

# Real-time Thrust Optimization for Multiple Wingsails using Extremum Seeking

Alexandre Rocha and Torsten Wik

**Abstract**—In this paper Extremum Seeking Control (ESC) is implemented for thrust optimization in sail-assisted ships. The multivariate extremum seeking of multiple wingsails is addressed. This is a complex problem due to the interactions between wingsails and the time-variant wind profile while sailing. In order to attenuate the interactions between wingsails, Newton-based ESC is implemented and compared with the gradient-based counterpart. The performance of the algorithms is simulated with real measurements of the wind angle during ship experiments in the Baltic sea.

**Index Terms**—extremum seeking, wingsail, sail-assisted ship

## I. INTRODUCTION

Maritime transportation is responsible for 80% of the global trade [1], being a main pillar of the globalized economy of today's world. Due to its high volume of cargo and low unit costs, ships are usually the most cost-efficient means of transportation for goods such as raw materials, fuel, and food. Although this industry was impacted in the short-term by the recent COVID-19 pandemic, maritime trade is predicted to keep growing in the long term in line with the average world economic growth, continuing to be a crucial player in international trade [2].

On the other hand, it is well known that ships are significant contributors of green house gases, accounting for almost 3% of the global anthropogenic emissions, according to [3]. In the same report, it is stated that, by 2050, shipping emissions are expected to rise up to 130% from the value in 2008. Therefore, to tackle the worldwide pollution problem, without inducing economic shocks, it is urgent to develop new technologies to reduce the pollution intensity of the maritime transportation sector.

Recently, sail-assisted ships have been pointed out as a solution to reduce fossil fuel consumption or, equivalently, to decrease shipping traveling time, attaining a more cost-efficient operation. In essence, a sail-assisted ship is an engine-powered ship that uses one or more sails as an additional propulsion source. By combining these two sources of energy - wind and fossil fuels - it is possible to exploit the advantages of each of them, while improving sustainability in overall. In a sail-assisted ship, the system is composed by two main actuators: the rudders and the sails. Usually, the former is used for ship heading control and the latter for speed optimization, where the intricate couplings between both are treated as disturbances. This work assumes that same decoupling, and focuses on the speed optimization problem. From the control

A. Rocha and T. Wik are with the [Division], Chalmers University of Technology, Gothenburg, Sweden. {rochaa, torsten.wik}@chalmers.se.

perspective, this problem is formulated as optimizing the angle of attack of each sail, which is the angle between the center line of the sail and the apparent wind, which is the wind measured on the ship.

Research in speed optimization for sail-assisted ships is still in its early days. Nevertheless, related problems have been addressed over the last two decades, such as the control of autonomous sailboats. The authors in [4] proposed the computation of the optimal sheeting angle as a thrust optimization problem based on a nonlinear dynamic model of a single-sail autonomous sailboat. Similarly, in [5], a model-based controller is obtained through feedback linearization for another autonomous sailboat. Model based approaches, however, often ignore unmodelled complex dynamics, which is problematic for larger systems with more control variables, i.e. with multiple sails. Furthermore, model-based methods rely on sail polar diagrams, which quantify the potential speed for a given wind angle and speed. However, these diagrams are only an approximation, which can constraint the system to achieve true optima [6]. Moreover, the major problem with methods depending on wind speed and angle is that point-wise measurements of those quantities are rather noisy and not informative of the overall wind effect on the sails' surface. Therefore, it is often the case that true optima are not achieved given measurements' low accuracy. For those reasons, the literature has converged to model-free online optimization methods, often attaining better performance with simpler solutions.

One very suitable type of such methods is Extremum Seeking Control (ESC), which has been increasingly used in the last two decades. In essence, the basic scheme of ESC consists of inputting a periodic perturbation to a system and estimate the gradient of a measured criterion, which is then integrated with tunable gain to reach the extremum of the criterion. The paper [7] presented a simpler form of ESC, a hill climbing controller, for online sheeting angle tuning, which was successfully tested in an single-sail experimental platform. For a similar application, the authors in [8] implemented an ESC with a periodic dither instead. The results were compared with the work in [7] and showed improvements in robustness and convergence rate, mainly due to the use of the gradient's magnitude and a periodic dither, which is not the case in [7]. Another conventional ESC was implemented in [9] for an autonomous sailboat. This paper proposed an additional feature of steady-state oscillations cancellation, which consists of considering the amplitude of the periodic perturbation as a function of the gradient magnitude. The same authors presented later in [10] a sliding mode ESC, which proved to

perform better than the ESC designed in [9]. Nevertheless, these works focus on the control of a single conventional sail. The use of multiple sails is expected to increase the complexity of the controller and the modelling of the system. Moreover, they all consider the measured speed as optimization criteria, which can not only be a too disturbed measurement for ESC but also, due to inertia, too slow for larger systems.

Finally, [11] addresses the problem of multiple wingsail control for a ship application, with thrust in the bow-direction as criterion to maximize. ESC is used both for online sheeting angle tuning and to train a Fuzzy Logic controller. It is shown that the latter replaces accurately the former, avoiding the presence of dithering, thus increasing performance and reducing actuator wearing. However, the implementation is a univariate control problem, since the control input of the three wingsails is the same.

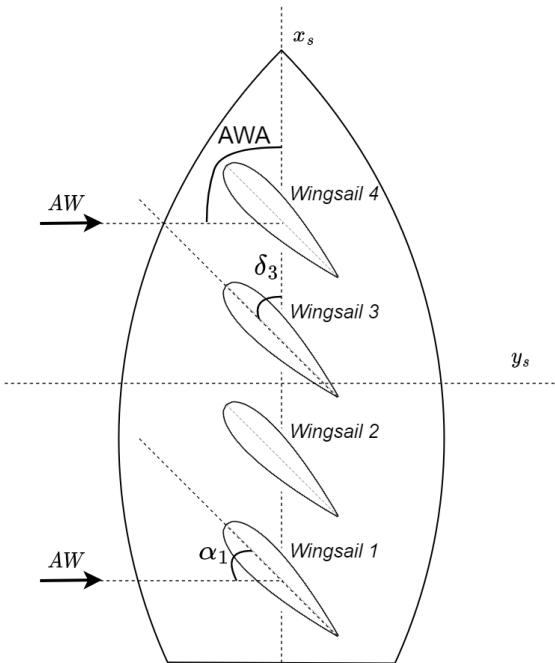


Fig. 1. System scheme.

The main contribution of this paper is the ESC of a sail-assisted ship with four inline, equally-spaced sails, as in Fig. 1. The sails of the ship are rigid wingsails [12], which facilitate modelling and are more suitable for control. The control variables are the sheeting angles of each wingsail. As optimization criterion, instead of speed, this work uses thrust which is the component in the bow direction of the aerodynamic force acting on the wingsails. Given the aforementioned decoupling between heading control and speed optimization, this force is assumed to be directly proportional to the acceleration of the ship, and therefore speed and thrust optimization are equivalent problems. The method is model-free, meaning that the unknown criterion is only measured at each sampling time.

In the basic form of ESC, the controller convergence to the extremum depends on the Hessian of the static-map. In this

work's application, the Hessian is not expected to be diagonal and it is not guaranteed that its diagonal terms are similar in magnitude. In other words, the convergence of each wingsail varies in rate and depends on the behavior of its peers, which is an intricate problem for a four-dimensional input scheme. To tackle this problem, this work evaluates the performance of the Newton-based ESC (NBESC) [13], and considers the traditional ESC as benchmark. As the name suggests, the former innovates in the sense that the convergence rate is user-defined. In NBESC, the Hessian and its inverse are estimated and the latter is defined as a feedback gain to cancel the closed-loop dependence on the Hessian. To avoid numerical singularities of Hessian invertibility, its inverse is estimated asymptotically through a differential Riccati equation.

The rest of the paper is organized as follows. In Section II notation is presented. In section III wingsail aerodynamics are described and the extremum seeking algorithms are described. Section IV discuss the results of simulations and evaluate performance of ESC from the sailing point of view. In the end, Section V state the conclusions and future directions.

## II. NOTATION AND PRELIMINARIES

The apparent wind (AW) is the wind velocity vector relative to the ship  $x$ -axis, as defined in Fig. 1. This vector can be decomposed in magnitude, the apparent wind speed (AWS), and in orientation, the apparent wind angle (AWA). In this paper, only horizontal wind is considered.

A wingsail is a rigid body and the horizontal line between its leading and trailing edge is called chord. The angle of attack  $\alpha$  is the angle between the wingsail chord and the AW.

The wingsail rotates around a fixed mast in the ship, and the angle between the ship  $x$ -axis and the wingsail chord is the sheeting angle  $\delta$ .

Throughout the following sections, each wingsail will be denoted by the index  $i$ , therefore the angles  $\alpha_i$  and  $\delta_i$  are defined individually for each wingsail.

## III. SYSTEM MODELING AND CONTROLLER DESIGN

The decoupling of speed optimization and heading control simplifies system modelling and the former results in a more tractable problem to solve. With this assumption, the aerodynamic forces on the sails are assumed to have no effect on ship pose. Therefore, this section focus on the modelling and control of the wingsails and the generated aerodynamic forces, while ship planar kinematics and dynamics are not studied. Additionally, this section formulates the optimization problem and presents the extremum seeking methods and their design.

### A. Wingsail Modeling

The wingsail is selected as a NACA 0018 profile, as it is illustrated in Fig. 2. It is a symmetric airfoil, meaning that generates a symmetric aerodynamic force for symmetric AWA and  $\alpha$ . This feature is crucial in sailing, in order to optimize speed for different AW vectors.

The aerodynamic force can be decomposed into the perpendicular and parallel components to the AW, the lift ( $L$ ) and the

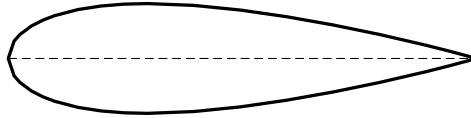


Fig. 2. Wingsail NACA-0080 profile.

drag ( $D$ ), respectively. These are computed for each wingsail as

$$L_i = \frac{1}{2} \rho (\text{AWS})^2 c_L(\alpha_i)$$

$$D_i = \frac{1}{2} \rho (\text{AWS})^2 c_D(\alpha_i) ,$$

where  $\rho$  is the density of air, and  $c_L$  and  $c_D$  are the lift and drag coefficients, respectively, which are dimensionless forces function of  $\alpha_i$ . From  $L$  and  $D$ , the thrust  $T$  is obtained for each wingsail as

$$T_i = L_i \sin(\text{AWA}) - D_i \cos(\text{AWA}) ,$$

and therefore it is a function of AW and  $\alpha$ . Moreover, it is important to highlight that, since ship kinematics are not modeled,  $T$  is not a function of the air resistance in the sail, i.e. the effect of ship velocity on the wingsails is not modeled.

Regarding the dynamics of the wingsail, their time scale is considered as the minimum time for an air particle to go along the wingsail chord. In sailing, for common magnitudes of AWS, this time is by convention defined around  $\tau = 0.1\text{s}$ .

### B. Thrust optimization

The optimization problem here addressed is the maximization of the global thrust  $T$ , which is the sum of the thrusts of each sail,  $T = \sum_i^4 T_i$ . The criterion  $T$  is a function of both  $\alpha$  and AWA. The former is controlled indirectly with  $\delta$ , given  $\delta = \text{AWA} - \alpha$ . Since AWA enters directly as an input of the criterion, and assuming the time-scale of wingsail aerodynamics is noticeably lower than the period of the frequencies, the criterion is defined as a static map. Note however that AWA is an uncontrollable input and therefore it will impact  $T$  in an unpredictable way.

Following the theory on fluid aerodynamics,  $T$  has multiple local optima. Nevertheless, when near the global optima, the criterion is rather smooth and it allows for gradient-based optimization schemes, as it is the case for ESC. This motivates the need of a sufficiently close to optima initial  $\delta$ . In practice, this can be guaranteed with the aid of a feedforward component, which can rely on rough knowledge of AWA measurements.

### C. Gradient-based ESC

Gradient-based ESC (GBESC) is the basic form of ESC, a model-free optimization scheme that is able to maximize an unknown measured criterion by estimating and integrating its gradient [14]. A diagram block of the GBESC algorithm is presented in Fig. 3.

The algorithm goes as follows. Consider the vector of input  $\theta = [\theta_1 \ \theta_2 \ \dots \ \theta_n] \in \mathbb{R}^n$ , and the convex static map

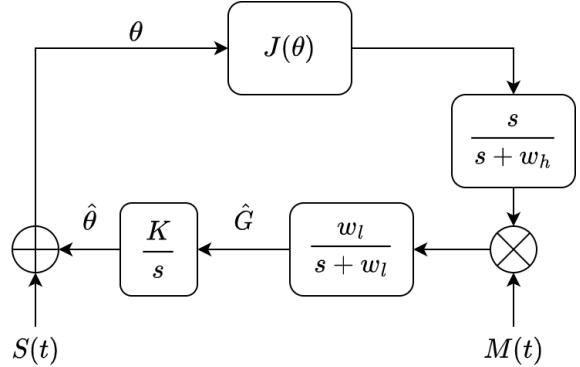


Fig. 3. GBESC scheme.

$J(\theta) \in \mathbb{R}$ , with a local optimum at  $\theta^*$ . The controller creates a perturbation  $S(t)$  in the input, and, from the consequent variation in the output, the gradient is estimated by multiplying the output with the demodulation signal  $M(t)$ . The gradient is then integrated with gain  $K$ , diagonal and positive definite. These signals are defined as follows:

$$S(t) = [a_1 \sin(w_1 t) \ a_2 \sin(w_2 t) \ \dots \ a_n \sin(w_n t)]$$

$$M(t) = [\frac{2}{a_1} \sin(w_1 t) \ \frac{2}{a_2} \sin(w_2 t) \ \dots \ \frac{2}{a_n} \sin(w_n t)] ,$$

where  $\frac{w_i}{w_j}$  are rational and constrained to  $w_i \neq w_j$  and  $w_i + w_j \neq w_k$ , for distinct  $i, j$ , and  $k$ .

Although the high- and low-pass filters in Fig. 3 are not strictly necessary for static maps, they are useful to filter out useless frequency content such as measurements' DC value, measurement noise or high-frequency components resulting from the demodulation operation.

The input error is defined as  $\tilde{\theta} = \hat{\theta} - \theta^*$ , where  $\theta = \hat{\theta} + S(t)$ , and its dynamics are given by  $\dot{\tilde{\theta}} = KM(t)J(\theta^* + S(t) + \tilde{\theta})$ . Considering the second order Taylor expansion of the static map  $J$  around  $\theta^*$ , it is demonstrated in [14] that the averaged closed loop dynamics are

$$\dot{\tilde{\theta}} = -KH\tilde{\theta} ,$$

where  $H$  denotes the Hessian of the static map at  $\theta^*$  and it is assumed to be negative definite. This means that the convergence rate of the input to the optimum depends on  $H$ , which is unknown.

There are five tuning parameters in this method, and they affect the system in different ways:

- $K$ : The integrator gain influences the convergence rate. Larger  $K$  yields faster convergence, but is less robust to disturbances. Also, too large  $K$  can lead to convergence instability.
- $\{a_i, w_i\}$ : Dither amplitude and frequency also influence convergence rate, where higher values allow for faster convergence. However, too large  $a_i$  can result in average loss in the criterion to maximize. Regarding  $w_i$ , it must be selected small enough such that there is a large enough time-scale separation between perturbation in the input and system's dynamics. On the other hand, higher

frequencies allow for faster estimation of gradient and thus higher gains  $K$  with higher level of robustness.

- $\{w_l, w_h\}$ : The cut-off frequencies of the filters must be selected lower than  $w$ , where specially  $w_l$  has to fulfill the constraint  $w_l << \min\{|w_i - w_j|\}, \forall i \neq j$  to ensure the smallest of the high-frequency components of the demodulation operation is filtered out.

#### D. Newton-based ESC

Newton-based ESC (NBESC) is presented in [13] and, in essence, follows the same logic of GBESC. A scheme of this algorithm is presented in Fig. 4. It innovates from the

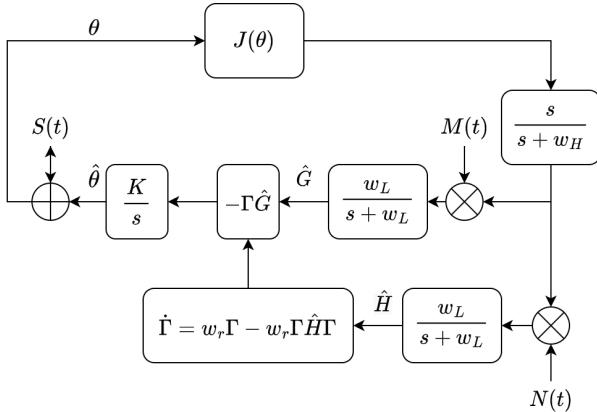


Fig. 4. NBESC scheme.

GBESC by tackling the problem of different convergence rate between inputs. As aforementioned, in GBESC, the closed loop dynamics of the input error depend on the unknown criterion's Hessian. In the multivariate case, this matrix is unknown and it is not guaranteed to be diagonal, neither to have similar magnitude between its diagonal entries. In practice, this implies that different inputs converge at different rates, and that interactions between inputs affect their convergence, which can affect robustness to disturbances. To counteract this problem, NBESC estimates the Hessian of the criterion, so that it can be canceled in the closed loop dynamics. To estimate the Hessian a new demodulation signal  $N(t)$  is introduced. The authors in [14] suggest the selection of  $N(t)$  as

$$N_{i,i}(t) = \frac{16}{a_i} \left( \sin(w_i t)^2 - \frac{1}{2} \right)^2$$

$$N_{i,j}(t) = \frac{4}{a_i a_j} \sin(w_i t) \sin(w_j t)$$

and prove that  $\hat{H} = N(t)y$  is an estimate of the average value of the Hessian over  $\Pi$ , which is the minimum common period to all dithers  $S_i$ . The addition of this demodulation operation results in additional constraints to the dithers' frequencies selection, given by:

$$w_i \neq \left\{ w_j, \frac{1}{2}(w_j + w_k), 2w_j + w_k, w_j + w_k \pm w_l \right\},$$

for distinct  $i, j$ , and  $k$ . Following the Hessian estimation, its inverse is estimated using a differential Riccati equation,  $\dot{\Gamma} =$

$w_r\Gamma - w_r\Gamma\hat{H}\Gamma$ , which has an locally stable equilibrium point at  $\hat{H}^{-1}$ . Estimating this matrix asymptotically is useful to avoid numerical singularities when inverting  $\hat{H}$ . The estimate  $\Gamma$  is then multiplied in the feedback loop, as in Fig. 4. Assuming  $\hat{H} \approx H$  and linearizing the closed loop dynamics around the equilibrium point, it results that

$$\dot{\theta} = -K\tilde{\theta},$$

which allows for user-assignable convergence of  $\hat{\theta}$  to 0, in a fixed-step fashion.

The tuning process is analogous to GBESC, with one additional parameter and different constraints on  $w_l$ :

- $w_r$ : The parameter of the differential Riccati equation dictates the convergence of  $\Gamma$  to the equilibrium point and also, implicitly influences the convergence of  $\hat{\theta}$  to 0.
- $w_l$ : With additional demodulation high-frequency components from  $\hat{H}$ , this cut-off frequency has now to be selected as  $w_l << \min\{w_i, |w_i - w_j|, |w_i - w_j - w_k|, |2w_i - w_j|\}, \forall i \neq j \neq k$ .

## IV. RESULTS AND DISCUSSION

This section starts with an analysis on wingsail aerodynamics. Then, extremum seeking is applied for measured AWA, which is collected during ship experiments in the Baltic Sea, from a ship prototype with the same sail scheme as in Fig. 1. The results of extremum seeking for such scenarios are presented first for a two-wingsail (2D) configuration scheme, followed by the four-wingsail (4D) counterpart. The reason why the results with less wingsails is presented is due to the lack of time and computational capacity to have finished the 4D tests and simulations, and thus the 2D scheme is used for provisional results.

### A. Wingsail Aerodynamics

Wingsail aerodynamics analysis, namely the computation of the aerodynamic force acting on the wingsail, is mathematically intricate and requires extensive numerical computations. This work relies on the JavaFoil software for that matter, an airfoil analysis software in the two-dimensional space.

The wingsail lift and drag coefficients,  $c_L$  and  $c_D$ , are presented in Fig. 5, as function of  $\alpha$ . Note that given the stationarity assumption,  $c_D$  is significantly smaller in comparison to  $c_L$ , and thus it is expected that thrust depends in greater extent on lift than on drag.

Then, the optimal  $\delta^* = [\delta_1^* \dots \delta_n^*]$  was obtained for  $AWA \in [0, 180]$ , for two- and four-dimensional wingsail configurations. The results are equal for the symmetric range of AWA, due to the symmetric geometry of the wingsail. Recall the function is not convex in all of its domain. Thus, for the solver to reach the global optimal  $\delta^*$ ,  $\delta(0)$  is selected close enough to  $\delta^*$ , as  $\delta(0) = -AWA + \alpha$ . This initialization correspond to  $\delta^*$  that maximizes lift, but given  $c_D$  is considerably low,  $\delta^*$  is similar for lift and thrust optimization. The optimization solver is a gradient-based nonlinear program solver in MATLAB, which applies an interior-point-method

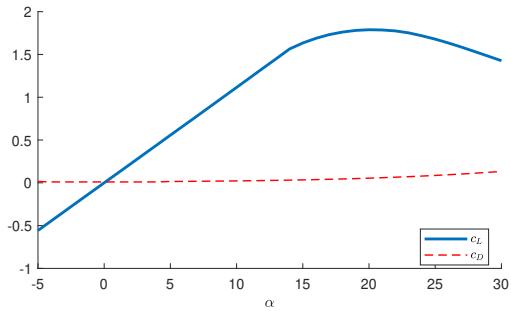


Fig. 5.  $c_L$  vs  $c_D$

type of algorithm. Fig. 6 and 7 present the computed  $\delta^*$  for both configurations.

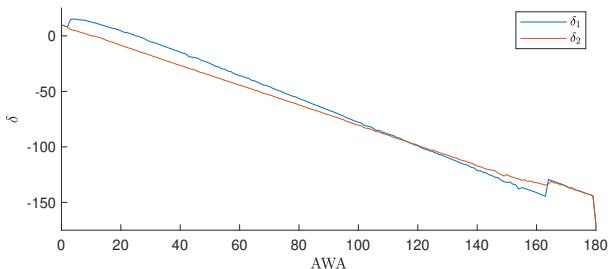


Fig. 6. Two wingsails:  $\delta^*$

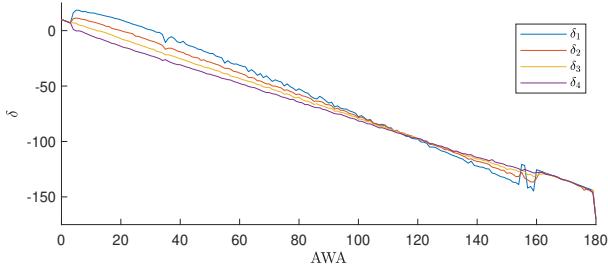


Fig. 7. Four wingsails:  $\delta^*$

In both,  $\delta^*$  evolve nearly linear with time, mainly given the low  $c_D$  of this airfoil. I.e., the thrust optimization is the same as lift optimization for values up to  $AWA \approx 160$ . The results after that value are not accurate, since there are nontrivial configurations that can also yield optimal values of  $c_T$  that were not considered, e.g.  $\delta = [\pm 90, \pm 90]$ . The results show also that around  $AWA \approx 130$  the difference between  $\delta_i$  is rather small, and after that the order of  $\delta$  magnitude changes.

### B. AWA datasets

Real AWA measurements from ship experiments are used to validate the extremum seeking implementation for this specific application. The data is collected from a sensor on the ship. Note that these are point-wise measurements of the wind and are rather noisy, and thus they are not totally informative of the resultant wind effect on a large three-dimensional bodies. For that reason, simulations were also conducted where the

datasets were low-pass filtered to simulate the averaging of AWA variations on the entire surface of the wingsails.

Two datasets are considered. In one, the ship is considered to be sailing with constant heading with the wind from port, around  $AWA = 105^\circ$ . The sampling frequency is 5 Hz. The dataset is of short duration, and is mainly a test-bed to evaluate the performance of extremum seeking in the presence of real AWA variations, which recall is the uncontrollable input of  $T$ . Then, as aforementioned, the dataset is low-pass filtered with a cut-off frequency  $f_c = \frac{1}{11}$  so that the frequency of the AWA variations is slower than the common frequency to all perturbations. The AWA measurements and their frequency spectrum are presented in Fig. 8 and 9, respectively. The latter is zoomed in the frequency range of interest for later comparison.

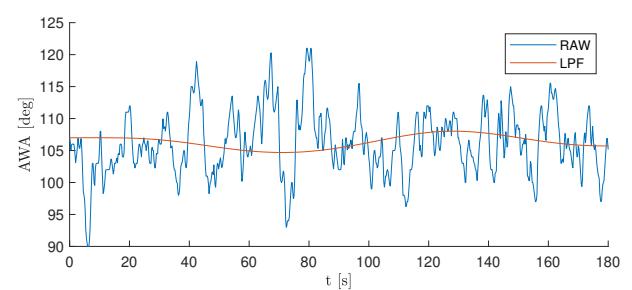


Fig. 8. Constant-heading: AWA measurements

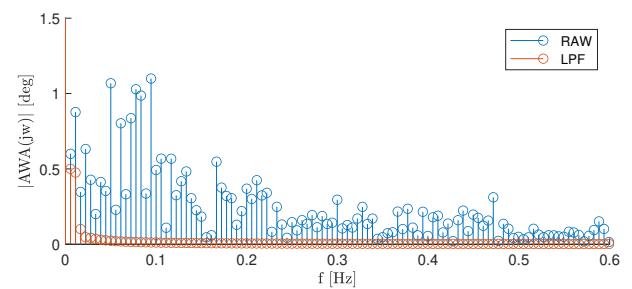


Fig. 9. Constant-heading: AWA measurements frequency spectrum

The second dataset is of longer duration and represents a tacking maneuver, where the ship is sailing upwind. During the constant-heading stages, wind angle is around  $AWA = \pm 45^\circ$ . This dataset is useful to evaluate longer-duration tracking of optimal sheeting angles, as well as to evaluate the algorithm behavior given large-magnitude AWA variations, which is the case while ship is maneuvering between different heading levels. The same filtering operation as described for the previous case applies. Fig. 10 and 11 are the AWA measurements and their frequency spectrum, respectively, where the latter is zoomed as before.

In what follows, the first dataset is referred to as constant-heading AWA dataset and the second as tacking AWA datasets, for simplification reasons.

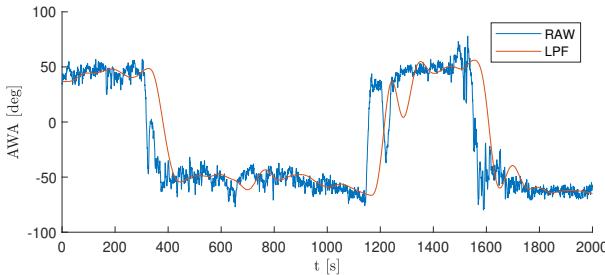


Fig. 10. Tacking: AWA measurements

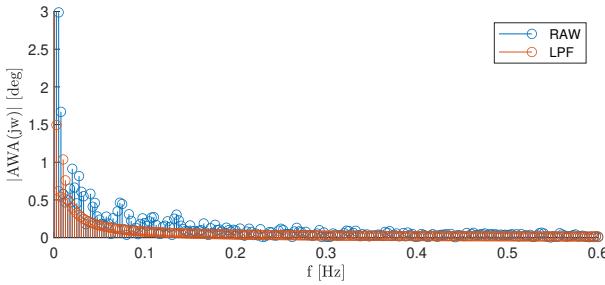


Fig. 11. Tacking: AWA measurements frequency spectrum

### C. Extremum seeking tuning

The tuning of the extremum seeking was conducted in a similar way for all datasets, and some parameters are globally defined for all simulations. Some considerations on this process are important to highlight before presenting the results.

First, the upper bound of the dithers' frequency was selected as  $\bar{f} = 0.2$  Hz, to ensure  $\frac{1}{\bar{f}} \gg \tau$ . The frequencies of the wingsail the furthest from the AW origin is selected as  $f_i = \bar{f}$ , while the frequency of the remaining dithers is maximized such that it fulfills both GBESC and NBESC constraints. A minimum difference between frequencies is imposed to avoid implementation-induced couplings.

The amplitude of the perturbation in  $\delta$  is also of important selection. If selecting frequencies in the same range of the rate of AWA variations, there will be couplings between both inputs in the output. This, in turn, affects the accuracy of demodulation later in the loop, and leads to poor performance. For that reason,  $a_i$  should be selected larger in case there are AWA harmonics in  $f_i$ .

Finally,  $K$  and  $w_r$  are the main tunable parameters between simulations, and their value is selected as large as possible to guarantee faster convergence without leading the system to diverge from the optima. For fair comparison,  $K_g = -K_n \Gamma(0)$ , where  $K_g$  and  $K_n$  are the integrator gains for GBESC and NBESC, respectively, and  $\Gamma(0)$  is the initial condition of the differential Riccati equation, which is derived numerically from the look-up table.

### D. 2D Extremum Seeking

The first step of the extremum seeking implementation was the computation of a look-up table of the criterion around the

optimal  $\delta$ , in order to speed up the simulations and to obtain references plots. For this configuration, the look-up table has a resolution of  $4^\circ$  and  $3^\circ$  for AWA and  $\delta$ , respectively. Then, for each sampling time instant, the criterion is linearly interpolated at the current value of AWA and  $\delta$ .

Secondly, GBESC and NBESC were implemented for the constant-heading AWA dataset. The tuning parameters are  $\epsilon = 0.001$ ,  $a_i = 2^\circ$ ,  $\forall i$ ,  $f_1 = 0.17$ ,  $f_2 = 0.2$ ,  $K_n = 5\epsilon\mathbf{I}$ ,  $\Gamma(0) = -0.06\mathbf{I}$ ,  $w_r = 2\pi 0.15\epsilon\bar{f}$ , and  $w_h = 30\epsilon$ . The low-pass filter after demodulation is not implemented since its too strict cut-off frequency results in too slow performance.  $T$  is low-pass filtered with cut-off frequency of 1 Hz, to remove high-frequency output variations. The sheeting angles of wingsail 1 and 2 are presented in Fig. 12 and 13, respectively; the output  $T$  is presented in Fig. 14. The reference curves,  $\delta^*$  and  $J^* = T^*$  are obtained from the look-up table by interpolating the optima for the given AWA. Note that the heading of the ship is considered constant, thus a constant feedforward is inputted.

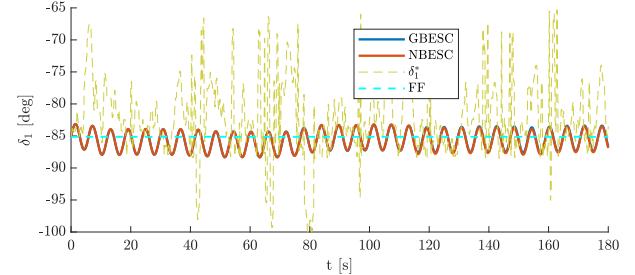


Fig. 12. Constant-heading AWA:  $\delta_1$

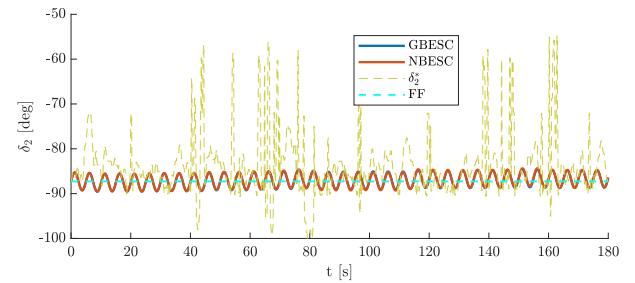


Fig. 13. Constant-heading AWA:  $\delta_2$

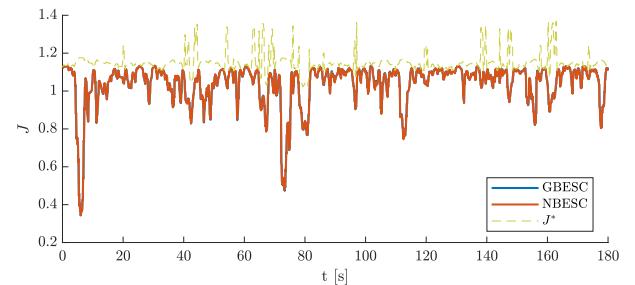


Fig. 14. Constant-heading AWA:  $c_T$

With the presented choice of  $f_i$ , the controller is too slow in tracking the AWA high-frequency variations. For that reason,  $K_n$  is purposely small to avoid divergence, and thus  $\delta$  is only able to track the low-frequency evolution of AWA. Furthermore, it is immediate to conclude that both GBESC and NBESC results are similar. This is due to the small value of  $w_r$ , which diminishes the adaptive property of the feedback gain  $K_n\Gamma$ .

Looking closer at the subsystem of NBESC responsible for Hessian estimation and asymptotically inversion, the average value of  $\hat{H}$  and  $\Gamma$  over  $\Pi$  is presented in Fig. 15 and Fig. 16, respectively.

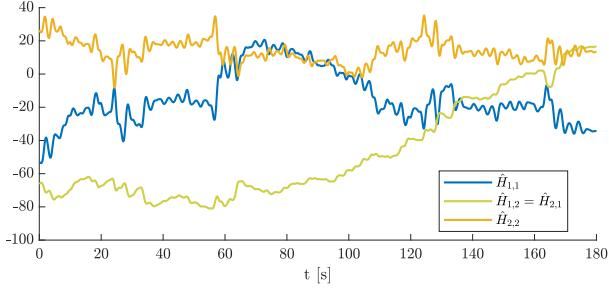


Fig. 15. Constant-heading AWA:  $\hat{H}$

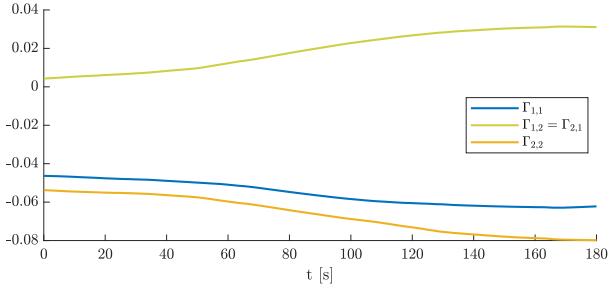


Fig. 16. Constant-heading AWA:  $\hat{H}$

Note that  $\hat{H}$  does not converge to any clear value. In fact,  $H$  is not constant over  $\Pi$ , thus it is not clear to which value  $\hat{H}$  is going to converge to. To avoid the instability of  $\hat{H}$  to be translated into instability in the feedback gain,  $w_r$  is selected small resulting in a slow evolution of  $\Gamma$ , as in Fig. 16.

From these results, one concludes that none of the algorithms can track accurately the optima, tracking rather a low-frequency evolution of it. Moreover, NBESC is not robust resulting in poor Hessian estimation, which forces a too conservative tuning of the Riccati equation to prevent divergence.

On the other hand, when filtering the AWA measurements as in Fig. 8, the conclusions differ. With such filtering, not only the perturbations are faster than the AWA variations, but also the couplings in frequency between AWA and dithers are removed, enhancing demodulation accuracy. For this case, the tuning is similar except from  $a_i = 1^\circ$ ,  $K_n = 100\epsilon I$  and  $w_r = 2\pi 5\epsilon$ . Figures 17, 18 and 19 present  $\delta_1$ ,  $\delta_2$  and  $T$  respectively.

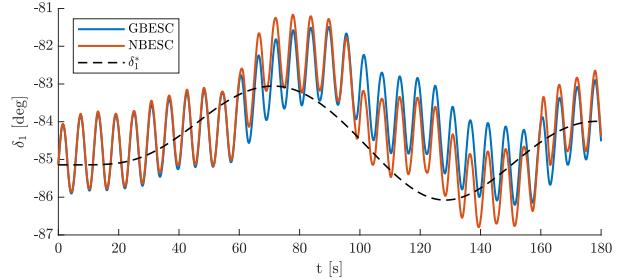


Fig. 17. Constant-heading filtered AWA:  $\delta_1$

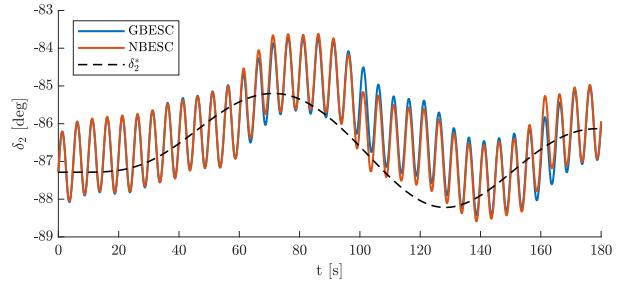


Fig. 18. Constant-heading filtered AWA:  $\delta_2$

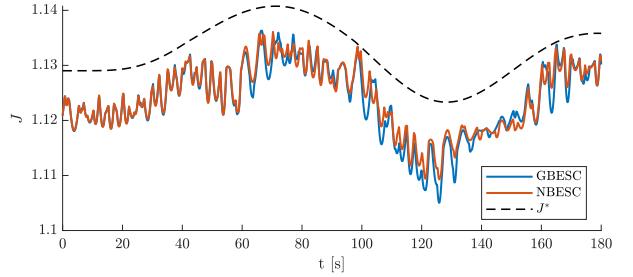


Fig. 19. Constant-heading filtered AWA:  $cT$

It is clear that the algorithm is significantly more accurate for this scenario. Moreover, NBESC is able to attenuate the difference in convergence rate between wingsails, where  $\Gamma$  evolution is a now faster and stable estimate of  $\hat{H}^{-1}$ , as it is shown in Fig. 20. This is as expected, since  $H$  variations are of lower frequency and magnitude.

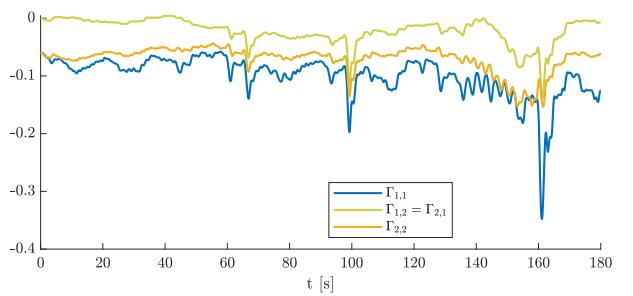


Fig. 20. Constant-heading filtered AWA:  $\Gamma$

Performance indicators (PI) are presented in Table I, where  $c_T^a$  is the accumulated  $cT$  along the simulation and MAE is the mean absolute error to  $\delta_i^*$ . From these, it is possible to note an

TABLE I  
2D CONSTANT-HEADING FILTERED AWA: PIs

	$c_T^a$	MAE $\delta_1$ [deg]	MAE $\delta_2$ [deg]
GBESC	2021	0.9214	0.8379
NBESC	2025	0.8265	0.7958
Optimal	2039	-	-

improvement on all indicators, but most importantly a decrease in the difference between MAE( $\delta_1$ ) and MAE( $\delta_2$ ) in NBESC. In overall, the results are promising for this case, suggesting good performance in the presence of less disturbed wind effect on  $T$ , where NBESC is slightly superior to GBESC.

The simulations for the tacking dataset follow. In this case, it is both interesting to analyze the constant-heading and maneuver stages. The tuning parameters are now  $\epsilon = 0.001$ ,  $a_i = 2^\circ$ ,  $\forall i$ ,  $f_1 = 0.2$ ,  $f_2 = 0.17$ ,  $K_n = 5\epsilon\mathbf{I}$ ,  $\Gamma(0) = -0.5\mathbf{I}$ ,  $w_r = 2\pi 0.005\epsilon\bar{f}$ , and  $w_h = 30\epsilon$ . In this case, a piecewise linear feedforward is inputted. Analogously to the previous simulations, the sheeting angles of wingsail 1 and 2 are presented in Fig. 21 and 22, respectively; the output  $T$  is presented in Fig. 23.

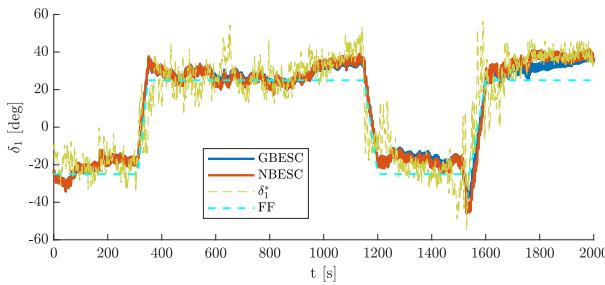


Fig. 21. Tacking AWA:  $\delta_1$

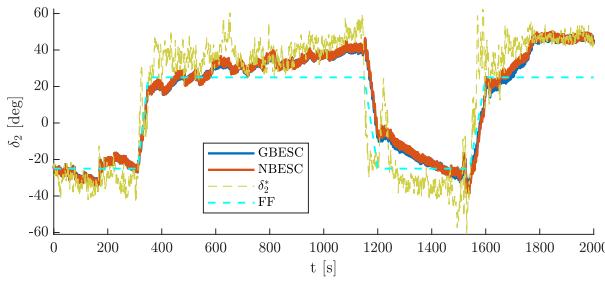


Fig. 22. Tacking AWA:  $\delta_2$

The tracking of the high-frequency optima is not accurate, where instead the low-frequency evolution is tracked. The feedforward is highly noninformative, suggesting that low-accuracy feedforward is sufficient in practical applications. In this case, as plotted in Fig. 24,  $\hat{H}$  is quite unstable, with large magnitude variations that have to be counteracted choosing a very small  $w_r$ . For that reason, and similarly to the previous dataset simulations, NBESC was not robust to output high-frequency variations, and thus does not show any potential of improvement from GBESC.

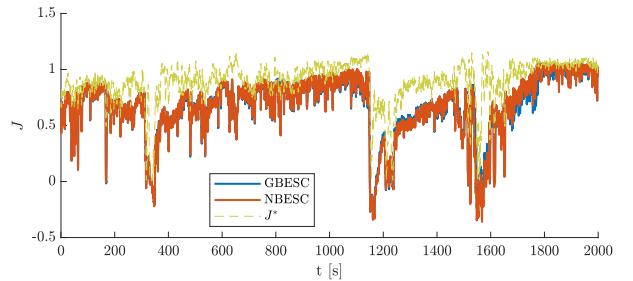


Fig. 23. Tacking AWA:  $c_T$

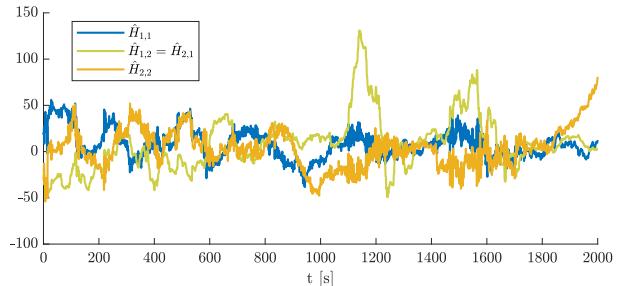


Fig. 24. Tacking AWA:  $\hat{H}$

When simulating both ESC algorithms with filtered AWA from this dataset, as in Fig. 10, the tuning is in every sense similar, except from:  $a_i = 1^\circ$ ,  $K_n = 40\epsilon\mathbf{I}$  and  $w_r = 2\pi 0.05\epsilon$ . The results are plotted in Fig. 25, 26 and 27, which present  $\delta_1$ ,  $\delta_2$  and  $T$ , respectively. In this case the tracking is indeed accurate. Nevertheless, the NBESC kept not being able to improve GBESC results, due to the small value of  $w_r$  to prevent instability. Mainly, the maneuver parts are particularly difficult for the hessian estimation, where  $\hat{H}$  is quite unstable, as in Fig. 28. Finally, the aforementioned performance indicators are presented in Table II.

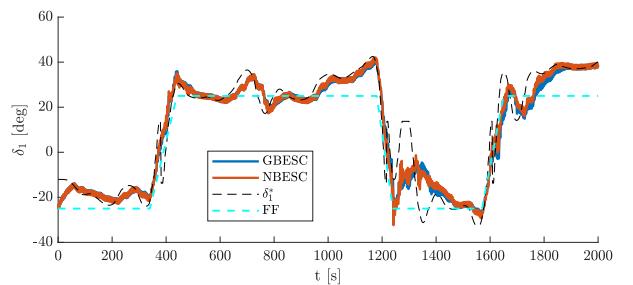


Fig. 25. Tacking filtered AWA:  $\delta_1$

TABLE II  
2D TACKING FILTERED AWA: PIs

	$c_T^a$	MAE $\delta_1$ [deg]	MAE $\delta_2$ [deg]
GBESC	15891	4.9480	3.4136
NBESC	16000	4.6152	3.2871
Optimal	17323	-	-

From these results, one concludes that GBESC is rather accurate in tracking the optima when in presence of lower AWA

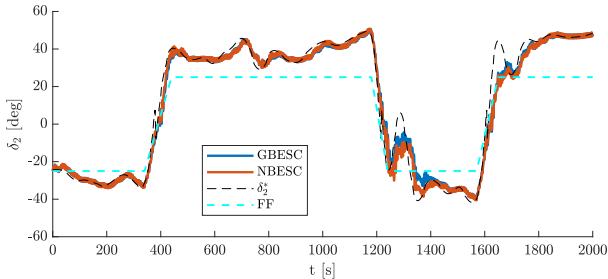


Fig. 26. Tacking filtered AWA:  $\delta_2$

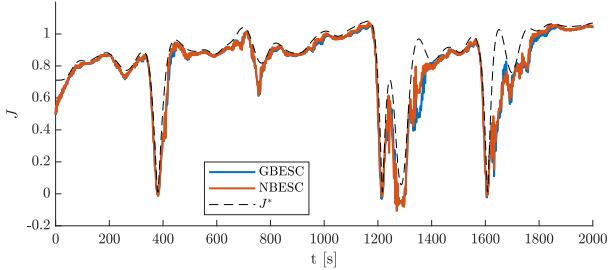


Fig. 27. Tacking filtered AWA:  $c_T$

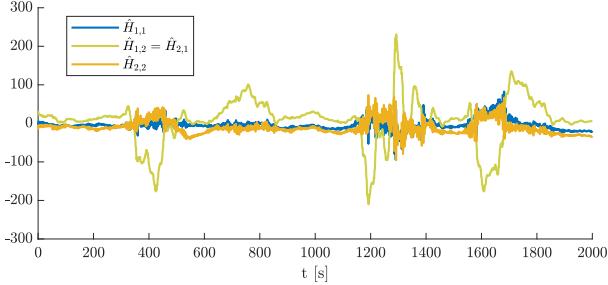


Fig. 28. Tacking filtered AWA:  $\hat{H}$

variations, which is promising for future practical applications, assuming the wind averaging effect. However, even with less noisy data, NBESC performed poorly in  $\hat{H}$  namely when in presence of heading variations.

Finally, the scenario where both wingsails are controlled with the same frequency is compared with the multivariate control counterpart. The goal is to validate that multivariate control can in fact improve tracking accuracy. The tuning was the same as for the tacking raw AWA dataset, apart from the fact that for the univariate GBESC (GBESC-UNI) both dithers' frequencies are selected as  $f_i = \bar{f}$ . Analogously to the previous plots, the sheeting angles of wingsail 1 and 2 are presented in Fig. 29 and 30, respectively, and the output  $T$  is presented in Fig. 31; the PIs are presented in Table III.

By selecting equal dither frequencies, the demodulation operations are not able to separate information from each wingsail. For that reason, both wingsails are equally actuated, thus never reaching the multivariate optima. This decreases performance, namely in loss of output value - in Table III it is clear the univariate case yields a significantly lower output than the multivariate one.

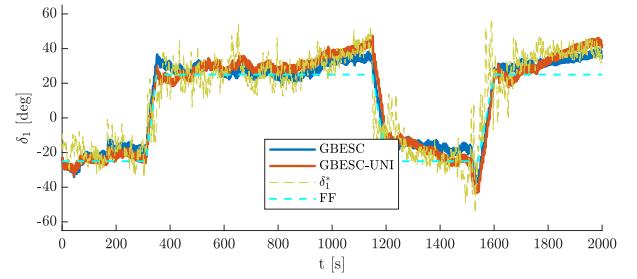


Fig. 29. Tacking AWA (GBESC vs GBESC-UNI):  $\delta_1$

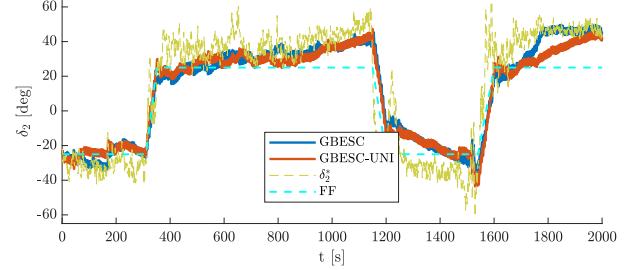


Fig. 30. Tacking AWA (GBESC vs GBESC-UNI):  $\delta_2$

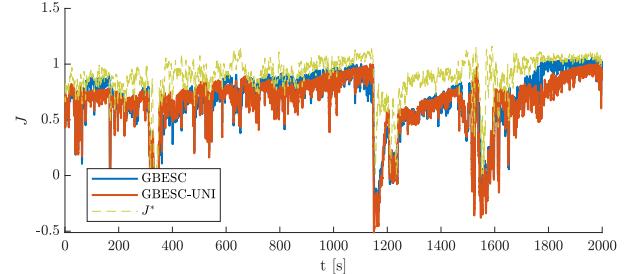


Fig. 31. Tacking AWA (GBESC vs GBESC-UNI):  $c_T$

TABLE III  
4D CONSTANT-HEADING FILTERED AWA: PIS

	$c_T^a$	MAE $\delta_1$ [deg]	MAE $\delta_2$ [deg]
GBESC	13542	7.5504	9.9924
GBESC-UNI	13086	7.2880	11.1778
Optimal	17759	-	-

#### E. 4D Extremum Seeking

Analogously to the 2D counterpart, look-up tables of the criterion around the optimal  $\delta$  were computed. For this configuration, the look-up table has a resolution of  $3^\circ$  and  $2^\circ$  for AWA and  $\delta$ , respectively, and the interpolation is also linear as before. Nevertheless, the JavaFoil software is twice as slow for this configuration, and also the dimension of the look-up table is exponentially larger than the 2D counterpart. For these reasons, the look-up table was computed only around a narrow neighborhood of the optima, and only for the constant heading dataset. Even with such search-space reductions, the time-scale of simulations was of days, which was definitely a constraint on development time.

For those reasons, only constant-heading filtered AWA sim-

ulations were conducted. The tuning parameters for this test are  $\epsilon = 0.001$ ,  $a_i = 1^\circ$ ,  $\forall i$ ,  $f_1 = 0.08$ ,  $f_2 = 0.14$ ,  $f_3 = 0.18$ ,  $f_4 = 0.2$ ,  $K_n = 150\epsilon\mathbf{I}$ ,  $\Gamma(0) = -0.2\mathbf{I}$ ,  $w_r = 2\pi 1.5\epsilon\bar{f}$ , and  $w_h = 30\epsilon$ . The sheeting angles of wingsail 1 to 4 are presented in Fig. 32 to 35, respectively; the output  $T$  is presented in Fig. 36.

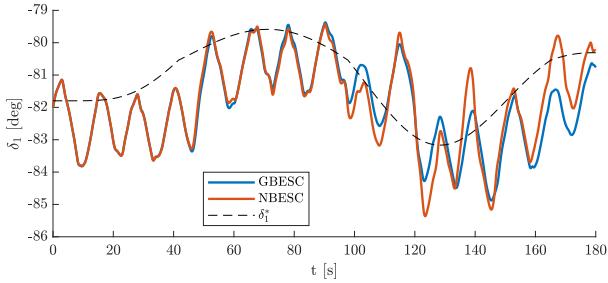


Fig. 32. 4D Constant-heading AWA :  $\delta_1$

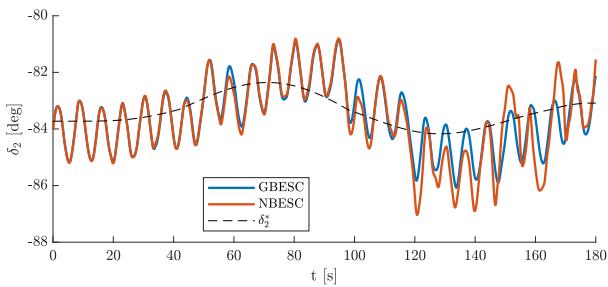


Fig. 33. 4D Constant-heading AWA :  $\delta_2$

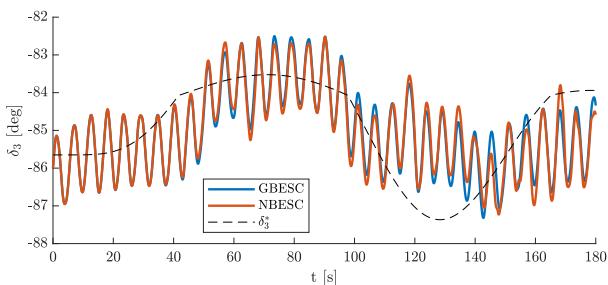


Fig. 34. 4D Constant-heading AWA :  $\delta_3$

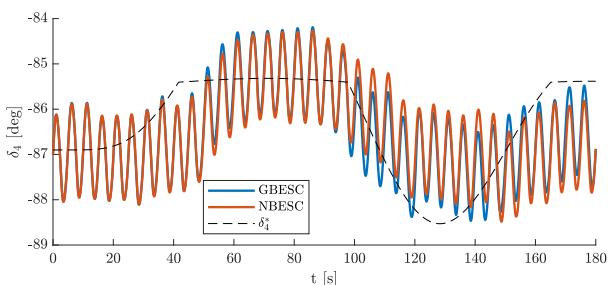


Fig. 35. 4D Constant-heading AWA :  $\delta_4$

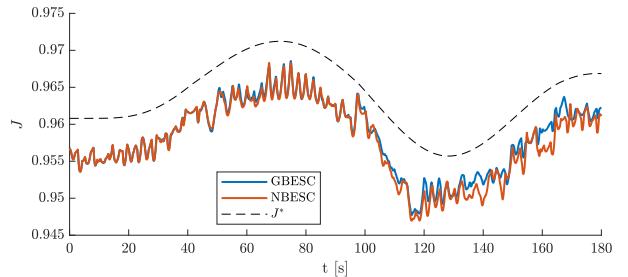


Fig. 36. 4D Constant-heading AWA :  $CT$

Recall that  $K$  and  $w_r$  are selected as high as possible before leading to instability. This resulted in a nearly accurate tracking of the optima, but with still a slower behavior than what it would be desired. This is as expected, since convergence rate in 4D is expected to be slower, due to greater interactions between sails and lower frequency inputs. There were no major differences between algorithms, mainly because high selection of gains attenuate the potential for improvement of Hessian cancellation. Experimentally, the major room for improvement was found to be in the increase in frequency selection. Increasing  $f_i$ , larger  $K$  can be implemented without leading to instability. Therefore, experiments show that GBESC outperforms NBESC for this case since it allows for less strict choice of  $f_i$ , and thus it is easy to select larger values for these parameters, and consequently larger  $K$  increasing convergence rate.

## V. CONCLUSION

GBESC and NBESC were implemented for airfoil thrust maximization in a static scenario. Real measurements of AWA were used to simulate optima high-frequency variation. Simulations with filtered versions of these datasets were also included, to simulate less disturbed wind scenarios and possible noise cancellation for large three-dimensional bodies. Low-informative feedforward was used to guarantee convergence to global optima. The results demonstrate that, with a conservative selection of  $\bar{f}$  necessary to ensure time-scale separation between dithers' perturbation and wingsail dynamics, the algorithms are only able to track the low-frequency evolution of AWA, where high-frequency variations of this angle are not tracked. The performance improves clearly for the filtered AWA case, where the algorithms performed accurately. NBESC, however, is not adequate for this application, since the intrinsic Hessian estimation leads the system to instability. The lack of robustness of this method is definitely a disadvantage for unpredictable and nondeterministic AWA operating points as in sailing. Finally, the 4D configuration proved to be slower to converge, and the frequency selection was too conservative for accurate tracking for this configuration. Therefore, in this configuration, and for this tuning, GBESC is preferred due to the less strict frequency selection constraints. Future work consists of evaluating carefully how much  $\bar{f}$  can be increased and studying the impact of the wind on the thrust for three-dimensional large-surface wingsails. Furthermore, it

is important to implement the algorithm in the presence of wave disturbances, using ship dynamics and kinematics, to better understand the effect of permanent oscillations on the overall system, as well as possible couplings between dither frequencies and e.g. ship rolling period or wave frequency.

#### REFERENCES

- [1] “Annual report 2021,” United Nations Conference on Trade and Development, Tech. Rep., 2021.
- [2] “Review of maritime transport,” United Nations Conference on Trade and Development, Tech. Rep., 2021.
- [3] “Fourth greenhouse gas study 2020,” International Maritime Organization, Tech. Rep., 2020.
- [4] H. Saoud, M.-D. Hua, F. Plumet, and F. B. Amar, “Optimal sail angle computation for an autonomous sailboat robot,” in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 807–813.
- [5] P. Herrero, L. Jaulin, J. Vehí, and M. A. Sainz, “Guaranteed set-point computation with application to the control of a sailboat,” *International journal of control, automation and systems*, vol. 8, no. 1, pp. 1–7, 2010.
- [6] I. Maria Viola and R. G. Flay, “Sail aerodynamics: on-water pressure measurements on a downwind sail,” *Journal of ship research*, vol. 56, no. 04, pp. 197–206, 2012.
- [7] L. Xiao, J. C. Alves, N. A. Cruz, and J. Jouffroy, “Online speed optimization for sailing yachts using extremum seeking,” in *2012 Oceans*, 2012, pp. 1–6.
- [8] M. Corno, S. Formentin, and S. M. Savarese, “Data-driven online speed optimization in autonomous sailboats,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 762–771, 2016.
- [9] Z. Shen, S. Wang, H. Yu, and C. Guo, “Online speed optimization with feedforward of unmanned sailboat via extremum seeking without steady-state oscillation,” *Ocean Engineering*, vol. 189, p. 106393, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0029801819305475>
- [10] Z. Shen, X. Fan, H. Yu, C. Guo, and S. Wang, “A novel speed optimisation scheme for unmanned sailboats by sliding mode extremum seeking control without steady-state oscillation,” *The Journal of Navigation*, pp. 1–18, 2021.
- [11] D. Yingjie, Z. Xianku, and Z. Guoqing, “Fuzzy logic based speed optimization and path following control for sail-assisted ships,” *Ocean Engineering*, vol. 171, pp. 300–310, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0029801818312460>
- [12] M. F. Silva, A. Fribe, B. Malheiro, P. Guedes, P. Ferreira, and M. Waller, “Rigid wing sailboats: A state of the art survey,” *Ocean Engineering*, vol. 187, pp. 106–150, 2019.
- [13] A. Ghaffari, M. Krstić, and D. Nešić, “Multivariable newton-based extremum seeking,” *Automatica*, vol. 48, no. 8, pp. 1759–1767, 2012.
- [14] K. B. Ariyur and M. Krstic, *Real-time optimization by extremum-seeking control*. John Wiley & Sons, 2003.