Online optimization of different objectives in robotic sailing: Simulations and experiments

Wrede, Daniel; Jost, Adam; Jouffroy, J.

Published in:
2015 IEEE Conference on Control and Applications, CCA 2015 - Proceedings

DOI:
https://doi.org/10.1109/CCA.2015.7320728

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

Link to publication

Citation for pulished version (APA):
Wrede, D., Jost, A., & Jouffroy, J. (2015). Online optimization of different objectives in robotic sailing: Simulations and experiments. In *2015 IEEE Conference on Control and Applications, CCA 2015 - Proceedings* IEEE. https://doi.org/10.1109/CCA.2015.7320728

2015 IEEE Conference on Control Applications (CCA)
Part of 2015 IEEE Multi-Conference on Systems and Control
September 21-23, 2015. Sydney, Australia

Online optimization of different objectives in robotic sailing: simulations and experiments

Daniel Wrede, Jost Adam and Jerome Jouffroy

***Abstract*— This paper investigates online optimization for the different objectives found in robotics sailing such as sailing angle optimization, but also the so-called Velocity-Made-Good, as well as heel angle optimization used as a means to detect an edge of the deadzone. Simulation as well as experimental results on an autonomous sailing platform are presented.**

***Index Terms*— Sailing robotics, hill-climbing, online optimization, sail control.**

I. Introduction

In the last decade, the community working on sailing autonomous vehicles or robotics proposed a number of interesting results related to the control and guidance of these energy-efficient vehicles (see for example [1], [3], and [9] for a few notable references). Many of these works are, roughly speaking, dedicated to strategies for guiding the vehicles towards a desired waypoint or path. Such investigations are very relevant, mostly because of the specificity of sailing vehicles, which, in order to go in a direction opposite to the wind, have to perform zigzag manoeuvres known as tacking. Surprisingly, there is comparatively little work on adjusting parameters linked to improvement of propulsion. In this context, tuning the angle of a sail or boom is but one area which was studied (see [13] and [15]).

However, in sailing, this kind of regular adjustment in order to improve performance is not limited to tuning a sail. For example, in order to win a race or save time and energy, one would like to improve or maximize the velocity in the general direction of an objective, which is known to sailors as Velocity-Made-Good.

In this paper, we present results on the application of online optimization on different possible scenarios in robotic sailing beyond mere sail optimization by also including, besides the former, Velocity-Made-Good optimization and heel angle optimization for deadzone edge detection. Because of its applicability, robustness and simplicity, as well as basic implementation issues, the well-known hill-climbing algorithm is used for the optimization method.

After this introduction, we start in section II by briefly recalling the equations of the dynamic model of a 4 DOF sailing yacht which will be used for the simulations. Then, after presenting the hill-climbing algorithm and its proof of convergence, we describe the above-mentioned optimization scenarios of this study. Simulation results and well as full-scale experiments on an autonomous sailing vehicle are presented and discussed in section IV. Finally, brief concluding remarks are given in section V.

II. Mathematical modelling and sail dynamics

A. A 4DOF sailing yacht model

In this section, we briefly recall the mathematical model of a sailing yacht simulator, which was used for development and validation purposes. The model, introduced in [16] has four degrees of freedom, and includes surge, sway, yaw and also roll motion. Note that it is one of the specificities of the model introduced by Xiao and Jouffroy to include the roll motion. The interest lies in the possibility to study different aspects such as control and stabilisation using a laterally moving weight (see in particular [7]), but aIso excess roll angle induced by excessive tightening of the main sail. Another feature of this model is the extensive use of Fossen's notation [5], which allows for a simple and compact notation useful in modelling and control purposes. Hence, a sailing yacht is represented by the following expression

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}+\boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu}+\boldsymbol{D}(\boldsymbol{\nu},\boldsymbol{\eta})+\boldsymbol{g}(\boldsymbol{\eta})=\boldsymbol{\tau}(\boldsymbol{\eta},\boldsymbol{\nu},\delta_r,\delta_s,\boldsymbol{v}_{tw}) \quad (1)$$

where $\boldsymbol{\nu}=[u,v,p,r]^T$ is the generalized velocity vector in the body-fixed frame and $\boldsymbol{\eta}=[x,y,\phi,\psi]^T$ is the vector of generalised positions and angles in the earth-fixed frame comprising the (x,y) position, heeling and heading angles of the yacht. $\boldsymbol{M}$ is the system inertia matrix, $\boldsymbol{C}$ the system Coriolis-centripetal matrix, $\boldsymbol{D}$ the damping matrix, $\boldsymbol{g}(\boldsymbol{\eta})$ contains the restoring forces and $\boldsymbol{\tau}$ comprising the forces used to control the vehicle, i.e. the forces generated by the sail and rudder (for more details on the exact expressions of these quantities, we refer the reader to [16]).

In expression (1), variables δ_r and δ_s are the rudder angle and the sail angle, respectively, and represent the control inputs of the system. The two-dimensional vector $\boldsymbol{v}_{tw}$ is the known disturbance created by the wind. The vector $\boldsymbol{\eta}$ is computed from the following coordinate transform:

$$\dot{\boldsymbol{\eta}}=\boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu}, \quad (2)$$

where $\boldsymbol{J}(\boldsymbol{\eta})$ is a transformation matrix between the two above-mentioned frames.

In the present study, we are chiefly concerned with focusing on issues related to the sail and the impact of sail propulsion, so that, for simplicity, we assume that the rudder is controlled by a heading controller of the form:

$$\delta_r=k(\psi-\psi_d) \quad (3)$$

where $k(\cdot)$ is a function of the error between the heading angle and ψ_d representing the desired heading of the vehicle.

Daniel Wrede, Jost Adam and Jerome Jouffroy are with the Mads Clausen Institute, University of Southern Denmark, DK-6400 Alsion 2, Sønderborg, Denmark (e-mail: jerome@mci.sdu.dk).

978-1-4799-7787-1/15/$31.00 ©2015 IEEE

Fig. 1. Simulator variables are visualized in a 3D simulation environment.

Note that there are many possibilities for the above heading controller (see for example [3] or [17]). For the present study, it suffices to assume that the system is stabilised around its desired heading.

Equations (1)-(3) are implemented in Matlab/Simulink and visualised through a virtual environment (see figure 1) using the Simulink 3D Animation toolbox. Note that the complete simulator is available for the community and can be downloaded on Matlab Central.

B. Sail dynamics

Some modifications of the original simulator were performed in order to have a more realistic behavior of the sail, especially around the singular case that is the wind astern configuration, where the wind angle in the body-fixed frame can jump directly from $-\pi$ to π. In order to avoid this resulting in unnatural fast switching of the sail, we introduced a relay-like nonlinearity producing a more natural behavior.

III. Hill-climbing for different sailing objectives

A. Motivations and choice of the hill-climbing strategy

Online optimization through feedback control has obviously many applications in different areas. In an autonomous sailing context, whether we talk about sailing drones or automated yachts, an application that directly comes to mind is the optimization of the trimming of the sail(s), which will allow the vehicle to reach the maximum reachable velocity, assuming the environmental conditions and the followed course are kept constant during the optimisation process.

Obviously, performing this kind of optimization can be done using quite a few different techniques. A recently widely-used one is referred to as extremum seeking (see for example [12]), which, roughly speaking, modulates the output or the objective to be optimized in order to perform a basic gradient estimate of the objective function, estimate which is then used to control the system towards its extremum. Our initial contribution [13] investigated the use of extremum seeking in the context of autonomous sailing.

Alternatives to extremum seeking also exists, and the so-called hill-climbing algorithm is one of them. Because of its simplicity, hill-climbing is used in many applications such as photovoltaics, wind and wave energy conversion in what is referred to as Maximum Power Point Tracking (see for example ref. [11] [18] in the context of photovoltaics). Compared to modulation-based extremum seeking techniques, one can see two interesting advantages: as we will see below, it is usually described in a very simple and straightforward discrete-time framework, and in its simple version has increments in the control input which are constant, thus allowing to quite straightforward tuning. In the context of autonomous sailing, this means that a new control input will be applied not all the time but at regular intervals, thereby being interesting on the energy perspective, while the fixed control increments allow to do with actuators which are not necessarily of high resolution.

B. The hill-climbing algorithm

For the sake of clarity, let us briefly describe the hill-climbing algorithm. Assume we would like to optimize a static scalar mapping $\gamma = J(\theta)$ which we assume to be concave, i.e. upper convex. Here the exact form of function $J(\cdot)$ is unknown, however, we measure both input θ and output γ. The hill-climbing algorithm can then be written as follows

$$\begin{aligned}\theta(k{+}1) &= \theta(k)\\ &\quad + h \cdot \operatorname{sign}\big(\gamma(k) - \gamma(k-1)\big) \cdot \operatorname{sign}\big(\theta(k) - \theta(k-1)\big)\end{aligned} \tag{4}$$

where h is a strictly positive constant representing the size of the fixed increments on the control input θ. The sign function is defined as

$$sign(r) = \begin{cases} 1 & \text{if } r \geq 0 \\ -1 & \text{otherwise} \end{cases} \tag{5}$$

Where $r \in \mathbb{R}$. Note that this sign function is not exactly the usual sign function, as the value 0 is avoided to prevent stagnation in case of identical initial conditions.

The ability for the algorithm to perform in an acceptable way and find a maximum for different initial conditions is important, hence we would like to know about the stability properties of the above algorithm. To do so, let us first transform system (4) into the following state-space form:

$$\begin{cases} \theta_1(k+1) = \theta_1(k) \\ \theta_2(k+1) = \theta_2(k) + h \cdot \operatorname{sign}(\theta_2(k) - \theta_1(k)) \\ \qquad\qquad\qquad \cdot \operatorname{sign}\left(J(\theta_2(k)) - J(\theta_1(k))\right) \end{cases} \tag{6}$$

where the state is defined as $\boldsymbol{\theta(k)} = [\theta(k-1), \theta(k)]^T$. In the following, we will also assume, without loss of generality and for the sake of simplicity, that the maximum of function $f(\theta)$ is situated at the origin, i.e. that $\arg\max f(\theta) = 0$. The next theorem states the stability property of system (HC-SS), and thereby of hill-climbing algorithm (4).

Theorem 1 *Consider system (6), then the set defined by*

$$M = \Big\{\boldsymbol{\theta} \in \mathbb{R}^2 : (\theta_1 - \theta_2)^2 = h^2, \max(||\theta_1||, ||\theta_2||) \leq 2h\Big\} \tag{7}$$

is an invariant set and all solutions of system (6) converge asymptotically to M as $k \to +\infty$. □

Proof: Combining the two lines of system (6) gives us that $\theta_1(k) = \theta_2(k) \pm h$, which then translates into

$$(\theta_1(k) - \theta_2(k))^2 = h^2 \tag{8}$$

after the first iteration, i.e. for all $k \in \mathbb{N}^*$. Then, introduce the Lyapunov function $V(k) = \theta_1^2(k) + \theta_2^2(k)$. First, note that $V(k) \leq 5h^2$ whenever $\boldsymbol{\theta} \in M$. Indeed, taking the root $\theta_1(k) = \theta_2(k) + h$ of (8) gives, together with constraints $\max(||\theta_1||, ||\theta_2||)$ of invariant set definition (7), the refined constraints

$$-h \leq \theta_1(k) \leq 2h \tag{9}$$

$$-2h \leq \theta_2(k) \leq h \tag{10}$$

This gives

$$V(k) = 2\theta_2^2(k) + h^2 + 2h\theta_2(k) \leq 5h^2 \tag{11}$$

while the other root $\theta_2(k) = \theta_1(k) + h$ of (8) is the symmetrical case and leads to a result similar to (11) with $\theta_2(k)$ replaced by $\theta_1(k)$. Let us now show that $V(k+1) \leq 5h^2$ as well. Function $V(k+1)$ is given by

$$\begin{aligned} V(k{+}1) = \theta_2^2(k) + (\theta_2(k) \\ +h \operatorname{sign}(\theta_2(k) - \theta_1(k)) \operatorname{sign}\left(J(\theta_2(k)) - J(\theta_1(k))\right))^2 \end{aligned} \tag{12}$$

Because function $J(\cdot)$ is concave and its maximum is assumed to be at the origin, when both $\theta_1(k)$ and $\theta_2(k)$ are positive, we have

$$\begin{aligned} V(k+1) &= \theta_2^2(k) + (\theta_2(k) - h)^2 \\ &= 2\theta_2^2(k) - 2h||\theta_2(k)|| + h^2 \leq 2\theta_2^2(k) + h^2 \leq 5h^2 \end{aligned} \tag{13}$$

when both $\theta_1(k)$ and $\theta_2(k)$ are negative, we obtain a similar result. Having $\theta_1(k)$ and $\theta_2(k)$ of different signs means that, because of constraint (8), we have $-h \leq \theta_2(k) \leq h$, which implies

$$V(k+1) \leq 2\theta_2^2(k) + h^2 + h||\theta_2(k)|| \leq 5h^2 \tag{14}$$

From there, using again constraint (8) and the fact that $V(k+1) \leq 5h^2$ gives inequality

$$\theta_1(k+1)\theta_2(k+1) \leq 2h^2 \tag{15}$$

Taking above expression (18) and inserting as before the first root $\theta_1(k+1) = \theta_2(k+1) + h$ of (8) into it, we get

$$\theta_2^2(k+1) + h\theta_2(k+1) - 2h^2 \leq 0 \tag{16}$$

Solving this quadratic inequality gives $-h \leq \theta_1(k{+}1) \leq 2h$ and $-2h \leq \theta_2(k+1) \leq h$ while the other root of (8) gives $-2h \leq \theta_1(k{+}1) \leq h$ and $-h \leq \theta_2(k{+}1) \leq 2h$. This proves that any trajectory started in M stays there for all $k \in \mathbb{N}^*$. Outside of M, for negative $\theta_1(k)$ and $\theta_2(k)$, concavity of $J(\cdot)$ implies that we have

$$V(k+1) - V(k) = \theta_2^2(k) - \theta_1^2(k) + h^2 - 2h||\theta_2(k)|| \tag{17}$$

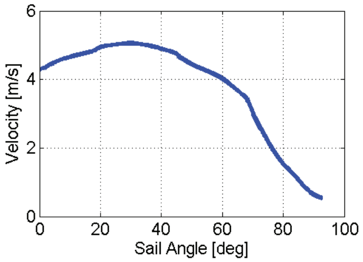


Fig. 2. Surge u versus sail angle δ_s on a reaching course.

Then, using the fact that $(\theta_1 - \theta_2)^2 = \theta_1^2 + \theta_2^2 - 2\theta_1\theta_2 = h^2$, equation (17) gives

$$\begin{aligned} V(k+1) - V(k) &= 2\theta_2^2(k) - 2h||\theta_2(k)|| - 2\theta_1(k)\theta_2(k) \\ &\leq 2||\theta_2(k)|| \left(||\theta_2(k)|| - h\right) \end{aligned} \tag{18}$$

which is strictly negative whenever $||\theta_2(k)|| > h$. Proceeding similarly for positive $\theta_1(k)$ and $\theta_2(k)$ results in the same conclusion. In the case of alternated signs, since constraint (8) is enforced after the first iteration of system (6), then we can come back to one of the above two cases. This concludes the proof of the theorem. ■

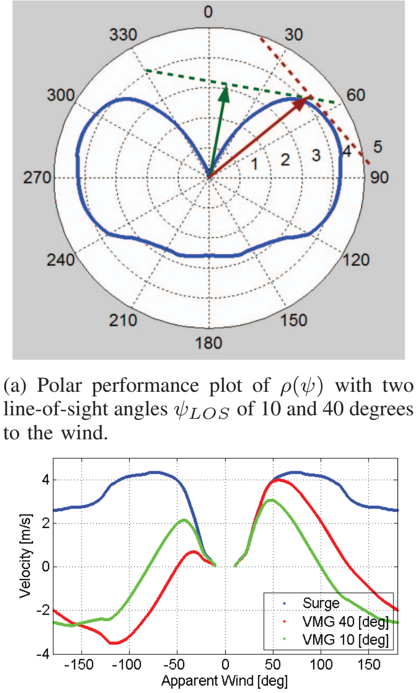
In the case, like it is in the present paper, where the function $\gamma = f(\theta)$ is a part of a larger systems that has dynamics, the hill-climbing algorithm can still be applied. A relatively mild required condition is that the dynamics of the system to be optimised need to be stable. For the hill-climbing algorithm, it means that a new step change $\theta(k)$ can be applied only once the dynamics of the system controlled by the last input (i.e. $\theta(k-1)$) have reached a steady state. Hence this puts constraints on the sampling time T_s for the discrete-time system (4)-(6). Because of the dynamics of the vehicles we consider here, sampling periods T_s of between 30 and 60 seconds are used.

C. Sail angle optimization

Roughly speaking, the amount of propulsion given to a sailing boat is principally dependent on the sail position, i.e. a change on the sail angle δ_s, with respect to the apparent wind α_{aw}, results in a change in surge. This is illustrated in figure 2, where different tunings of δ_s were applied to our 4 DOF yacht model when sailing on a reaching course. Hence, we consider the mapping $u = f_s(\delta_s)$ for the use of the hill-climbing algorithm (4)-(6) for the sail angle optimization scenario. Note that this optimization is performed for long transients of the vehicle, i.e. on constant headings and can take several minutes before converging to an extremum.

D. Velocity-Made-Good optimization

In a sense, the Velocity-Made-Good optimization algorithm can be seen as the dual of the sail angle optimization algorithm, as, in this scenario, we are still trying to optimise the velocity of the vehicle, but this time changing the desired heading ψ_d (see equation (3)) while the sail is maintained at a fixed angle $\delta_s = cst$. The maximum velocity the hill-climbing algorithm is searching for is the so-called Velocity-Made-Good. Roughly speaking, the Velocity-Made-Good,



(a) Polar performance plot of $\rho(\psi)$ with two line-of-sight angles ψ_{LOS} of 10 and 40 degrees to the wind.

(b) Polar performance in cartesian coord. versus v_{VMG} for ψ_{LOS} =10 and 40 degrees.

Fig. 3. Influence of the choice of ψ_{LOS} and its relation to the choice of $\rho(\psi)$ (see expressions (19) and (20)).

well-known to sailors, is the maximum attainable velocity for a particular chosen direction. This vector is not necessarily aligned with the exact direction when one wants to go. Formally, the Velocity-Made-Good is given by

$$\psi_{VMG} = \arg\max_{\psi} \left(\rho(\psi)\cos(\psi - \psi_{LOS})\right) \qquad (19)$$

where constant ψ_{LOS} is the so-called line-of-sight angle representing the chosen direction of travel, while $\rho(\psi)$ represents the steady-state surge velocity that the vehicle will reach on a heading ψ for a specific and constant sail angle δ_s. As an illustration, consider figure 3(a) where the blue curve is a function $\rho(\psi)$ with the two vectors (red and green) represent different line-of-sight directions (resp. 40 and 10 degrees). The dashed perpendicular lines represents the maximum reachable velocity $\max_{\psi}\left(\rho(\psi)\cos(\psi - \psi_{LOS})\right)$ projected with respect to each line-of-sight angle. Figure 3(b) gives a cartesian representation of these quantities for the two line-of-sight angles. For both angles, it can be seen that the maximum is around 50 degrees (see red and blue curves). Hence, in this particular scenario, the hill-climbing algorithm's task is to search for ψ_{VMG} by changing ψ, so that we have the mapping

$$v_{VMG} := f_{VMG}(\psi_d) = \rho(\psi_d)\cos(\psi_d - \psi_{LOS}) \qquad (20)$$

Note that for the particular case where $\psi_{LOS} = 0$, the symmetry of $\rho(\psi)$ results in the function $f_{VMG}(\psi_d)$ having two local maxima, and that one of them will be reached by hill-climbing depending on the initial condition of the search algorithm, i.e. which initial ψ_d we started from.

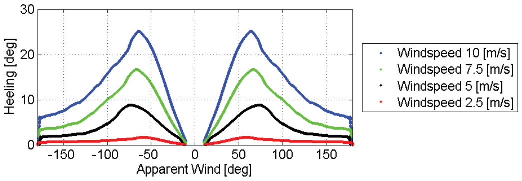


Fig. 4. Heel angle versus apparent wind angle for different wind speeds.

As another remark, since the Velocity-Made-Good measurement is typically obtain from a GPS giving speed over ground, drifts from currents are implicitly taken into account.

Finally, another interesting feature of the VMG optimization scenario is that, in the case where the line-of-sight angle ψ_{LOS} lies *inside* the deadzone, the optimization process results in what can be regarded as a detection algorithm for the limit of the deadzone. Indeed, considering again figures 3(a) and 3(b), one can see that for the value $\psi_{LOS} = 10$ degrees, which is inside the deadzone, the resulting ψ_{VMG} is somewhat on the border of the deadzone. Interestingly, this is obtained without the need for a wind direction sensor, so that, in a sense, wind direction (or more directly the deadzone limit) can be inferred from the VMG optimization process. Deadzone edge tracking can be carried out by choosing the line-of-sight angle to differ from the current heading of a constant value. For example, one can choose $\psi_{LOS} = \psi \pm d$, where $d \in]0\,;\,\pi/6]$.

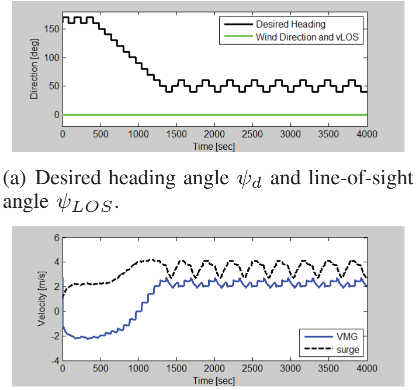
E. Heel angle optimisation

Another way to detect a limit or edge of the deadzone is to look at the evolution of the heel angle versus the heading angle. Basically, for a constant sail angle and wind direction and velocity, the heel angle (its absolute value to be more precise) always goes through a maximum before the vehicle enters the deadzone, as illustrated for our yacht model in figure 4 for several wind speeds. Hence one can search for the edge of the deadzone by applying the hill-climbing algorithm on the mapping $\phi = f_\phi(\psi_d)$. Interestingly, note that this way of detecting the edge of the deadzone does not need a wind direction sensor nor a GPS, it just needs detection of the gravity field in the body frame, then used to infer the roll angle of the vehicle. In this case, one can imagine a fault-tolerant scenario whereby a damaged vehicle could still go upwind despite the loss of a wind direction sensor and a GPS.

IV. Case study

A. Simulation results

All of the above-mentioned scenarios have first been tested on our simulator in a wide range of different conditions. For the sail angle optimization scenario, good results were obtained for $h = 5$ degrees and the sampling period $T_s = 30$ seconds. For Velocity-Made-Good optimization, we simulated the case where the line-of-sight angle was directly upwind, i.e. $\psi_{LOS} = 0$, with control increments on the desired heading ψ_d were of $h = 10$ degrees and the sampling



(a) Desired heading angle ψ_d and line-of-sight angle ψ_{LOS}.

(b) Velocity-Made-Good v_{VMG} and surge velocity u.

Fig. 5. Simulation result of the Velocity-Made-Good optimization scenario.

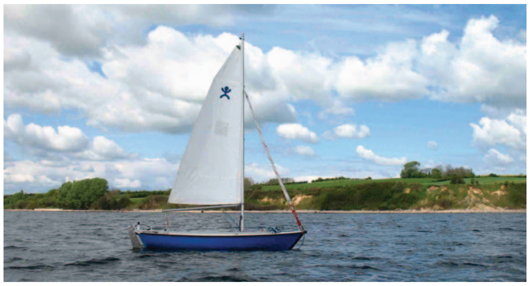
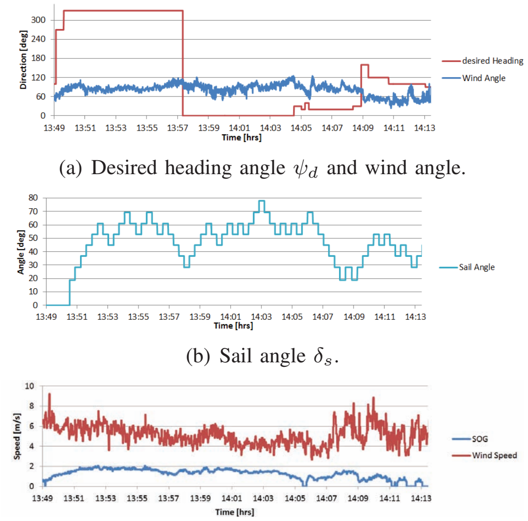


Fig. 6. Our Lærling autonomous sailing vehicle used for the experiments.

period $T_s = 30$ seconds. In this scenario, the sail angle was maintained to a closed haul constant of $\delta_s = 10$ degrees. The results of this particular simulation can be seen in figure 5. Note in figure 5(a) the initial heading $\psi_d(0)$ of the vehicle, which is started on a broad reach (more than 150 degrees), and then gradually goes upwind towards an area close to the deadzone. Similar but more noisy results are obtained for the heel angle optimization scenario, where we have set again $h = 10$ degrees and $T_s = 30$ seconds.

B. Experimental results

The test platform which was used for the experiments is an automated a keel yawl sailing boat of the "Lærling" type equipped with all necessary sensors and actuators, as well as a photovoltaic energy system and wifi communication (see picture in figure 6). This autonomous sailing vehicle is capable of the usual basic features such as sailing autonomously to a desired waypoint of interest, perform mapping of a desired area, detect and avoid obstacles, etc. Several preliminary dynamic tests were conducted in order to assess time of convergence towards steady state, especially in terms of surge velocity. The conclusion of these tests showed quite a similar behavior compared to our simulator and thus allowed to keep the same parameters, for the hill-climbing algorithm, as for the simulations. Note that in all the experiments, measurement values were filtered/averaged to decrease the influence of high-frequency noise on the experimental results.

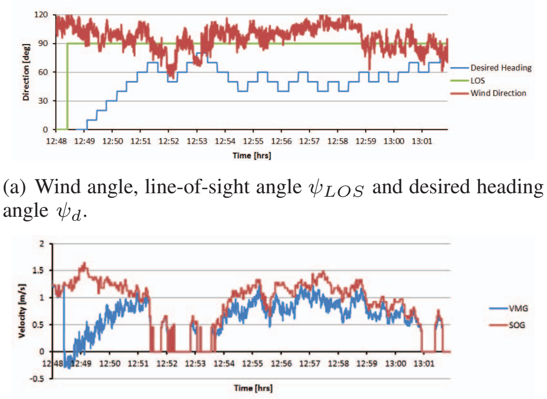


(a) Desired heading angle ψ_d and wind angle.

(b) Sail angle δ_s.

(c) Wind velocity and speed over ground.

Fig. 7. Experimental results for the sail angle optimization scenario.



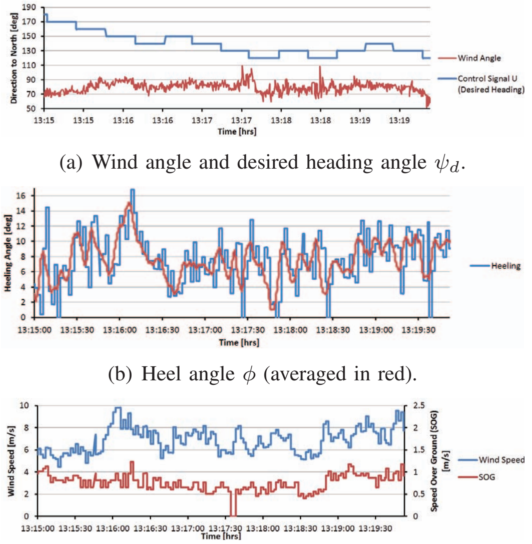
(a) Wind angle, line-of-sight angle ψ_{LOS} and desired heading angle ψ_d.

(b) Speed over ground and Velocity-Made-Good.

Fig. 8. Experimental results for the Velocity-Made-Good optimization scenario.

Experimental results for the sail angle optimization scenario can be seen in figure 7. During this test, the desired heading ψ_d shown in figure 7(a) was changed several times to see how the optimization algorithm would cope with different situations. As can be seen in figure 7(c), the speed over ground (SOG) is increasing after each change in desired heading ψ_d, as for example between times 13:49 and 13:51 for the first heading level. Similar results were consistently obtained for all our subsequent tests of the sail angle optimization scenario.

The Velocity-Made-Good optimization scenario was tested in the situation where the line-of-sight angle ψ_{LOS} is roughly aligned with the wind direction and in any case *inside* the deadzone. The result of the optimization process can be seen in figure 8. Figure 8(a) depicts an initial approach (blue line) towards the wind direction (red line), which ends quite close to the wind direction that we can assume the boat being in the deadzone after around 12:51. This is verified by the velocity drop shown seen in figure 8(b).



(a) Wind angle and desired heading angle ψ_d.

(b) Heel angle ϕ (averaged in red).

(c) Wind velocity and speed over ground.

Fig. 9. Experimental results for the heeling optimization scenario.

The straight deadzone entering can be explained by the rather sudden wind direction change, occurring from 12:51 to 12:52. Although the VMG optimization is not able to adapt to this change immediately, the autonomous sailing vehicle ends up (around 12:54) leaving the deadzone by itself and finds a stable course, until the deadzone is entered again if a too important wind change occurs (see around 13:01). The test has been repeated, approaching the deadzone from the other side, with comparable results. Again the sailing robot might get stuck in the deadzone, but is also able to leave it again by itself. Note that heading upwind is already for a sailor an attention craving task, who can still read the wind direction and interpret the behaviour of the sailing boat better than any algorithm can at the present state. The reason is that small changes in apparent wind angle can result in entering the deadzone. Therefore it can be assessed as a challenge to sail on an upwind course without using the knowledge of wind direction. One way of reducing the risk for entering the deadzone would be to keep a larger distance to it once the edge of the deadzone was detected.

Although in quite good agreement with our simulation tests, the experimental results for the heel optimization scenario are clearly less conclusive. Indeed, and as can be seen in figure 9, if the desired heading angle ψ_d in figure 9(a) converges to a specific interval around 130 degrees, the results are quite sensitive to fluctuations in wind direction and velocity as shown in figure 9(b). However, note that even though the system converges to values ψ_d which makes the vehicle closer to the deadzone, it never actually enters the latter, thus representing, at least on this aspect, an improvement compared to the Velocity-Made-Good scenario.

V. Concluding remarks

In this paper, we have presented a few results on the use of the widely-known hill-climbing algorithm for different online optimization scenarios in robotic sailing. Because of its simplicity and relative robustness, we believe that hill-climbing is a valid alternative to more advanced techniques such as extremum seeking [12]. We have tested our scenarios extensively on our autonomous sailing platform. If the heel angle optimization scenario lacks robustness mostly because of the sensitivity to the wind fluctuations of its pendulum-like dynamics, the sail angle optimization as well as the Velocity-Made-Good scenarios showed good and consistent experimental results and are definitely promising. Future research will look at multi-dimensional online optimization scenarios whereby, for example, both the rudder and sail angle would span the search space.

References

[1] Y. Briere, “IBOAT: An autonomous robot for long-term offshore operation ” in *IEEE Mediterranean Electrotechnical Conference (MELECON)*, Ajaccio, France, 2008.

[2] R. Bruder, B. Stender and A. Schlaefer, “Model sailboats as a testbed for artificial intelligence methods ” in *2nd Int. Robotic Sailing Conference (IRSC)*, Matosinhos, Portugal, 2009.

[3] N. A. Cruz and J. C. Alves, “Auto-heading controller for an autonomous sailboat ” in *IEEE Oceans*, Seattle, WA, 2010.

[4] N. A. Cruz and J. C. Alves, “Navigation performance of an autonomous sailing robot ” in *IEEE Oceans*, St. John’s, NL, 2014.

[5] T. I. Fossen. Marine Control Systems: *Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*, Marine Cybernetics AS, Trondheim, 2002.

[6] R. Goebel, R. G. Sanfelice and A. Teel, “Hybrid dynamical systems,” *IEEE Control Systems*, vol. 29, no. 2, pp. 28–93, 2009.

[7] J. He, L. Xiao and J. Jouffroy, “Towards heading control of an autonomous sailing platform through weight balancing ” in *IFAC Conference in Maneuvering and Control of Marine Craft (MCMC)*, Arenzano, Italy, 2012.

[8] O. L. R. Jacobs and W. M. Wonham, “Extremum control in the presence of noise,” *Journal of Electronics and Control*, vol. 11, no. 3, pp. 193–211, 1961.

[9] L. Jaulin and F. Le Bars, “An interval approach for stability analysis: Application to sailboat robotics,” *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 282–287, 2013.

[10] J. Jouffroy, “A control strategy for steering an autonomous surface sailing vehicle in a tacking maneuver ” in *IEEE Int. Conf. on Systems, Man and Cybernetics (SMC)*, San Antonio, TX, 2009.

[11] R. Leyva, C. Alonso, I. Queinnec, A. Cid-Pastor, D. Lagrange and L. Martinez-Salamero, “MPPT of photovoltaic systems using extremum-seeking control,” *IEEE Transactions on Aerospace and Eletronic Systems*, vol. 42, no. 1, pp. 249–258, 2006.

[12] Y. Tan, D. Nesic, and I. Mareels, “On non-local stability properties of extremum seeking control,” *Automatica*, vol. 42, no. 6, pp. 889–903, 2006.

[13] K. Treichel and J. Jouffroy, “Real-time sail and heading optimization for a surface sailing vessel by extremum seeking control ” in *International Scientific Colloquium (IWK)*, Ilmenau, Germany, 2010.

[14] M. Vidyasagar, *Nonlinear Systems Analysis (2nd ed.)*. Prentice-Hall, 1993.

[15] L. Xiao, J. C. Alves, N. A. Cruz and J. Jouffroy, “Online speed optimization for sailing yachts using extremum seeking ” in *IEEE Oceans*, Hampton Roads, VA, 2012.

[16] L. Xiao and J. Jouffroy, “Modeling and nonlinear heading control for sailing yachts,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 2, pp. 256–268, 2014.

[17] L. Xiao, T. I. Fossen and J. Jouffroy, “Nonlinear robust heading control for sailing yachts ” in *IFAC Conference in Maneuvering and Control of Marine Craft (MCMC)*, Arenzano, Italy, 2012.

[18] W. Xiao and W. G. Dunford, “A modified adaptive hill climbing MPPT method for photovoltaic power systems ” in *IEEE Power Electronics Specialists Conference*, Aachen, Germany, 204.

[19] B. Yang, L. Xiao and J. Jouffroy, “A control-theoretic outlook at the no-go zone in sailing vessels ” in *IEEE Oceans*, Waikoloa, HI, 2012.