



Routing and course control of an autonomous sailboat

Hadi Saoud, Minh Duc Hua, Frédéric Plumet, Faiz Ben Amar

► To cite this version:

Hadi Saoud, Minh Duc Hua, Frédéric Plumet, Faiz Ben Amar. Routing and course control of an autonomous sailboat. European Conference on Mobile Robots (ECMR'15), 2015, Lincoln, United Kingdom. 10.1109/ECMR.2015.7324218 . hal-03135867

HAL Id: hal-03135867

<https://hal.science/hal-03135867>

Submitted on 9 Feb 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Routing and course control of an autonomous sailboat

Hadi Saoud^{1,2}, Minh-Duc Hua^{1,2}, Frédéric Plumet^{2,3} and Faiz Ben Amar^{1,2}

¹ Sorbonne Universités, UPMC Univ Paris 06, UMR 7222, ISIR, F-75005, Paris, France

² CNRS, UMR 7222, ISIR, F-75005, Paris, France

³ Université Versailles St-Quentin, UMR 7222, ISIR, F-75005, Paris, France

Abstract—In this paper, we present a layered control scheme for an autonomous sailboat. The high level control uses an adjusted PRM-Dijkstra algorithm for the routing (global path planning) of autonomous sailboat. This algorithm exploits the sailboat kinematics and wind distribution on the map. For the low level control, we design a new nonlinear course (direction of the velocity vector) controller that exhibits superior performance compared to conventionally used heading controller. A smoothing function is also introduced in the design of the controller to switch easily from course control to heading control, especially when course measurements are noisy at low speed.

I. INTRODUCTION

Autonomous sailboat robots have received in recent years the attention of many researchers as evidenced by the various projects launched worldwide during the last decade [1]–[8]. For these kind of vehicles, the propulsion force is directly extracted from the wind, with only a small amount of energy needed to trim the sails. In addition, they can be equipped with energy harvesting systems such as solar panels or wind turbines. Due to this very low energy consumption (see [9], [10] for examples of power budget), these robots are an attractive solution for long-term autonomy and can be deployed for semi-persistent presence (more than 10 days) and observation or monitoring missions in the oceans. However, the control of an autonomous sailboat (i.e. motion planning and low level control of the heading while ensuring a “good” trimming of the sails) is challenging since the thrust force depends precisely on uncontrollable and partly unpredictable wind. Moreover, such vehicles exhibit complex behaviour due to aero- and hydrodynamic properties of their sails and hull.

Compared with other mobile robots, sailboats exhibit two characteristics that will be discussed in this article:

- First, these vehicles have to travel long distances in environments with few obstacles, except the coasts, but in which the wind characteristics are not accurately known in advance. To plan the motion of the robot, we propose to use a layered architecture (see § II) based on two levels of planning: a first level (global path planning, see § II-A), based on weather forecasts, allows the calculation of a list of waypoints. A second level (local path planning, see § II-B) is dedicated to the real-time computation of the desired direction of motion

and sail trim according to wind conditions measured by onboard sensors.

- Secondly, there exists a sliding motion of the sailboat, mainly due to the sideways force of the wind on the sail. In practice, this implies that the direction of the velocity vector (i.e the *course*) is not aligned with the heading of the boat. Unlike controllers used in previous works (see [1], [3], [8], [11]–[13] among others), which are based on heading control, we propose in § III a new backstepping controller based on the course angle. Eventually, we show in § IV that such a controller allows a more accurate direction following: when using a course controller, the path drawn by the sailboat is closer to the line between two consecutive waypoints than when using a heading controller.

II. GLOBAL CONTROL ARCHITECTURE

The proposed overall control scheme of an autonomous sailing robot is depicted on figure 1.

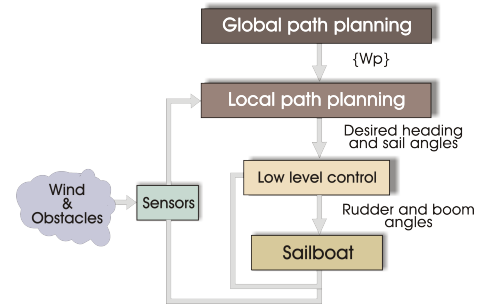


Fig. 1: Overall control scheme for an autonomous sailboat

- **Global path planning:** this mission planning task can be performed once, giving the initial position, final position, known obstacles positions (such as island, fixed mooring,...) and environmental conditions (weather forecast) or periodically (typically every few hours) when new weather data is received. A list of waypoints $\{W_p\}$ and a map with known obstacle positions is then sent to the local path planner.
- **Local path planning:** the aim of this local planning is to actively react to temporally and locally modifications in environment conditions (wind and sea currents) and moving obstacles. Using the list of waypoints, the local path planner computes in real time a feasible direction

to follow as well as a sail angle to reach the current desired waypoint while avoiding obstacles.

- **Low level control:** giving the desired direction to follow, this low level control loop (described in § III) computes the boom and rudder angle of the sailboat.

A. Global path planning using PRM-Dijkstra

We use first the classical “Probabilistic Road Map” (PRM) method [14] to create a graph by generating points randomly distributed in the cruising area. Such methods reduce significantly computing time of path finding compared with a discrete map with constant points distribution (used for example by [15]) mainly because of the lower number of points in a PRM map. Moreover, point generation process can be biased to have more points near zones of interest (like coasts, fixed obstacles, ...). When this map has been generated, algorithms like RRT, A* or Dijkstra can be used to find the best path. In the framework of this work, we adapt the Dijkstra Algorithm [16] to exploit the sailboat kinematics and find an optimal path for the global path planning level.

The optimal path is the one that minimize the cost. Each path being made of several segments, the cost of a path C is the sum of the cost C_{seg_i} of each segment seg_i :

$$C = \sum_{i=1}^k C_{\text{seg}_i}$$

The cost of each segment is not defined by its length but by the time required to come over it. Thus, the cost C_{seg_i} of each segment seg_i is function of its length $|\text{seg}_i|$ and of the sailboat velocity along this segment:

$$C_{\text{seg}_i} = \frac{|\text{seg}_i|}{|\vec{v}_{\text{seg}_i}|}$$

with the hypothesis of constant real wing along a segment. Thus, segment length should be not higher than wind grid resolution (i.e. within segment generation process, all segments with length higher than grid resolution are rejected and can not be used as a part of the path).

Sailboat velocity $|\vec{v}_{\text{seg}_i}|$ is obtained from the polar diagram $P(|\vec{v}_{w,\text{seg}_i}|, \Theta_{\text{seg}_i})$ where $|\vec{v}_{w,\text{seg}_i}|$ and Θ_{seg_i} are respectively the real wind velocity and angle along the segment seg_i in the sailboat frame.

This diagram gives the maximum speed of the sailboat according to the true wind speed $|\vec{v}_w|$ and true wind angle Θ (figure 2). It assumes an optimal sail configuration. The polar diagram of the sailboat used in this study and depicted on figure 2 has been obtained by numerical simulations, in steady state conditions [12]. The direction to follow γ_r is equal to the angle θ_{seg_i} of the segment seg_i w.r.t. inertial frame.

Because segments are not necessarily aligned, this may lead to a large number of tacking or jibbing. To reduce these maneuvers, we introduce a time-dependent cost that depends on the relative angle between two consecutive segments. This additional cost represents the time $t(\text{seg}_{i-1}, \text{seg}_i)$ to switch

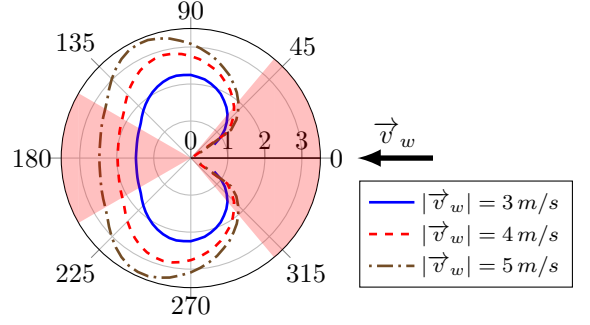


Fig. 2: Polar diagram $P(|\vec{v}_{w,\text{seg}_i}|, \Theta_{\text{seg}_i})$ of a sailboat. Red areas correspond to sailing directions to avoid

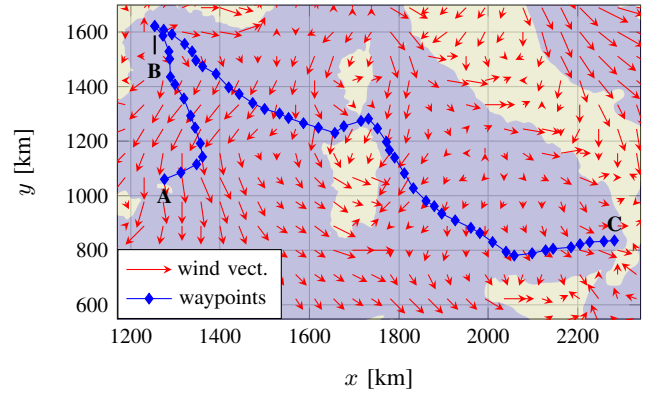


Fig. 3: Waypoints generated by the PRM-Dijkstra algorithm

between two consecutive segments:

$$C_{\text{seg}_i} = \frac{|\text{seg}_i|}{|\vec{v}_{\text{seg}_i}|} + t(\text{seg}_{i-1}, \text{seg}_i)$$

with $t(\text{seg}_{i-1}, \text{seg}_i) = f(\theta_{\text{seg}_{i-1}}, \theta_{\text{seg}_i})$. Time needed for a tacking manoeuvre can be estimated by simulations or extracted from experimental results.

With this method, the resulting path is not necessarily the shortest but will be the least time consuming.

To take into account obstacles, segments that are too close to horizons h_j of the obstacle obs_j are penalized: let c_j be a circle with radius h_j , centered on the obstacle obs_j . The cost of the segment tends to infinity if it intersects the circle c_j :

$$C_{\text{seg}_i} \rightarrow \infty \quad \text{if} \quad \exists j : c_j \cap \text{seg}_i \neq \emptyset$$

This formula can be extended to line obstacles obsline_k :

$$C_{\text{seg}_i} \rightarrow \infty \quad \text{if} \quad \left\{ \begin{array}{l} (\exists j : c_j \cap \text{seg}_i \neq \emptyset) \cup \\ (\exists k : \text{obsline}_k \cap \text{seg}_i \neq \emptyset) \end{array} \right.$$

The resulting path is made up of several segments. Thus, the points delimiting segments can be used as intermediate waypoints for the local path planning level. Mobile obstacles can be processed in the local path planning step instead of the global one to react dynamically to change in their positions.

Figure 3 shows waypoints generated by the algorithm for a sailboat that must reach $B = [1255, 1620]$ then $C = [2281, 834]$ from initial position $A = [1275, 1060]$.

Absolute wind direction and velocity change spatially (wind grid resolution is 0.5°). Wind data are prediction for the 04/20/2015 at 15:00 UTC from the “National Oceanic and Atmospheric Administration”.

We observe from the simulation that the generated waypoints make the sailboat avoid obstacles while keeping a good point of sail.

B. Local path planning

For the local path planning, we use an adaptation of the potential fields method, already published in [17]. Environment and specific sailboat navigation constraints are represented by a local potential built around the vehicle location. Changes of wind direction and detected obstacles affect this periodically updated potential, which guarantees the real-time computation of a feasible direction to follow for the boat. More precisely, an attractive potential is attached to the current waypoint while a repulsive potential is used to represent the no-go zones (shown in red on figure 2) of the sailboat.

III. CONTROLLER DESIGN

A. 3-DOF Dynamic model

A simplified 3-DOF model of the sailboat presented in [12] is used for control design purposes. This assumes that:

- the sailboat has only one mainsail with well identified parameters,
- roll, pitch and vertical dynamics are neglected.

These assumptions are well satisfied for particular design of sailboats for which the restoring torque dominates all the other external torques along the roll and pitch axes. Let $M = \text{diag}([m_{11}, m_{22}, m_{33}])$ be the sailboat mass matrix including real-body and added-mass components. Using the previous assumptions and the approximations $m_{22} \approx m_{33}$, the translational dynamics is given by:

$$\begin{cases} m_{11}\dot{V}_{\text{long.}} = F_{d,1}^{\mathcal{B}} + F_{s,1}^{\mathcal{B}} + F_{r,1}^{\mathcal{B}} + F_{k,1}^{\mathcal{B}} + m_{22}\omega V_{\text{lat.}} \\ m_{22}\dot{V}_{\text{lat.}} = F_{d,2}^{\mathcal{B}} + F_{s,2}^{\mathcal{B}} + F_{r,2}^{\mathcal{B}} + F_{k,2}^{\mathcal{B}} - m_{11}\omega V_{\text{long.}} \end{cases} \quad (1)$$

TABLE I: Notations

Notation	Description
e_1, e_2, e_3	canonical basis of \mathbb{R}^3
$x^{\mathcal{X}}$	vector of coordinates of the affine vector \vec{x} in the basis of the frame \mathcal{X}
$X_{i,n}^{\mathcal{X}}$	n^{th} component of the vector $X_i^{\mathcal{X}} \in \mathbb{R}^3$
G	sailboat center of mass (CoM)
\mathcal{I}	inertial frame chosen as the North-West-Up frame.
	$\mathcal{I} = \{0; \vec{v}_0, \vec{j}_0, \vec{k}_0\}$
\mathcal{B}	body frame fixed to the hull. $\mathcal{B} = \{G; \vec{v}, \vec{j}, \vec{k}\}$
\vec{v}_w	absolute wind vector
\vec{v}	linear velocity of G w.r.t. the inertial frame.
$V_{\text{long.}}, V_{\text{lat.}}$	longitudinal and lateral velocity of the sailboat
σ_r	rudder constant that depends on the water density, rudder aspect, dimension and position
δ_r	rudder angle
ψ	yaw angle of the sailboat
$\omega = \dot{\psi}$	yaw velocity of the sailboat

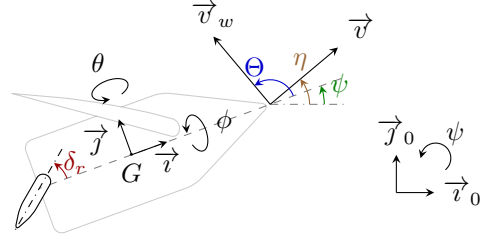


Fig. 4: Frames and angles definition

where $\vec{F}_d, \vec{F}_s, \vec{F}_r, \vec{F}_k$ are respectively hydrodynamic, sail, rudder and keel force vectors, $V_{\text{long.}} = \vec{v} \cdot \vec{v}$ and $V_{\text{lat.}} = \vec{v} \cdot \vec{j}$.

Let $J = \text{diag}([J_{11}, J_{22}, J_{33}])$ be the sailboat inertia matrix including real-body and added-mass components. The rotational dynamics is approximately given by:

$$\begin{cases} \dot{\psi} = \omega \\ J_{33}\ddot{\psi} = \tau_{d,3}^{\mathcal{B}} + \tau_{s,3}^{\mathcal{B}} + \tau_{r,3}^{\mathcal{B}} + \tau_{k,3}^{\mathcal{B}} \end{cases} \quad (2)$$

where $\vec{\tau}_d, \vec{\tau}_s, \vec{\tau}_r, \vec{\tau}_k$: respectively hydrodynamic, sail, rudder and keel torque vectors

From [12], the third component of the rudder torque can be expressed as follows:

$$\tau_{r,3}^{\mathcal{B}} \approx -\sigma_r |\vec{v}|^2 \sin(2\delta_r) \quad (3)$$

where $\sigma_r = 0.5\rho_w S_r r_r C_1$ with ρ_w the water density, S_r the surface of the rudder, r_r the horizontal distance between G and the rotational axis of the rudder and C_1^r the parameter related to the ideal lift and drag coefficients [12].

B. Course control

Due to the sideways force of the wind, the course angle η and heading angle ψ are not necessarily equals. This effect is mainly observed when sailing close hauled or reaching and the leeway angle $\zeta = \eta - \psi$ can be as large as 10° .

Because the sailboat motion is along the course direction, a sailboat with a course controller is expected to diverge less from the line between two waypoints than a heading controller may does.

In the current study, we make the assumption that the torque produced by the rudder can compensate all other external torques around the \vec{k}_0 axis. Thus, for control design, let us simply consider the following second-order system:

$$\ddot{\psi} = u + c(t) \quad (4)$$

with $u := \tau_{r,3}^{\mathcal{B}}/J_{33}$ the control input assumed to be unbounded for control design purpose, and $c(t) := (\tau_{s,3}^{\mathcal{B}} + \tau_{k,3}^{\mathcal{B}} + \tau_{d,3}^{\mathcal{B}})/J_{33}$ the perturbation term assumed to be slowly time-varying (i.e. $\dot{c} \approx 0$) so that it can be compensated by an integral action.

Thus, one can deduce from the yaw dynamic (4) and the rudder torque equation (3) the value of the rudder angle as function of the control input u :

$$\delta_r = -\frac{1}{2} \arcsin \left(\text{sat}_1 \left(\frac{J_{33}u}{\sigma_r |\vec{v}|^2} \right) \right) \quad (5)$$

The proposed controller will regulate the unit vector $\vec{\gamma}$: the objective is to make $\vec{\gamma}$ tracks $\vec{\gamma}_r$ where $\vec{\gamma}_r$ is the unit vector collinear to the course. The control design is based on the backstepping procedure.

To simplify expressions, let's define the following angles:

- $\gamma_r \triangleq \angle \gamma_r^{\mathcal{I}}$ is the direction angle to follow (reference)
- $\gamma \triangleq \angle \gamma^{\mathcal{I}} = \psi + \angle \gamma^{\mathcal{B}}$

The control objective is the stabilization of γ about γ_r or equivalently, the stabilization of the error angle $\tilde{\gamma} \triangleq \gamma - \gamma_r$ around zero. However, since there is no distinction between $\tilde{\gamma}$ and $\tilde{\gamma} + k2\pi$ with $k \in \mathbb{N}$, it's more relevant to stabilize $\sin \tilde{\gamma}$ around zero.

To achieve this objective, a first energy function \mathcal{V}_1 is proposed:

$$\begin{aligned}\mathcal{V}_1 &= 1 - \cos \tilde{\gamma} \\ \dot{\mathcal{V}}_1 &= (\omega + \omega_\gamma - \omega_r) \sin \tilde{\gamma}\end{aligned}$$

where ω_γ and ω_r are the rotational dynamics of respectively $\gamma^{\mathcal{B}}$ and $\gamma_r^{\mathcal{I}}$. Let ω_d be the desired rotational speed, we modify the energy function, introducing ω_d in its derivative:

$$\dot{\mathcal{V}}_1 = (\omega + \omega_\gamma - \omega_r - \omega_d + \omega_d) \sin \tilde{\gamma}$$

Thus, the desired rotational speed is expressed as:

$$\omega_d = -\omega_\gamma + \omega_r - K_{\omega_d} \frac{\sin \tilde{\gamma}}{1 + \cos \tilde{\gamma}} \quad (6)$$

With this value of ω_d , the dynamic \mathcal{V}_1 becomes:

$$\dot{\mathcal{V}}_1 = (\omega - \omega_d) \sin \tilde{\gamma} - 2K_{\omega_d} \sin^2 \frac{\tilde{\gamma}}{2}$$

The tracking of ω_d is ensured by the storage function \mathcal{V}_2 :

$$\begin{aligned}\mathcal{V}_2 &= \mathcal{V}_1 + \frac{1}{2K_p}(\omega - \omega_d)^2 \Rightarrow \\ \dot{\mathcal{V}}_2 &= \dot{\mathcal{V}}_1 + \frac{1}{K_p}(\omega - \omega_d)(\dot{\omega} - \dot{\omega}_d) \\ &= -2K_{\omega_d} \sin^2 \frac{\tilde{\gamma}}{2} + \frac{1}{K_p}(\omega - \omega_d) \underbrace{(\dot{\omega} - \dot{\omega}_d)}_{u+c} + K_p \sin \tilde{\gamma}\end{aligned}$$

The input u is chosen as:

$$u = \dot{\omega}_d - K_p \sin \tilde{\gamma} - K_d(\omega - \omega_d) - \hat{c} \quad (7)$$

where \hat{c} is the estimation of the unknown perturbations c . The dynamic of the estimation is defined as:

$$\dot{\hat{c}} = K_{i_\omega}(\omega - \omega_d), \quad \hat{c}(0) = 0 \quad (8)$$

with K_{i_ω} positive gain.

To check the stability of the system with the estimation of the perturbations, let's write the following Lyapunov function:

$$\mathcal{V} = \mathcal{V}_2 + \frac{1}{2K_p K_{i_\omega}}(c - \hat{c})^2$$

Its derivative is:

$$\dot{\mathcal{V}} = -\frac{K_d}{K_p}(\omega - \omega_d)^2 - 2K_{\omega_d} \sin^2 \frac{\tilde{\gamma}}{2} \leq 0$$

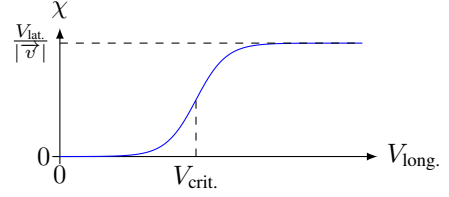


Fig. 5: Smoothing function χ

From here, by application of LaSalle's theorem, one deduces that $\dot{\mathcal{V}}$ converges to zero, which in turn implies the convergence of γ to γ_r and of ω to ω_d .

To avoid singularity in the equation (6), we must add a saturation:

$$\omega_d = -\omega_\gamma + \omega_r - K_{\omega_d} \frac{\sin \tilde{\gamma}}{\max(1 + \cos \tilde{\gamma}, \varepsilon)} \quad (9)$$

where $\varepsilon \rightarrow 0^+$

Finally, the rudder angle δ_r is given by (5), using the control law (7) and (9).

In the 3DOF model, the input of the system u is $\tau_{r,3}^{\mathcal{B}}/J_{33}$ and does not depend on heel angle ϕ . In practice, a real world sailboat is affected by the heeling. To take this effect into account, one can weight the input u by $\cos \phi$ i.e. on a heeling sailboat, more torque must be generated on the rudder to obtain the same input u that will stabilize the direction error $\tilde{\gamma}$ around zero.

$$u = \frac{\tau_{r,3}^{\mathcal{B}} \cos \phi}{J_{33}}$$

Thus, the value of the rudder angle as a function of the control input u and heeling angle ϕ is given by:

$$\delta_r = -\frac{1}{2} \arcsin \left(\text{sat}_1 \left(\frac{J_{33} u}{\sigma_r |\vec{v}|^2 \cos \phi} \right) \right) \quad (10)$$

The main drawback of this controller is that the course is not well defined at low velocity because of the measurement noise of the sensors (IMU and GPS). Also, when velocity vanishes, the course becomes undefined leading to a singularity in the controller. To overcome these problems, a smooth switching function will be introduced.

C. Smooth heading and course switching

The previous controller allows the tracking of $\vec{\gamma}_r$ by $\vec{\gamma}$, with $\vec{\gamma}$ being implicitly the current course vector. However, one can use a different definition of $\vec{\gamma}$ while still using the same controller.

In order to make $\vec{\gamma}$ well defined, whatever the sailboat velocity is, let's redefine $\gamma^{\mathcal{B}}$ using $\bar{\gamma}^{\mathcal{B}} \in \mathbb{R}^2$ as follow:

$$\gamma^{\mathcal{B}} = [\bar{\gamma}^{\mathcal{B}^\top}, 0]^\top \text{ with } \bar{\gamma}^{\mathcal{B}} = \begin{bmatrix} \sqrt{1 - \chi^2} \\ \chi \end{bmatrix}$$

$\chi \in [0; \frac{V_{\text{lat}}}{|\vec{v}|}]$ is a function as in figure 5 and with the following properties:

- if $V_{\text{long}} > V_{\text{crit}}$, $\vec{\gamma} \rightarrow \frac{\vec{v}}{|\vec{v}|}$, i.e. $\gamma \equiv$ course,
- if $V_{\text{long}} < V_{\text{crit}}$, $\vec{\gamma} \rightarrow \frac{\vec{v}}{v_B}$, i.e. $\gamma \equiv$ heading.

χ can be defined, for example, using a sigmoid function $\text{sig}(x) = 1/(1 + e^{-\lambda x})$ with λ some positive constant:

$$\chi = \frac{V_{\text{lat.}}}{|\vec{v}|} \text{sig}(V_{\text{long.}} - V_{\text{crit.}})$$

Then, the derivative of $\bar{\gamma}^B$ becomes:

$$\dot{\bar{\gamma}}^B = \begin{bmatrix} \frac{-\chi \dot{\chi}}{\sqrt{1-\chi^2}} \\ \dot{\chi} \end{bmatrix} = -\omega_\gamma \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \bar{\gamma}^B \text{ and } \omega_\gamma \triangleq \frac{-\dot{\chi}}{\sqrt{1-\chi^2}}$$

With this definition of $\bar{\gamma}^B$, one obtains a smooth switching controller that acts as a heading one when sailboat velocity is low (i.e. when noise effects are the highest) or as a course one elsewhere.

The choice of $V_{\text{crit.}}$ has an influence on the controller behaviour. Setting it to a value higher than the maximal velocity of the sailboat will disable the switching ability and turn the controller to a heading one. Setting it to a negative value will also disable the switching ability but will turn the controller to a pure course one (with the risk of falling in the singularity case when $|\vec{v}| \rightarrow 0$). To enable the smooth switching and the course following whatever $|\vec{v}|$ is, $V_{\text{crit.}}$ must be positive. It's value must be chosen depending on the sailboat sensors characteristics to ensure that when $|\vec{v}| > V_{\text{crit.}}$, the signal-to-noise ratio of \vec{v} is high enough, allowing a precise computation of the course.

IV. SIMULATION RESULTS

First, simulations have been done to compare the course controller and the heading controller. In these simulations, the sailboat must reach a waypoint located 5000 m away. Absolute wind is coming from the west and its velocity is 4 m/s. In one simulation, a course controller is used ($V_{\text{crit.}} = 0.5 \text{ m/s}$) while in the other simulation, $V_{\text{crit.}} \rightarrow \infty$ to obtain a pure heading controller ($\vec{\gamma}$ is always collinear to the heading). All other parameters and initial conditions are the same. In both cases, reference direction is computed by the potential field method as presented in [17] and sailboat trimming method is a linear one as in [3].

We notice from figure 6 that crosstrack error from the line between the two consecutive waypoints is more important for the heading controller. In this case, the maximal distance is 256 m while it's only 20 m with a course controller. Because doing a real path following is hard to execute on a sailboat, this maximal distance may be of first concern when the goal of a sailing mission is to perform a spatially controlled measurements.

Other simulations have been run to observe transient heading and course behaviour of each controller (figure 7). The simulation parameters (initial and final position, wind, ...) are the same as in the previous one, except that a normally distributed noise with variance $\sigma = 0.02$ is added to the linear velocity measurements in order to simulate a noisy velocity sensor. No noise is added to the heading since, on an IMU, the heading estimation mainly based on magnetometers is usually more accurate.

We define three different controllers:

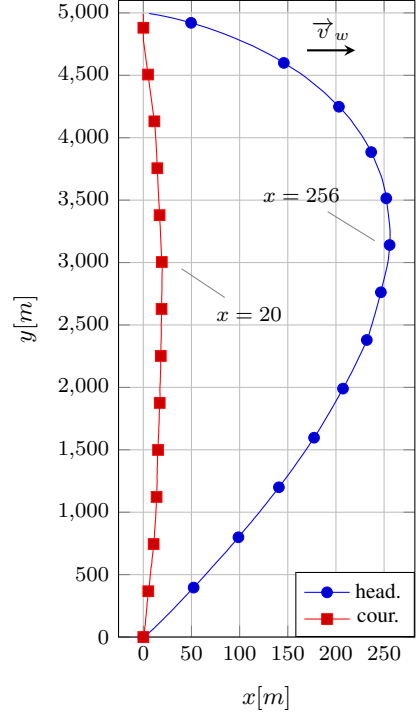


Fig. 6: Sailboat trajectories with a course controller (cour.) and a heading controller (head.)

- a course controller with smooth switching towards a heading controller at low speed ($V_{\text{crit.}} = 1.3 \text{ m/s}$)
- a pure course controller by disabling the switching function ($V_{\text{crit.}} \ll 0 \text{ m/s}$)
- a heading controller ($V_{\text{crit.}} \rightarrow \infty$)

As expected with heading controllers, final heading reaches the reference ($\psi_{\text{final}} \rightarrow 0$) while the final course $\gamma_{\text{final}} = 5.5^\circ$ because of the leeway effect. When using a course controller (with or without the switching function), the final course $\gamma_{\text{final}} \rightarrow 0^\circ$ because controller balance leeway effect. One can observe that before critical velocity $V_{\text{crit.}}$ is reached, the smooth switching course controller corrects heading angle. When $|\vec{v}| \approx V_{\text{crit.}}$, it begins correcting route angle.

Because of the relationship between the velocity $|\vec{v}|$ and the rudder angle δ_r , measurement noise affects the controller. A course controller is more affected because, which such a controller, $\vec{\gamma}$ is a function of \vec{v} . The effects of \vec{v} on the controller are mainly noticeable at low velocity: one can observe from figure 7 that, when $t < 5 \text{ sec}$, the rudder with a pure course controller oscillates. These oscillations are less observed on the smooth switching course controller because, for that time interval, sailboat velocity is under $V_{\text{crit.}}$ and γ is equal to the heading.

V. CONCLUSIONS

In this paper, an layered control scheme for an autonomous sailboat is presented. This scheme includes 1) a global path planning for waypoints generation giving mission objectives and environmental conditions, 2) a local path planning to

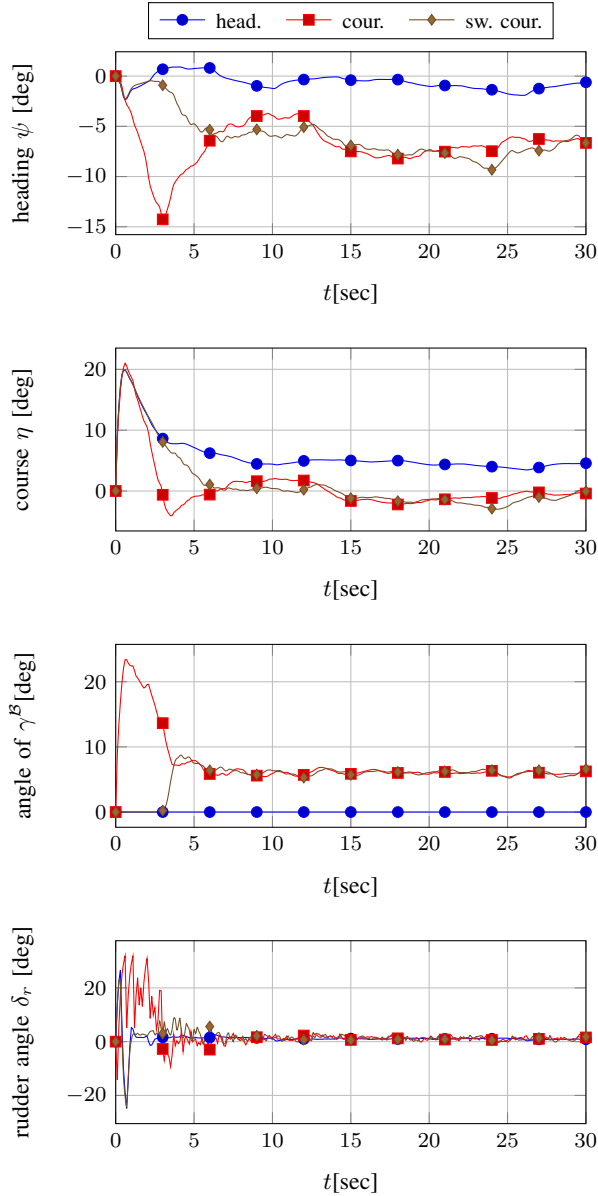


Fig. 7: Comparison between a heading controller (head.), a pure course controller (cour.) and a smooth switching course controller (sw. cour.)

actively react to moving obstacles and changes in environmental conditions and 3) a low level control to compute boom and rudder angles for the sailboat.

A main contribution of the article is the development of an adjusted Dijkstra algorithm to perform global path planning for a sailboat. This adjusted algorithm exploits the sailboat kinematics and wind distribution on the map. With these modifications, Dijkstra method is able to find the less time consuming path between an initial and a final position, generating waypoints used by the local path planning level.

Another main contribution is the design of a smooth

switching course controller that can switch from heading tracking to course tracking depending on the sailboat velocity and measurements noise. We show that a course control allows a better following of the direction toward a waypoint than a heading control. The advantage of the smooth switching controller is that it acts as a heading one when course is hardly defined because of the slow velocity of the sailboat and the noise in measurements. On the other hand, when velocity increases and noise effects decrease, smooth switching controller acts as a course one allowing good following of the direction.

Future works will focus on designing a 4-DOF course controller that act on both sail and rudder to maximize sailboat velocity while controlling its course.

REFERENCES

- [1] G. Elkaim, "The atlantis project: A gps-guided wing-sailed autonomous catamaran," *Journal of the Institute of Navigation*, vol. 53, pp. 237–247, 2006.
- [2] M. Neal, "A hardware proof of concept of a sailing robot for ocean observation," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 2, pp. 462–469, 2006.
- [3] Y. Briere, "IBOAT: An autonomous robot for long-term offshore operation," in *14th IEEE Mediterranean Electrotechnical Conf. MELECON*, 2008, pp. 323–329.
- [4] N. Cruz and J. Alves, "Autonomous sailboats: An emerging technology for ocean sampling and surveillance," in *MTS/IEEE OCEANS*, 2008.
- [5] R. Stelzer, K. Jafarmadar, H. Hassler, R. Charwot, *et al.*, "A reactive approach to obstacle avoidance in autonomous sailing," in *Int. Robotic Sailing Conf.*, 2010, pp. 33–39.
- [6] H. Erckens, G.-A. Busser, C. Pradalier, and R. Siegwart, "Avalon: Navigation strategy and trajectory following controller for an autonomous sailing vessel," *IEEE Robotics Automation Mag.*, vol. 17, no. 1, pp. 45–54, 2010.
- [7] C. Petres, M.-A. Romero-Ramirez, F. Plumet, and B. Alessandrini, "Modeling and reactive navigation of an autonomous sailboat," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011, pp. 3571–3576.
- [8] 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.
- [9] Y. Briere, "Between the drifting buoy and the autonomous sailing boat : the microtransat concept," in *Int. Autonomous Surface Ship Symposium*, 2008, pp. 1–10.
- [10] C. Schröder and L. Hertel, "Development of a low-budget robotic sailboat," in *Int. Robotic Sailing Conf.*, 2013, pp. 13–24.
- [11] R. Stelzer and T. Prohl, "Autonomous sailboat navigation for short course racing," *Robotics and Autonomous Systems*, vol. 56, no. 7, pp. 604–614, 2008.
- [12] H. Saoud, M.-D. Hua, F. Plumet, and F. Ben Amar, "Modeling and control design of a robotic sailboat," in *Int. Robotic Sailing Conf.* Springer, 2013, pp. 95–110.
- [13] L. Xiao and J. Jouffroy, "Modeling and nonlinear heading control of sailing yachts," *IEEE Journal of Oceanic Engineering*, vol. 39, pp. 256–268, 2014.
- [14] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [15] J. Langbein, R. Stelzer, and T. Frühwirth, "A rule-based approach to long-term routing for autonomous sailboats," in *Int. Robotic Sailing Conf.*, 2011, pp. 195–204.
- [16] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [17] C. Pêtrès, M. Romero-Ramirez, and F. Plumet, "A potential field approach for reactive navigation of autonomous sailboats," *Robotics and Autonomous Systems*, vol. 60, no. 12, pp. 1520–1527, 2012.