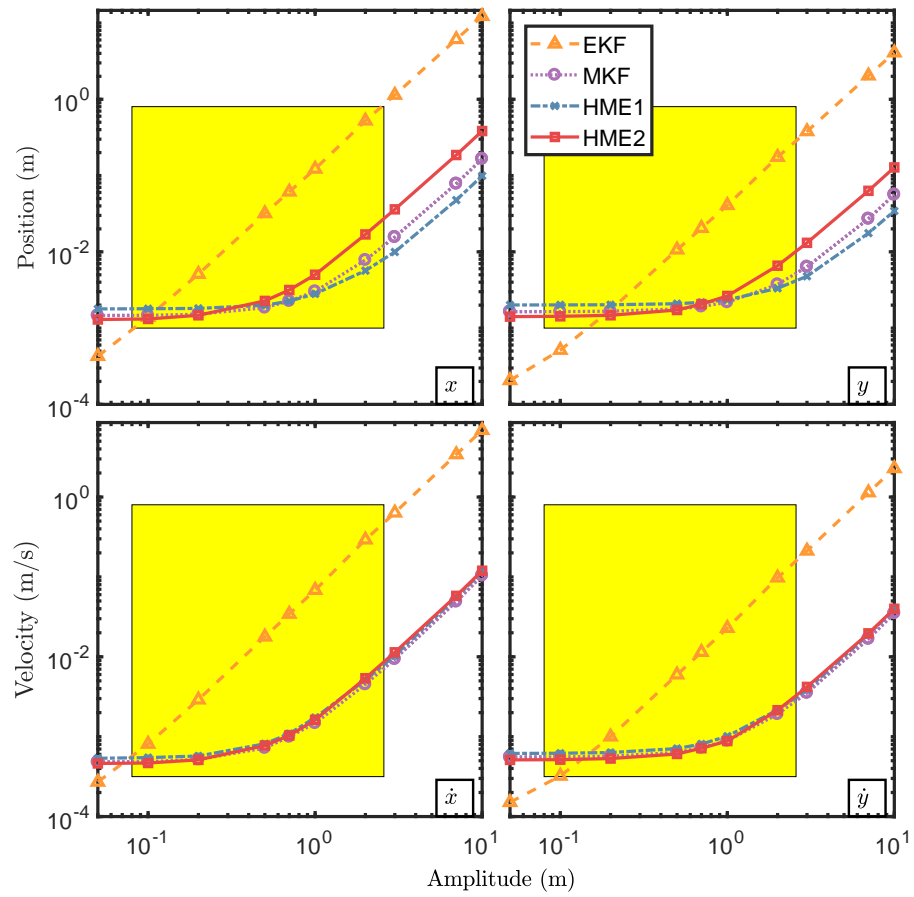
Figure 17: MSE of hub motion estimates using different methods in various hub motion amplitudes, natural period 8 seconds.

Table 6: The MSE of different algorithms, monopile natural period 4 seconds, oscillation amplitude 0.2 m.

Natural period (s)	x	y	v_x	v_y
EKF	5.98e-3	2.07e-3	3.53e-3	1.25e-3
MKF	1.52e-3	1.66e-3	0.52e-3	0.57e-3
HME1	1.79e-3	2.02e-3	0.57e-3	0.63e-3
HME2	1.42e-3	1.46e-3	0.49e-3	0.52e-3

Table 7: The MSE of different algorithms, monopile natural period 8 seconds, oscillation amplitude 0.2 m.

Natural period (s)	x	y	v_x	v_y
EKF	3.42e-3	1.25e-3	1.98e-2	0.74e-3
MKF	1.63e-3	1.71e-3	0.56e-3	0.58e-3
HME1	1.88e-3	2.05e-3	0.59e-3	0.64e-3
HME2	1.38e-3	1.45e-3	0.49e-3	0.52e-3

Table 8: The MSE of different algorithms, monopile natural period 8 seconds, oscillation amplitude 1 m.

Natural period (s)	x	y	v_x	v_y
EKF	8.46e-2	2.82e-2	4.78e-2	1.59e-2
MKF	7.81e-3	3.89e-3	3.25e-3	1.50e-3
HME1	6.59e-3	3.68e-3	2.81e-3	1.39e-3
HME2	4.99e-3	2.60e-3	1.73e-3	0.92e-3

Table 9: The MSE of different algorithms, monopile natural period 3 seconds, oscillation amplitude 2 m.

Natural period (s)	x	y	v_x	v_y
EKF	5.23e-1	1.74e-1	2.92e-1	0.98e-1
MKF	8.61e-3	4.05e-3	4.06e-3	1.75e-3
HME1	2.25e-3	2.17e-3	2.89e-3	1.40e-3
HME2	13.61e-3	5.52e-3	2.96e-3	1.34e-3

6. Conclusions

Due to the slow GNSS sampling interval and GNSS delay, traditional discrete methods fail to provide good real-time motion estimations for a system
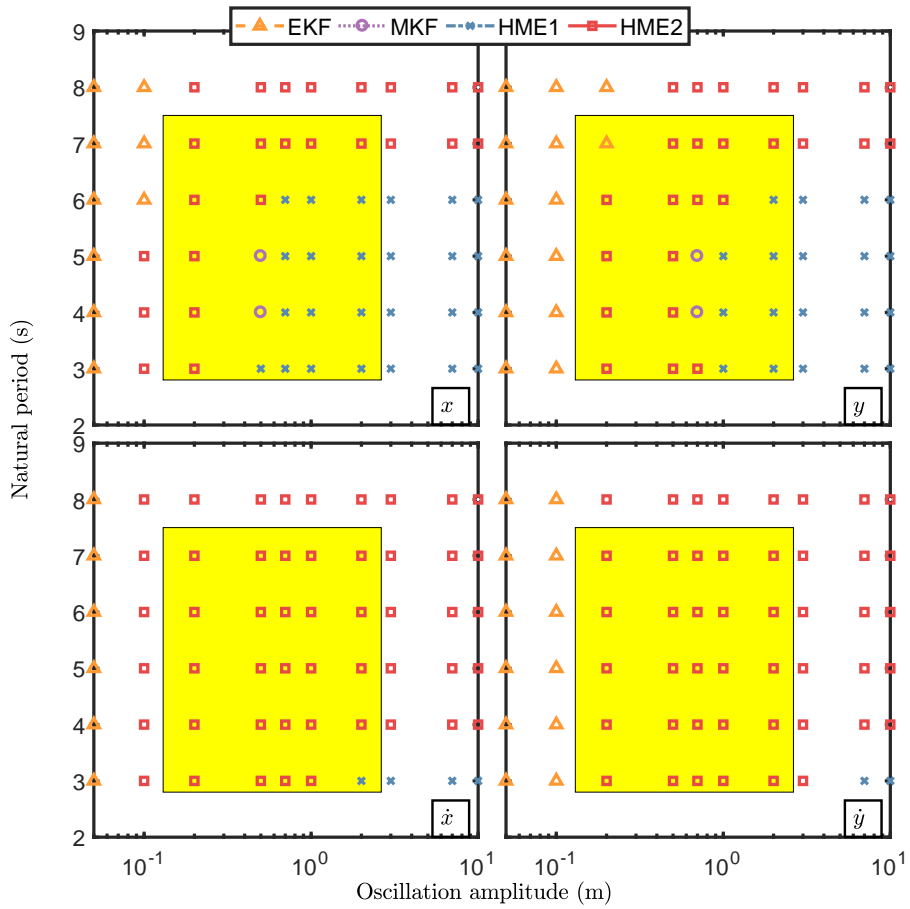
Figure 18: The best schemes with different oscillation amplitudes and natural periods.

with higher frequency. This paper proposes two GNSS/IMU integration schemes to estimate the offshore wind turbine hub position with a bottom-fixed foundation during the mating process. The systems include a multirate Kalman filter, an online RTS smoother, a moving horizon estimator, and a predictor. Simulations have verified that in the fast system, the estimators are sufficiently accurate. The estimates agree well with the real values for position, velocity, acceleration, and bias. The state observer filters the noise in the GPS and the IMU, overcomes the GPS delay, and achieves high-frequency displacement and velocity monitoring functionalities in real time.

Based on a comparison with the EKF and the MKF, the HME1 algorithm performs significantly better used as a real-time hub motion estimator during blade installation under some relevant conditions, while HME2 performs significantly better under some other relevant conditions. The performance of the state estimators is affected by the oscillation amplitude and the natural period of the hub structure. HME2 works well in the low-amplitude and low-frequency region, whereas HME1 shows superior performance in the region with higher motion amplitudes. Hence, both estimators could be implemented with a supervisory switching logic [47]. Both algorithms provide a sufficiently accurate estimate of the hub motion states in relevant conditions, thereby providing improved hub monitoring and increased safety and efficiency in practical applications.

Acknowledgments

References

[1] Z. Jiang, W. Hu, W. Dong, Z. Gao, Z. Ren, Structural reliability analysis of wind turbines: a review, Energies 10 (2017) 2099.

[2] P. Murtagh, A. Ghosh, B. Basu, B. Broderick, Passive control of wind turbine vibrations including blade/tower interaction and rotationally sampled turbulence, Wind Energy 11 (4) (2008) 305–317.

[3] T. Yang, X. Yang, Y. Li, B. Fang, Passive and adaptive vibration suppression of pipes conveying fluid with variable velocity, Journal of Vibration and Control 20 (9) (2014) 1293–1300.

[4] T. J. Larsen, T. D. Hanson, A method to avoid negative damped low frequent tower vibrations for a floating, pitch controlled wind turbine, Journal of Physics: Conference Series 75 (1) (2007) 012073.

[5] J. Li, Z. Zhang, J. Chen, Experimental study on vibration control of offshore wind turbines using a ball vibration absorber, Energy and Power Engineering 4 (03) (2012) 153.

[6] T. J. Stehly, D. M. Heimiller, G. N. Scott, 2016 cost of wind energy review, Tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States) (2017).

[7] M. Gaunaa, J. Heinz, W. Skrzypiński, Toward an engineering model for the aerodynamic forces acting on wind turbine blades in quasisteady standstill and blade installation situations, Journal of Physics: Conference Series 753 (2) (2016) 022007.

[8] Z. Jiang, Z. Gao, Z. Ren, Y. Li, L. Duan, A parametric study on the final blade installation process for monopile wind turbines under rough environmental conditions, Engineering Structures 172 (2018) 1042–1056.

[9] P. Zhang, Y. Han, H. Ding, S. Zhang, Field experiments on wet tows of an integrated transportation and installation vessel with two bucket foundations for offshore wind turbines, Ocean Engineering 108 (2015) 769–777.

[10] A. S. Verma, Z. Jiang, N. P. Vedvikc, Z. Gao, Z. Ren, Impact assessment of blade root with a hub during the mating process of an offshore monopile-type wind turbine, Engineering StructuresAccepted.

[11] Z. Ren, Z. Jiang, R. Skjetne, Z. Gao, An active tugger line force control method for single blade installations, Wind Energy 21 (2018) 1344–1358.

[12] W. Yang, P. J. Tavner, C. J. Crabtree, Y. Feng, Y. Qiu, Wind turbine condition monitoring: technical and commercial challenges, Wind Energy 17 (5) (2014) 673–693.

[13] B. Lu, Y. Li, X. Wu, Z. Yang, A review of recent advances in wind turbine condition monitoring and fault diagnosis, in: Power Electronics and Machines in Wind Applications, 2009. PEMWA 2009. IEEE, IEEE, 2009, pp. 1–7.

[14] T. I. Fossen, Handbook of marine craft hydrodynamics and motion control, John Wiley & Sons, 2011.

[15] T. H. Bryne, J. M. Hansen, R. H. Rogne, N. Sokolova, T. I. Fossen, T. A. Johansen, Nonlinear observers for integrated INS/GNSS navigation: Implementation aspects, IEEE Control Systems Magazine 37 (3) (2017) 59–86.

[16] W. Shi, H.-C. Park, J.-H. Baek, C.-W. Kim, Y.-C. Kim, H.-K. Shin, Study on the marine growth effect on the dynamic response of offshore wind turbines, International Journal of Precision Engineering and Manufacturing 13 (7) (2012) 1167–1176.

[17] A. Noureldin, T. B. Karamat, J. Georgy, INS/GPS Integration, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 247–271.

[18] R. Mahony, T. Hamel, J.-M. Pflimlin, Nonlinear complementary filters on the special orthogonal group, IEEE Transactions on automatic control 53 (5) (2008) 1203–1218.

[19] R. G. Brown, P. Y. Hwang, Introduction to random signals and applied kalman filtering: with matlab exercises and solutions, Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions, by Brown, Robert Grover.; Hwang, Patrick YC New York: Wiley, c1997.

[20] M. Zhu, A. Hahn, Y.-Q. Wen, Identification-based controller design using cloud model for course-keeping of ships in waves, Engineering Applications of Artificial Intelligence 75 (2018) 22–35.

[21] T. A. Johansen, T. I. Fossen, The exogenous Kalman filter (XKF), International Journal of Control 90 (2) (2017) 161–167.

[22] F. Caron, E. Duflos, D. Pomorski, P. Vanheeghe, Gps/imu data fusion using multisensor kalman filtering: introduction of contextual aspects, Information fusion 7 (2) (2006) 221–230.

[23] A. Smyth, M. Wu, Multi-rate kalman filtering for the data fusion of displacement and acceleration response measurements in dynamic system monitoring, Mechanical Systems and Signal Processing 21 (2) (2007) 706–723.

[24] Y. Bock, D. Melgar, B. W. Crowell, Real-time strong-motion broadband displacements from collocated gps and accelerometers, Bulletin of the Seismological Society of America 101 (6) (2011) 2904–2925.

[25] H. Lan, C. Yu, N. El-Sheimy, An integrated pdr/gnss pedestrian navigation system, in: China Satellite Navigation Conference (CSNC) 2015 Proceedings: Volume III, Springer, 2015, pp. 677–690.

[26] B. Zhao, R. Skjetne, M. Blanke, F. Dukan, Particle filter for fault diagnosis and robust navigation of underwater robot, IEEE Transactions on Control Systems Technology 22 (6) (2014) 2399–2407.

[27] T. H. Bryne, R. H. Rogne, T. I. Fossen, T. A. Johansen, A virtual vertical reference concept for aided inertial navigation at the sea surface, Control Engineering Practice 70 (2018) 1 – 14. `doi:https://doi.org/10.1016/j.conengprac.2017.09.009`.

[28] H. F. Grip, T. I. Fossen, T. A. Johansen, A. Saberi, Globally exponentially stable attitude and gyro bias estimation with application to GNSS/INS integration, Automatica 51 (2015) 158–166.

[29] J. M. Hansen, T. I. Fossen, T. A. Johansen, Nonlinear observer design for GNSS-aided inertial navigation systems with time-delayed GNSS measurements, Control Engineering Practice 60 (2017) 39–50.

[30] T. J. Larsen, A. M. Hansen, How 2 HAWC2, the user's manual, Tech. rep., Risø National Laboratory (2007).

[31] W. Shi, H. C. Park, C. W. Chung, H. K. Shin, S. H. Kim, S. S. Lee, C. W. Kim, Soil-structure interaction on the response of jacket-type offshore wind turbine, International Journal of Precision Engineering and Manufacturing-Green Technology 2 (2) (2015) 139–148.

[32] P. Zhang, W. Wei, N. Jia, H. Ding, R. Liu, Effect of seepage on the penetration resistance of bucket foundations with bulkheads for offshore wind turbines in sand, Ocean Engineering 156 (2018) 82–92.

[33] H. Matlock, L. C. Reese, Generalized solutions for laterally loaded piles, Journal of the Soil Mechanics and foundations Division 86 (5) (1960) 63–94.

[34] American Petroleum Institute, API RP 2A-WSD Recommended Practice for Planning, Designing and Constructing Fixed Offshore Platforms: Working Stress Design, American Petroleum Institute, 2000.

[35] S. Bhattacharya, S. Adhikari, Experimental validation of soil–structure interaction of offshore wind turbines, Soil Dynamics and Earthquake Engineering 31 (5) (2011) 805–816.

[36] T. Yang, Y. Tang, Q. Li, X.-D. Yang, Nonlinear bending, buckling and vibration of bi-directional functionally graded nanobeams, Composite Structures.

[37] L. Arany, S. Bhattacharya, J. H. Macdonald, S. J. Hogan, Closed form solution of eigen frequency of monopile supported offshore wind turbines in deeper waters incorporating stiffness of substructure and ssi, Soil Dynamics and Earthquake Engineering 83 (2016) 18–32.

[38] J. Jonkman, S. Butterfield, P. Passon, T. Larsen, T. Camp, J. Nichols, J. Azcona, A. Martinez, Offshore code comparison collaboration within IEA wind annex XXIII: phase ii results regarding monopile foundation modeling, Tech. rep., National Renewable Energy Laboratory (NREL), Golden, CO. (2008).

[39] J. Jonkman, W. Musial, Offshore code comparison collaboration (oc3) for iea wind task 23 offshore wind technology and deployment, Tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States) (2010).

[40] IEC 61400-1, Wind turbine generator systems-part 1: Safety requirements, Standard, International Electrotechnical Commission and others, Geneva, Switzerland (Aug. 2005).

[41] M. Zhu, A. Hahn, Y.-Q. Wen, A. Bolles, Identification-based simplified model of large container ships using support vector machines and artificial bee colony algorithm, Applied Ocean Research 68 (2017) 249–261.

[42] W. Deng, R. Yao, H. Zhao, X. Yang, G. Li, A novel intelligent diagnosis method using optimal ls-svm with improved pso algorithm, Soft Computing (2017) 1–18.

[43] W. Deng, S. Zhang, H. Zhao, X. Yang, A novel fault diagnosis method based on integrating empirical wavelet transform and fuzzy entropy for motor bearing, IEEE Access 6 (2018) 35042–35056. `doi:10.1109/ACCESS.2018.2834540`.

[44] P. Batista, C. Silvestre, P. Oliveira, On the observability of linear motion quantities in navigation systems, Systems & Control Letters 60 (2) (2011) 101–110.

[45] H. E. Rauch, F. Tung, C. T. Striebel, et al., Maximum likelihood estimates of linear dynamic systems, AIAA journal 3 (8) (1965) 1445–1450.

[46] Z. Ren, Z. Jiang, R. Skjetne, Z. Gao, Development and application of a simulator for offshore wind turbine blades installation, Ocean Engineering 166 (2018) 380–395.

[47] V. Hassani, A. M. Pascoal, A. J. Srensen, Detection of mooring line failures using dynamic hypothesis testing, Ocean Engineering 159 (2018) 496 – 503. `doi:https://doi.org/10.1016/j.oceaneng.2018.01.021`.