Sailboat as a Windmill

Luc Jaulin and Fabrice Le Bars

Abstract. This paper proposes to transform a sailboat robot into a big wind turbine (or windmill) corresponding to the boat itself. The main idea is to make the sailboat rotating as fast as possible. When the wind open the sail, the mainsheet is able to pull a generator in order to produce electric energy. The resulting controller is simple to implement and its parameters are easy to tune. A simulated test-case shows that the proposed technique could generate an average power of approximatively 100W.

1 Introduction

Sailboat robots (see e.g. [17] [16] [6] [2] [3]) need energy for the actuators, for the sensors [20], for the embedded computer and for communication [21] [5]. Sonar panels cannot be considered as sufficient in many situations (during the night, or in cloudy areas) and we would like to consider other sources of energy that do not depend on the sun. A wind turbine or water turbine have sometimes been used, but the energy brought cannot be considered as significant [19]. In this paper, we propose to use the sailboat itself as a huge wind turbine, or equivalently to reconstruct a mobile windmill. The windmill behavior of the robot assumes the boat is in a station keeping mode. Such a mode can be chosen in case where the robot has to wait for a rendezvous, or when the robot has its batteries almost empty. We assume here that the robot has only two actuators: the rudder and a blocker for tuning the sail. The corresponding controller is illustrated by Figure 1, where u_1, u_2 correspond to the inputs (i.e., the rudder angle and the tuning of the sail) and $\mathbf{m}, \theta, \psi$ are the outputs (i.e., the GPS, the compass and the weather vane). If we consider that

Luc Jaulin · Fabrice Le Bars
IHSEV, Lab-STICC, OSM, Pôle STIC, ENSTA Bretagne, 2 Rue F. Verny,
29806 Brest, France
e-mail: {Luc.Jaulin,Fabrice.Le_Bars}@ensta-bretagne.fr

F. Le Bars and L. Jaulin (eds.), *Robotic Sailing 2013*,
DOI: 10.1007/978-3-319-02276-5_7, © Springer International Publishing Switzerland 2014

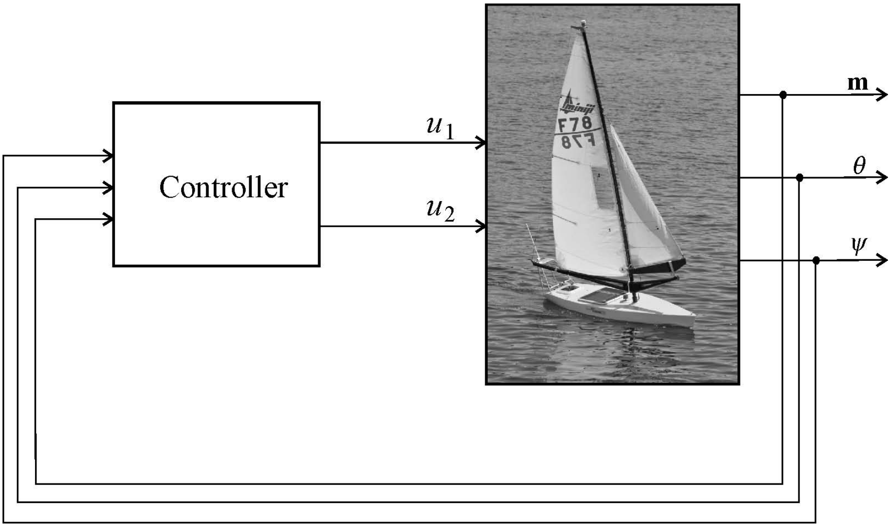

Fig. 1 The controller makes the robot rotating on itself as a windmill in order to produce energy

the blocker does not consume any energy, the only energy used for control is the rudder which consume less than 0.1W, if it is well balanced [21]. When the locker is open and the sail is opening pushed by the wind, the positive power delivered by the wind through the sail can be collected by a generator and stored inside batteries. A spring makes it possible to maintain the mainsheet tight, i.e., when the sail is in a flag mode the spring will rewind the mainsheet and close the sail. The power can be collected either at the mainsheet level via a winch or at the mastfoot. The purpose of this paper is to demonstrate the feasibility of the approach and to evaluate the amount of energy we could expect to collect with this technique.

The paper is decomposed as follows. Section 2 presents a model for the sailboat taking into account the energy and the blocker. Section 3 proposes a control strategy giving the robot windmill like behavior to produce energy. Some simulated experiments detailed on Section 4 show that it is possible to solve the station keeping problem [4] while collecting an average of 100W for the batteries.

2 State Space Model

Different types of models exists for sailboats [8], [9] [10]. To our knowledge, the most accurate one has been provided by Xiao and Jouffroy [23]. Here, to describe the dynamic of the sailboat robot, we propose a model that is sufficiently accurate to illustrate the behavior of our controller and able to give an approximation of the energy that could be collected. Classically, a state space model for a robot has the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$$

where $\mathbf{x}$ is the state vector and $\mathbf{u}$ is the input vector. Sometimes, it is more convenient to write this state equation under the form

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, \mathbf{z}, \mathbf{u}) \\ \mathbf{z} = \mathbf{h}(\mathbf{x}, \mathbf{u}) \end{cases}$$

where

$$\mathbf{g}(\mathbf{x}, \mathbf{z}, \mathbf{u}) = \mathbf{g}(\mathbf{x}, \mathbf{h}(\mathbf{x}, \mathbf{u}), \mathbf{u}) = \mathbf{f}(\mathbf{x}, \mathbf{u}).$$

The vector $\mathbf{z}$ contains link variables which are intermediate variables that are used to shorten the equation. Link variables can correspond to forces, angles, ... and are often needed for the simulation to draw the robot and also to control that some feasibility state constraints are satisfied.

Model. The model is given by the following state space equations (see Figure 2).

$$\begin{cases} \text{(i)} & \dot{x} = & v\cos\theta + p_1 a\cos\psi \\ \text{(ii)} & \dot{y} = & v\sin\theta + p_1 a\sin\psi \\ \text{(iii)} & \dot{\theta} = & \omega \\ \text{(iv)} & \dot{v} = & \frac{f_s\sin\delta_s - f_r\sin u_1 - p_2 v^2}{p_9} \\ \text{(v)} & \dot{\omega} = & \frac{f_s(p_6 - p_7\cos\delta_s) - p_8 f_r\cos u_1 - p_3\omega v}{p_{10}} \\ \text{(vi)} & \dot{\ell} = & u_2 \text{ if } \gamma > 0 \\ \text{(vii)} & \dot{E} = & p_6\ |f_s|\ u_2 \end{cases} \tag{1}$$

where the link variables are given by

$$\begin{cases} \text{(viii)} & \mathbf{w}_{\text{ap}} & = & \begin{pmatrix} a\cos(\psi - \theta) - v \\ a\sin(\psi - \theta) \end{pmatrix} \\ \text{(ix)} & \psi_{\text{ap}} & = & \operatorname{atan2}(\mathbf{w}_{\text{ap}}) \\ \text{(x)} & a_{\text{ap}} & = & \|\mathbf{w}_{\text{ap}}\| \\ \text{(xi)} & \gamma & = & \cos\psi_{\text{ap}} + \cos\ell \\ \text{(xii)} & \ell & = & |\delta_s| \text{ if } \gamma \leq 0 \\ \text{(xiii)} & \delta_s & = & \begin{cases} -\tan^{-1}(\tan\psi_{\text{ap}}) & \text{if } \gamma \leq 0 \\ -\ell\,\operatorname{sign}(\sin\psi_{\text{ap}}) & \text{otherwise} \end{cases} \\ \text{(xiv)} & f_s & = & p_4 a_{\text{ap}}\sin(\delta_s - \psi_{\text{ap}}) \\ \text{(xv)} & f_r & = & p_5 v\sin u_1 \end{cases}$$

This model is close to the models developed in [11], except that here, *(a)* we added the direction of the wind ψ and its amplitude a as parameters, *(b)* the control is not anymore the sail angle, but the length of the mainsheet, which is more realistic, *(c)* the speed of the robot is not considered as small

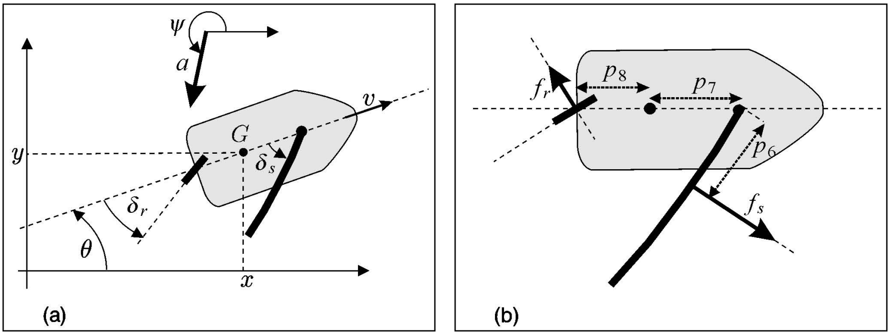

Fig. 2 Sailboat to be used as a windmill

compared to the true wind (the notion of apparent wind has thus to be introduced), *(d)* the angular friction now depends on the speed, which is more consistent with actual sailboats and *(e)* the length ℓ of the mainsheet and the energy of the batteries E are introduced as state variables. All quantities are expressed using the international unit system. For simplicity, the lenght ℓ of the mainsheet is expressed in radian (rad). ℓ corresponds to the absolute value of the maximal angle δ_s that could reach the sail when the mainsheet is tight. Let us now describe more deeply all variables involved in this model.

Inputs. The sailboat has two inputs. The first input $u_1 = \delta_r$ is the angle between the rudder and the sailboat. The second input u_2 corresponds to the blocker. When $u_2 = 1$, the locker is unblocked and the length of the mainsheet ℓ may increase (if the direction of the wind allows it). Otherwise, $u_2 = 0$ and the blocker is active.

State variables. The state variable occurring in our model (1) are $x, y, \theta, v, \omega, \ell, E$ where (x, y) are coordinates of the robot, θ is its heading, v is its speed along the main axis, ω is its rotational speed. The energy of the batteries E will increase with time. The length of the mainsheet ℓ corresponds of the maximal angle of the sail. In the particular case where the mainsheet is tight, it can be computed from the state variables θ, u_2, ψ, v and thus it cannot be considered as a state variable anymore. Therefore, the dimension of the state vector (either 6 or 7) changes with time. The sailboat thus corresponds to an hybrid system.

Parameters. In our model, p_1 is the drift coefficient, p_2 is the tangential friction, p_3 is the angular friction, p_4 is the sail lift, p_5 is the rudder lift, p_9 is the mass of the boat and p_{10} is the mass moment of inertia. The distances p_6, p_7, p_8 are represented in Figure 2. All parameters p_i are assumed to be known exactly. Two other quantities should also be considered as parameters: the speed a of the wind and its direction ψ.

Link variables. These variables are used to shorten the expression of the state equations. *(viii)* The vector $\mathbf{w}_{\text{ap}}$ corresponds to the apparent wind expressed in the robot frame. The amplitude *(ix)* and the angle *(x)* of $\mathbf{w}_{\mathrm{ap}}$ (in the robot frame) are denoted by a_{ap} and ψ_{ap}. *(xi)* The coefficient γ is positive if the mainsheet is tight. *(xii)* In this case, ℓ is a state variable and its evolution obeys to the differential equation $\dot{\ell} = u_2$. Otherwise, the mainsheet is tight, ℓ is a link variable and its value is equal to $|\delta_s|$. This change of status of ℓ is typical of what happen for hybrid systems. *(xiii)* When the mainsheet is not tight, the angle of the sail δ_s, is equal to $-\psi_{\mathrm{ap}} \pm 2k\pi$ and it behaves as a flag. Since we want $\delta_s \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, we have written $\delta_s = -\tan^{-1}\left(\tan\left(\psi_{\mathrm{ap}}\right)\right)$. When the mainsheet is not tight, δ_s is determined by ℓ and the direction of the apparent wind. *(xiv)* f_s represents the force of the wind on the sail and *(xv)* f_r is the force of the water on the rudder.

State equations. The two first equations *(i),(ii)* of (1) express that the boat follows its heading, but always looses with respect to the wind. Equations *(iv)* and *(v)* are obtained using the Newton laws. Equation *(vi)* tells us that the length ℓ of the mainsheet can only increases when the sail is inflated and when the blocker is off (i.e., $u_2 = 1$). Equation *(vii)* provides the power delivered to the batteries: when $u_2 = 1$, and $f_s \neq 0$, the sail opens with an angular velocity of $1\mathrm{rad.sec}^{-1}$ and the power collected is $p_6 \;\; |f_s|$. Figure 3 represents the differential graph of the state equations. The state variables are represented by grey nodes and the inputs by square nodes. The integral relations are represented by bold arrows and the link relation by dotted arrows. The two bold dotted arrows illustrate that the differential dependency between u_2, γ and ℓ are valid for some conditions only.

Note that this model for the sailboat could be made more realistic by adapting the modeling tools described by Fossen in the context of marine vessel [7] to sailboats. But to our knowledge, realistic state equations for sailboats do not exist yet.

3 Controller

A classical approach to build controllers is to take a realistic model of the system to be controlled (such as [8] for the control of sailboats) and then to use classical control methods to get the controller. Here, we follow a pragmatic approach influenced by the potential field strategy proposed by [18] for sailboat robots. Our sailboat robot is assumed to have three sensors and two actuators. The controller will have some parameters which are easy to tune, two state variables $q \in \{1, 2, 3, 4\}$, and $t_0 \in \mathbb{R}^+$, two outputs $u_1 \in [-\frac{\pi}{4}, \frac{\pi}{4}]$, $u_2 \in \{0, 1\}$ and three inputs $\mathbf{m} \in \mathbb{R}^2, \theta \in \mathbb{R}, \psi \in \mathbb{R}$. Let us now describe all these variables.

Sensors (which also correspond to the input of the controller). The heading θ of the robot is measured by a compass. The angle of the wind ψ is returned by a weather vane (even if this sensor can sometimes be omitted

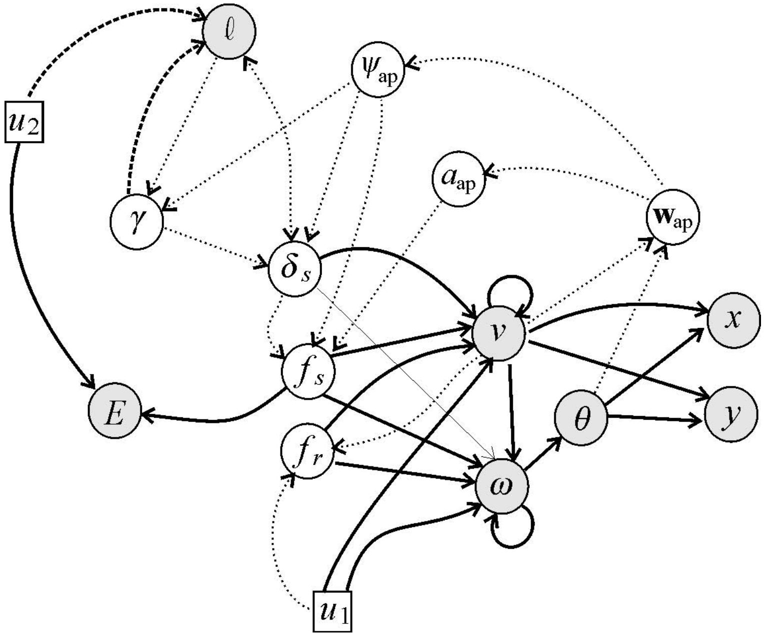

Fig. 3 Differential graph of our sailboat robot

as shown in [22]). The position $\mathbf{m}$ is given by a GPS. These sensors are the inputs of our controller.

Actuators (which also correspond to the output of the controller). The inputs of the robot are the angle $u_1 \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ of the rudder and the blocker $u_2 \in \{0,1\}$ which makes it possible to tune indirectly the length of the mainsheet.

Parameters. $\delta_r^{\max}$ is the maximal angle of the rudder is taken as $\delta_r^{\max} = \frac{\pi}{4}$. ζ is the close hauled angle. For the simulation, we will choose $\zeta = 1$rad.

State variable. The discrete variable $q \in \{1,2,3,4\}$, gives three modes: the *traction* $(q = 1)$, the *rewind* $(q = 2)$ and the *positioning* $(q = 3, q = 4)$. The start time t_0 corresponds to the time at which the timer is started when the controller is in the positioning mode.

The basic idea of the controller is to decompose the plane into three cones, the intersection of which corresponds to the origin, as illustrated by Figure 4. In the *mill cone* (painted gray), the robot rotates as a windmill to produce energy. In the mill cone, the robot slowly moves downwind. The points $\mathbf{m}$ that are inside the mill cone satisfy the inequality $\cos(\psi - \arg(\mathbf{m})) > -\cos\zeta$. In the hatched cone, the controller will carry favor to the close-hauled heading $\pi + \psi - \zeta$. In the white cone, it will prefer the heading $\pi + \psi + \zeta$.

We now give the details of the controller which is clearly influenced by the line following controller proposed in [14] [12] and already experimented [15] on the sailboat robot *VAIMOS*.

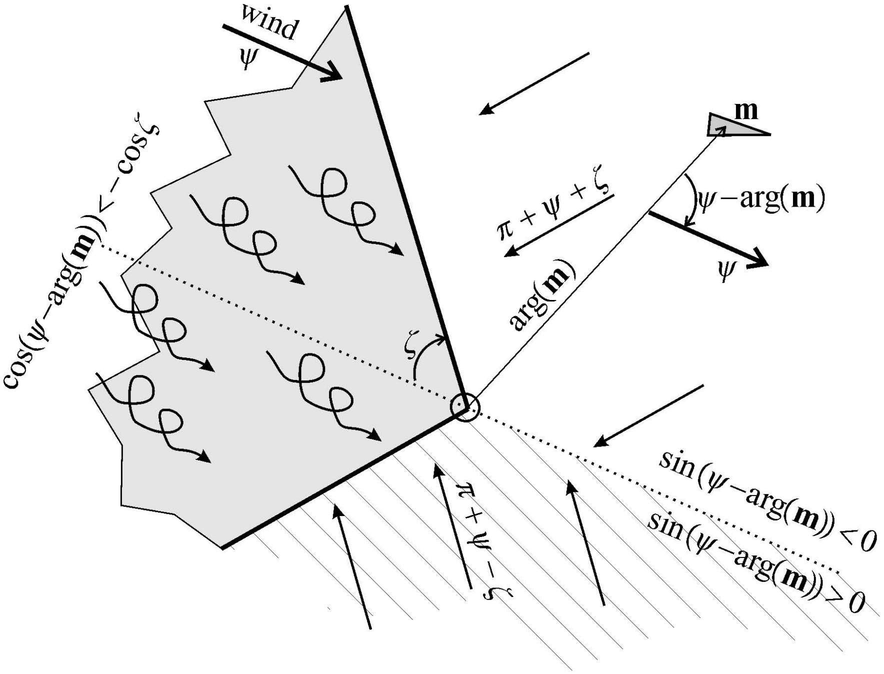
wind
ψ
m
$\psi-\arg(\mathbf{m})$
ψ
$\pi+\psi+\zeta$
$\arg(\mathbf{m})$
ζ
$\cos(\psi-\arg(\mathbf{m}))<-\cos\zeta$
$\pi+\psi-\zeta$
$\sin(\psi-\arg(\mathbf{m}))<0$
$\sin(\psi-\arg(\mathbf{m}))>0$

Fig. 4 Principle of the controller to maintain the robot around the origin still trying to spin as a mill in order to charge the batteries

```
Function in: $\mathbf{m}, \theta, \psi$;  out: $u_1, u_2$;  inout: $q, t_0$
if ($q = 1$ and $\psi \sim \theta$) then $q = 2$;
if ($q = 2$ and  $\psi \sim \theta + \pi$) then
        if $\cos(\psi{-}\arg(\mathbf{m})) > -\cos\zeta$
                $t_0 = t$
                if $(\sin(\psi - \arg(\mathbf{m})) > 0)$ then $q = 3$; else $q = 4$
        else $q = 1$
if ($q \in \{3, 4\}$ and $t - t_0 > 30$) then $q = 1$;
if $q = 1$ then $\bar{\theta} = \psi$
if $q = \{2, 4\}$ then $\bar{\theta} = \pi + \psi + \zeta$
if $q = 3$ then $\bar{\theta} = \pi + \psi - \zeta$
if ( $\cos\left(\theta - \bar{\theta}\right) \leq 0$ or $q \leq 2$)
        then $u_1 = \frac{\pi}{4}.\text{sign}\left(\sin\left(\theta - \bar{\theta}\right)\right)$
        else $u_1 = \frac{\pi}{4}.\sin\left(\theta - \bar{\theta}\right)$
$\bar{\ell} = \frac{\pi}{2}.\left(\frac{\cos(\psi - \bar{\theta}) + 1}{2}\right)$.
if $\bar{\ell} > \ell$ then $u_2 = 1$ else $u_2 = 0$.
```

Steps 1 to 7 correspond to the discrete event part of our hybrid controller. It is illustrated by the Petri net of Figure 5. The gray places represent actual states and white places represent fake states (the token leaves a fake place as soon as it enters in it). Bold arrows have a higher priority and are necessary to make the Petri net deterministic. Let us now describe the different discrete states for q.

- *Traction* ($q = 1$). The controller opens the sail and maneuvers to go downwind (see Step 8) as fast as possible (see Steps 11, 12). The controller escapes the state $q = 1$ at Step 1 as soon as $\psi \sim \theta$ (i.e., $\psi = \theta \pm 2k\pi$). When $q = 1$, the mainsheet pulls the generator and energy is produced.
- *Rewind* ($q = 2$). The controller makes the boat rotating toward the wind, in order to close the sail. When the robot is upwind ($\psi \sim \theta + \pi$), then the rewind mode terminates (see Step 2). If the robot is inside the mill cone, the controller goes to the state $q = 1$ at Step 6. Otherwise, depending on which cone the robot is, the controller chooses the states $q = 3$ or $q = 4$ at Step 5.
- *Positioning* ($q \in \{3, 4\}$). The controller chooses a close-hauled heading for 30 second, in order to bring closer to the mill cone.

Steps 8 to 10 provide the desired heading $\bar{\theta}$ to follow, depending of the value of q. When $q = 1$, the controller asks to go downwind (Step 8), Otherwise, it ask to go to a close hauled mode (Steps 9,10). Steps 11,12,13 tune the rudder (see [14] for more explanations). If the boat goes toward the wrong direction ($\cos\left(\theta - \bar{\theta}\right) \leq 0$) or if $q \in \{1, 2\}$, the rudder at its maximum, i.e., $\pm\frac{\pi}{4}$. otherwise, a proportional control is proposed (Step 13). For tuning of the sail, we propose to take a Cardioid model [13] at Step 14 to compute the right angle for the sail. At Step 15, the controller suggests to open the sail ($u_2 = 1$) in order to reach the desired length $\bar{\ell}$, by opening the blocker. This will mainly happen when $q = 1$.

4 Test-Case

In order to illustrate the principle of the controller, we now propose a simulation of the controlled sailboat robot. For the parameters, we have chosen

$$p_1 = 0.05,\, p_2 = 0.2 \text{ kg} \cdot \text{s}^{-1},\, p_3 = 6000 \text{ Kg} \cdot \text{m},$$
$$p_4 = 1000 \text{ kg} \cdot \text{s}^{-1},\ p_5 = 2000 \text{ kg} \cdot \text{s}^{-1},$$
$$p_6 = 1\text{m},\ p_7 = 1\text{m},\ p_8 = 2\text{m},\ p_9 = 300 \text{ kg},\, p_{10} = 10000 \text{ Kg} \cdot \text{m}^2.$$

Except for p_6, these values correspond approximately to the coefficients of the sailboat robot *VAIMOS* [14]. The value for p_6 is almost zero for *VAIMOS*, due to the balestron rig (or balanced rig). For our application, it is important

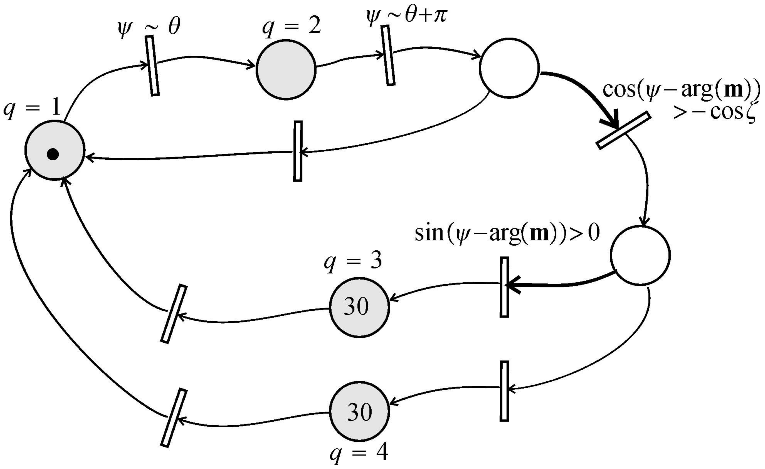
$\psi \sim \theta$
$q = 2$
$\psi \sim \theta+\pi$
$q = 1$
$\cos(\psi - \arg(\mathbf{m})) > -\cos\zeta$
$\sin(\psi - \arg(\mathbf{m})) > 0$
$q = 3$
30
30
$q = 4$

Fig. 5 Petri net associated with our windmill sailboat

to have an important p_6 for the energy production. For the speed a of the wind and its direction ψ, we took.

$$a = 4 \text{ m} \cdot \text{s}^{-1}, \quad \psi = \pi.$$

For the parameter of the controller, we have chosen $\zeta = 1$ rad. The resulting trajectory is illustrated by Figure 6 where the robot is initialized at $\mathbf{m} = (-400, 200)$ (small black disk). The trajectory corresponds to 30 minutes and the average of the collected power is around 93W. On the picture, we clearly see that on the mill cone, the boat rotates and move downwind. As soon as it goes outside the cone, it comes back to the mill cone choosing the right close hauled heading. The executable program and the C++ source code of the simulator that has generated Figure 6 can be found at

`http://www.ensta-bretagne.fr/jaulin/mill.html`

Remark. Betz's law [1] claims that the maximum power that can be extracted from the wind, independent of the design of a wind turbine, is given by

$$P_{\text{Betz}} = \frac{8}{27} \rho S a^3,$$

where S is the surface of the turbine, ρ is the fluid density and a is the speed of the wind. From this formula, we can deduce that to get the same power than that collected by the batteries in our test-case, a wind turbine with a diameter of 2.4 meters would be needed. Of course, such a wind turbine would change significantly the dynamic performances of the sailboat.

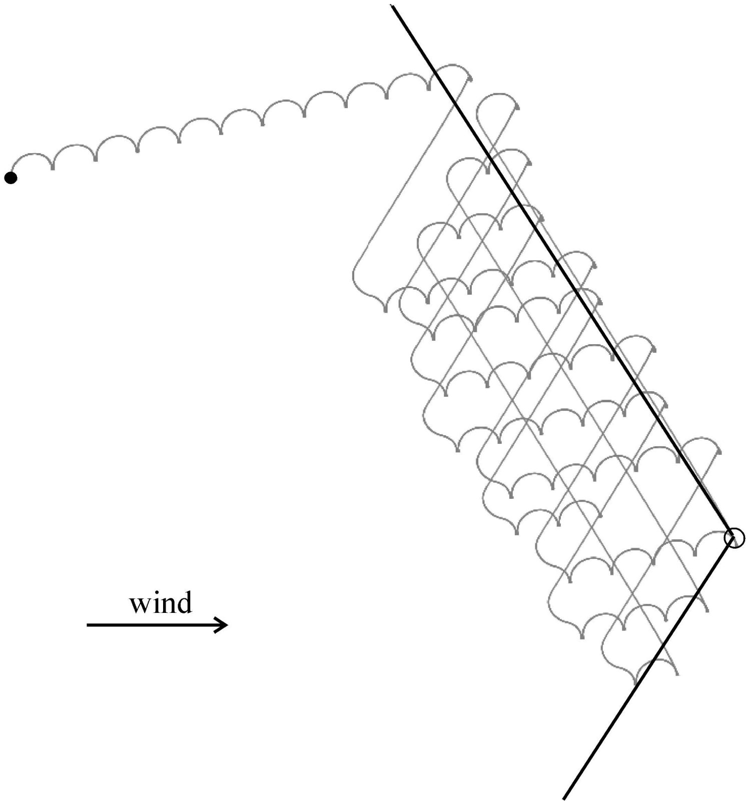

Fig. 6 Trajectory of the sailboat robot which tries to remain inside the circle and also to collect energy from the wind

5 Conclusion

In this paper we have presented a new concept that allows a sailboat to take advantage of a station keeping mode in order to charge its batteries. The basic idea is to transform the sailing boat into a windmill using a hybrid controller. When the wind inflates the sail, the mainsheet pull a generator that produces energy for the batteries. A test-case has shown that a mean power of 93W could be collected for a wind speed equal to 4 m·s^{-1} . All computations made by the controller can be performed using any cheap and low-powered microcontroller.

Acknowledgements. The robot *VAIMOS* is the result of a collaboration between LPO (Laboratoire de Physique des Océans), RDT (Recherches et Développements Technologiques) of Ifremer (Institut Français de Recherche pour l'Exploitation de la

Mer) and ENSTA Bretagne (Ecole Nationale Supérieure de Techniques Avancées). The authors render thanks to all people involved in the project: Y. Auffret, S. Barbot, L. Dussud, B. Forest, E. Menut, S. Prigent, L. Quemeneur, P. Rousseaux (RDT, Ifremer); F. Gaillard, T. Gorgues, O. Ménage, J. Moranges, T. Terre (LPO) and B. Clément, Y. Gallou, O. Reynet, J. Sliwka and B. Zerr (ENSTA Bretagne).

References

1. Betz, A.: Introduction to the Theory of Flow Machines. Pergamon Press, Oxford (1966)
2. Brière, Y.: The first microtransat challenge. In: ENSICA (2006), `http://web.ensica.fr/microtransat`
3. Cruz, N., Alves, J.: Ocean sampling and surveillance using autonomous sailboats. In: International Robotic Sailing Conference, Austria (2008)
4. Elkaim, G., Kelbley, R.: Station Keeping and Segmented Trajectory Control of a Wind-Propelled Autonomous Catamaran. In: Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, USA (2006)
5. Elkaim, G., Lee Boyce Jr., C.: An Energy Scavenging Autonomous Surface Vehicle for Littoral Surveillance. In: ION Global Navigation Satellite Systems Conference (2008)
6. Erckens, H., Büsser, G., Pradalier, C., Siegwart, R.: Navigation Strategy and Trajectory Following Controller for an Autonomous Sailing Vessel. IEEE RAM 17, 47–54 (2010)
7. Fossen, T.: Guidance and Control of Ocean Vehicles. Wiley, New York (1995)
8. Gale, T., Walls, J.: Development of a sailing dinghy simulator. Simulation 74(3), 167–179 (2000)
9. Guillou, G.: Architecture multi-agents pour le pilotage automatique des voiliers de compétition et extensions algébriques des réseaux de petri. PhD dissertation, Université de Bretagne, Brest, France (2011)
10. Jaulin, L.: Modélisation et commande d'un bateau à voile. In: CIFA2004, (Conférence Internationale Francophone d'Automatique), CDROM. Douz (Tunisie) (2004)
11. Jaulin, L.: Représentation d'état pour la modélisation et la commande des systèmes (Coll. Automatique de base). Hermès, London (2005)
12. Jaulin, L., Le Bars, F.: A simple controller for line following of sailboats. In: 5th International Robotic Sailing Conference, pp. 107–119. Springer, Cardiff (2012)
13. Jaulin, L., Le Bars, F.: An interval approach for stability analysis; Application to sailboat robotics. IEEE Transaction on Robotics 27(5) (2012)
14. Jaulin, L., Le Bars, F., Clément, B., Gallou, Y., Ménage, O., Reynet, O., Sliwka, J., Zerr, B.: Suivi de route pour un robot voilier. In: CIFA 2012, Grenoble, France (2012)
15. Le Bars, F., Jaulin, L.: An experimental validation of a robust controller with the VAIMOS autonomous sailboat. In: 5th International Robotic Sailing Conference, pp. 74–84. Springer, Cardiff (2012)
16. Miller, P.H., Hamlet, M., Rossman, J.: Continuous improvements to USNA sailbots for inshore racing. In: 5th International Robotic Sailing Conference, pp. 49–60. Springer, Cardiff (2012)

17. Neumann, T., Schlaefer, A.: Feasibility of basic visual navigation for small sailboats. In: 5th International Robotic Sailing Conference, pp. 13–22. Springer, Cardiff (2012)
18. Petres, C., Ramirez, M.R., Plumet, F.: Reactive path planning for autonomous sailboat. In: IEEE International Conference on Advanced Robotics, pp. 1–6 (2011)
19. Rynne, P., von Ellenrieder, K.: Unmanned autonomous sailing: Current status and future role in sustained ocean observations. MTS Journal 43(1), 21–30 (2009)
20. Sauze, C., Neal, M.: An autonomous sailing robot for ocean observation. In: Proceedings of TAROS 2006, Guildford, UK, pp. 190–197 (2006)
21. Stelzer, R., Dalmau, D.E.: A study on potential energy savings by the use of a balanced rig on a robotic sailing boat. In: 5th International Robotic Sailing Conference, pp. 89–93. Springer, Cardiff (2012)
22. Xiao, K., Sliwka, J., Jaulin, L.: A wind-independent control strategy for autonomous sailboats based on voronoi diagram. In: CLAWAR 2011 (best paper award), Paris (2011)
23. Xiao, L., Jouffroy, J.: Modeling and nonlinear heading control of sailing yachts. IEEE Journal of Oceanic Engineering (2013)