Tight slalom control for sailboat robots

Maël Le Gallic, Joris Tillet, Luc Jaulin, Fabrice Le Bars

Lab-STICC, ENSTA Bretagne, Brest, France

Abstract. Existing controllers for sailboat robots are usually developed for speed performances and for long straight lines. In this context, the accuracy is not the main concern. In this paper, we consider the tight slalom problem which requires accuracy. We propose a feedback-linearization based method combined with a vector field approach to control the sailboat. Some simulations show that the robot is able to perform the slalom without missing any gate.

1 Introduction

We consider a mobile robot described by the state equations [5]

$$\left\{ \begin{array}{lll} \dot{\mathbf{x}} & = & \mathbf{f}(\mathbf{x},\mathbf{u}) \\ \mathbf{p} & = & \mathbf{g}(\mathbf{x}) \end{array} \right. \tag{1}$$

with an input vector $\mathbf{u} = (u_1, \dots, u_m)$, a state vector $\mathbf{x} = (x_1, \dots, x_n)$ and a pose vector $\mathbf{p} = (p_1, \dots, p_{m+1})$ with $n \geq m+1$. The goal of this paper is to show we can follow a chosen vector field in the $\mathbf{p}$ space [8][12][13], using a feedback-linearization based method. It means that we can control $m+1$ state variables and not only m of them, as given by the theory [4]. This is due to the fact that we perform a path following instead of a trajectory tracking where the time is involved. In practice, the vector $\mathbf{p}$ corresponds to the position of the center of the robot and may be of dimension 2 (if $m = 1$) or 3 (if $m = 2$). This is consistent with the fact that we need one actuator to control the direction of a 2D vehicle such as a car or a boat and two actuators for a 3D vehicle such as a plane.

The approach we propose is to find a controller so that the vector $\dot{\mathbf{p}}$ be collinear (instead of equal) to the required field. This is illustrated in this paper in the case where the mobile robot is a sailboat [10][2][9]. The input $\mathbf{u}$ is scalar (*i.e.*, $m = 1$) and corresponds to the rudder. Moreover, we will show that this approach is particularly adapted to sailboats where the speed is hardly controllable [11][14].

2 Method

In order to facilitate the understanding of our approach, we will deal with a Dubins car, which is much simpler than a sailboat. The extension to other type of mobile robots is straightforward.

2.1 Line following for a Dubins car

To introduce our approach, we consider a robot (here a Dubins car) moving on a plane and described by the following state equations:

$$\left\{\begin{array}{lcl}\dot{x}_1 & = & \cos x_3 \\ \dot{x}_2 & = & \sin x_3 \\ \dot{x}_3 & = & u\end{array}\right. \tag{2}$$

where x_3 is the heading of the robot and $\mathbf{p} = (x_1, x_2)$ are the coordinates of its center. The state vector is given by $\mathbf{x} = (x_1, x_2, x_3)$.

Let us choose as the control output the variable

$$y = x_3 + \text{atan} x_2. \tag{3}$$

and let us find a classical feedback linearization based controller [6] such that the output y (which can be interpreted as an error) converges to 0. In such a case, we will have $x_3 + \text{atan} x_2 = 0$ and the robot will perform a line following. Differentiating (3) we have

$$\dot{y} = \dot{x}_3 + \frac{\dot{x}_2}{1 + x_2^2} = u + \frac{\sin x_3}{1 + x_2^2}. \tag{4}$$

Since u occurs in (4), the relative degree of the system is 1. We may thus choose a first order equation for the error y, such as

$$\dot{y} + y = 0, \tag{5}$$

We then choose u to have this error equation satisfied. From (4) and (5), we get:

$$\begin{array}{rcl}u & = & -y - \frac{\sin x_3}{1+x_2^2} \\ & = & -x_3 - \text{atan} x_2 - \frac{\sin x_3}{1+x_2^2}\end{array} \tag{6}$$

Note that we do not have any singularity. As illustrated by the simulation depicted on Figure 1, the associated vector field makes the car attracted by the line $x_2 = 0$.

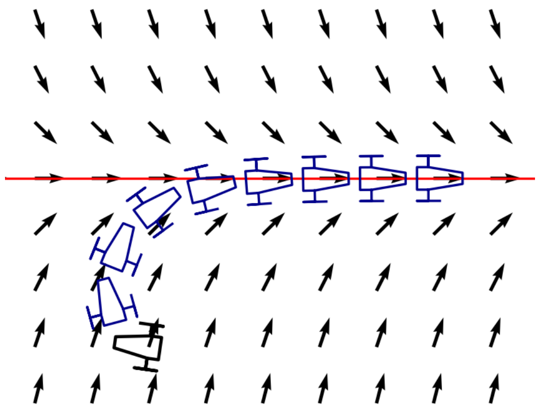

Figure 1: Precise line following

Remark. For more robustness with respect to small uncertainties, a sliding mode effect could be added. It suffices to take for required error

$$y = x_3 + \operatorname{atan}\left(x_2 + \alpha \cdot \operatorname{sign}\left(x_2\right)\right),$$

where α is a small positive coefficient, *e.g.*, $\alpha = 0.1$. In such a case, the robot will go to the line in a finite time (and not asymptotically, as previously). Moreover, it will remains exactly on the line even if some small uncertainties occur.

2.2 Generalization

We want our robot to follow the field $\boldsymbol{\psi}(\mathbf{p})$, more precisely, we want that $\boldsymbol{\psi}(\mathbf{p})$ and $\dot{\mathbf{p}}$ point toward the same direction. This condition can be translated into the form $\boldsymbol{\varphi}\left(\boldsymbol{\psi}(\mathbf{p}), \dot{\mathbf{p}}\right) = 0$, where $\boldsymbol{\varphi}$ is a *collinearity function* which satisfies

$$\boldsymbol{\varphi}\left(\mathbf{r}, \mathbf{s}\right) = \mathbf{0} \Leftrightarrow \exists \lambda > 0,\ \lambda \mathbf{r} = \mathbf{s}. \tag{7}$$

Typically, this function corresponds to one angle (the heading) if $m = 1$ and two angles (heading, elevation) for $m = 2$. Note that the function $\boldsymbol{\varphi}$ cannot be expressed with a determinant since $\mathbf{r}, \mathbf{s}$ should not point toward opposite directions. We define the output

$$\mathbf{y} = \boldsymbol{\varphi}\left(\boldsymbol{\psi}(\mathbf{p}), \dot{\mathbf{p}}\right) = \boldsymbol{\varphi}\left(\boldsymbol{\psi}(\mathbf{g}\left(\mathbf{x}\right)), \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\mathbf{x}) \cdot \mathbf{f}\left(\mathbf{x}, \mathbf{u}\right)\right). \tag{8}$$

Since $\mathbf{y} \in \mathbb{R}^m$, we can apply a feedback linearization method and we get $\mathbf{y} \to \mathbf{0}$. This means that the robot will follows the required field. Note that we have no control on the speed, which is not our main concern in this paper.

2D case. Consider for instance the case where $m = 1$. We have

$$\boldsymbol{\psi}(\mathbf{p}) = \begin{pmatrix} \psi_1(\mathbf{p}) \\ \psi_2(\mathbf{p}) \end{pmatrix}. \tag{9}$$

We take as an output y, the angle between the actual heading vector $\dot{\mathbf{p}} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}, \mathbf{u})$ and the desired heading vector given by $\boldsymbol{\psi}(\mathbf{p})$. Denote by $\theta(\mathbf{x})$ the argument of the vector $\dot{\mathbf{p}}$. We have

$$\begin{array}{rcl} y \overset{(8)}{=} & = & \text{angle}\left(\boldsymbol{\psi}(\mathbf{g}(\mathbf{x})), \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}, \mathbf{u})\right) \\ & = & \text{angle}\left(\boldsymbol{\psi}(\mathbf{g}(\mathbf{x})), \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}\right) \\ & = & \text{sawtooth}(\theta - \text{atan2}(\underbrace{\psi_2(\mathbf{g}(\mathbf{x}))}_{b}, \underbrace{\psi_1(\mathbf{g}(\mathbf{x}))}_{a})) \end{array} \tag{10}$$

The *sawtooth* function is given by:

$$\text{sawtooth}(\widetilde{\theta}) = 2\text{atan}\left(\tan\frac{\widetilde{\theta}}{2}\right) = \text{mod}(\widetilde{\theta} + \pi, 2\pi) - \pi \tag{11}$$

As illustrated by Figure 2, the function corresponds to an error in heading. The interest in taking an error $\widetilde{\theta}$ filtered by the *sawtooth* function is to avoid the problem of the $2k\pi$ modulus: we would like a $2k\pi$ to be considered non-zero.

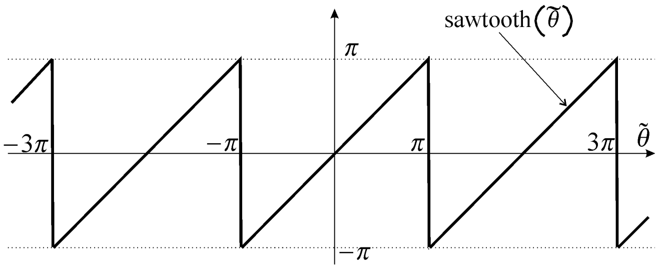

Figure 2: *Sawtooth* function used to avoid the jumps in the heading control

We have

$$\begin{array}{rcl} \dot{y} & = & \dot{\theta} - (\underbrace{-\frac{b}{a^2 + b^2}}_{\frac{\partial \text{atan2}(b,a)}{\partial a}} \cdot \dot{a} + \underbrace{\frac{a}{a^2 + b^2}}_{\frac{\partial \text{atan2}(b,a)}{\partial b}} \cdot \dot{b}) \\ & = & u + \frac{b \cdot \dot{a} - a \cdot \dot{b}}{a^2 + b^2}, \end{array} \tag{12}$$

if we assume that the input u corresponds to the desired angular velocity. We propose a feedback linearization based control based on the required equation

$\dot{y} = -y$. We have

$$\begin{array}{rcl} u & \overset{(12)}{=} & \dot{y} - \frac{\left(b\cdot\dot{a}-a\cdot\dot{b}\right)}{a^2+b^2} \\ & = & -y - \frac{\left(b\cdot\dot{a}-a\cdot\dot{b}\right)}{a^2+b^2} \qquad\qquad (\text{since } \dot{y}=-y) \\ & \overset{(10)}{=} & -\left(\text{sawtooth}(\theta - \text{atan2}(b,a)\right) - \frac{\left(b\cdot\dot{a}-a\cdot\dot{b}\right)}{a^2+b^2}. \end{array} \tag{13}$$

We thus have the guarantee that after some time, the error angle y is 0 and that we follow exactly the vector field.

2.3 Dubins car following the Van der Pol cycle

We would like our Dubins car to follow a path corresponding to the limit cycle of the Van der Pol equation:

$$\boldsymbol{\psi}(\mathbf{p}) = \begin{pmatrix} p_2 \\ -\left(0.01\ p_1^2 - 1\right) p_2 - p_1 \end{pmatrix}. \tag{14}$$

Take $\mathbf{g}(\mathbf{x}) = (x_1, x_2)^{\mathrm{T}}$ which means that we want to build the paths in the (x_1, x_2)-space. We have

$$\boldsymbol{\psi}(\mathbf{g}(\mathbf{x})) = \begin{pmatrix} x_2 \\ -\left(0.01\ x_1^2 - 1\right) x_2 - x_1 \end{pmatrix} \tag{15}$$

and

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \cos x_3 \\ \sin x_3 \\ u \end{pmatrix} \tag{16}$$

Thus

$$\begin{array}{rcl} a & = & x_2 \\ b & = & -(0.01\ x_1^2 - 1)x_2 - x_1 \\ \theta & = & x_3 \end{array} \tag{17}$$

and

$$\begin{array}{rcl} \dot{a} & = & \sin x_3 \\ \dot{b} & = & -\left(0.01 \cdot 2x_1\dot{x}_1\right) x_2 - \left(0.01\ x_1^2 - 1\right)\dot{x}_2 - \dot{x}_1 \\ & = & -0.02 \cdot x_1 x_2 \cos x_3 - \left(0.01\ x_1^2 - 1\right)\sin x_3 - \cos x_3 \end{array} \tag{18}$$

From (13), we get that final controller is

$$\begin{array}{rcl} u & = & -\text{sawtooth}\left(x_3 - \text{atan2}\left(-\left(\frac{x_1^2}{100} - 1\right) x_2 - x_1, x_2\right)\right) \\ & & + \frac{\left(\left(\frac{x_1^2}{100}-1\right)x_2+x_1\right)\cdot\sin x_3 + x_2\cdot\left(\frac{x_1 x_2 \cos x_3}{50} + \left(\frac{x_1^2}{100}-1\right)\sin x_3 + \cos x_3\right)}{x_2^2+\left(\left(\frac{x_1^2}{100}-1\right)x_2+x_1\right)^2} \end{array} \tag{19}$$

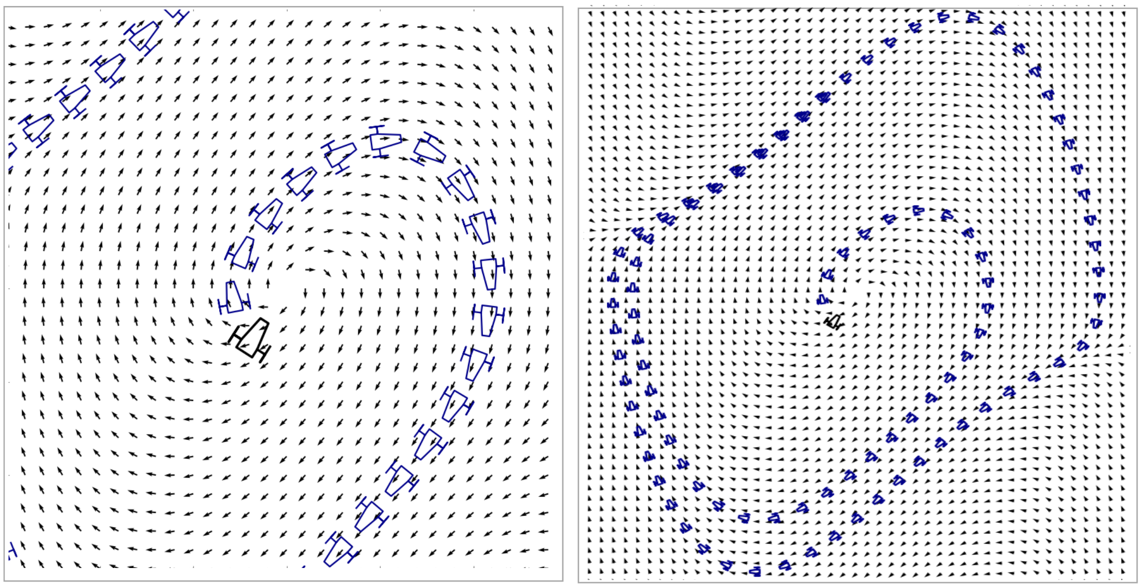
Figure 3: Dubins describing accurately the Van der Pol cycle

The behavior of the control law is illustrated by Figure 3. The car is very close to the true limit cycle, which is not the case if we consider a classical linear controller. Indeed, the controller anticipates the fact that the required trajectory have to take into account the curvature of the vector field.

3 Application to the slalom problem

We consider the following model which corresponds to a simplified version of the sailboat model given in [7]. The state equations are

$$\left\{\begin{array}{lll} \dot{x}_1 & = & v\cos\theta \\ \dot{x}_2 & = & v\sin\theta \\ \dot{\theta} & = & -\rho_2 v\sin 2u_1 \\ \dot{v} & = & \rho_3\left\|\mathbf{w}_{\text{ap}}\right\|\sin\left(\delta_s-\psi_{\text{ap}}\right)\sin\delta_s-\rho_1 v^2 \\ \sigma & = & \cos\psi_{\text{ap}}+\cos u_2 \\ \delta_s & = & \left\{\begin{array}{cc} \pi+\psi_{\text{ap}} & \text{if } \sigma\leq 0 \\ -\text{sign}\left(\sin\psi_{\text{ap}}\right)\cdot u_2 & \text{otherwise} \end{array}\right. \\ \mathbf{w}_{\text{ap}} & = & \left(\begin{array}{c} -a\ \sin\left(\theta\right)-v \\ -a\ \cos\left(\theta\right) \end{array}\right) \\ \psi_{\text{ap}} & = & \text{angle }\mathbf{w}_{\text{ap}} \end{array}\right. \tag{20}$$

where $\rho_1 = 0.003, \rho_2 = 0.2, \rho_3 = 3$. In this equation u_1, u_2 correspond to the tuning of the rudder and the sail, respectively. We would like our robot to follow a path which makes a tight slalom through doors that have to be passed. We assume that we have a Cartesian equation for our path. For instance, we consider that the path is described by

$$e\left(\mathbf{p}\right)=10\sin\left(\frac{p_1}{10}\right)-p_2=0 \tag{21}$$

where $e\left(\mathbf{p}\right)$ corresponds to an error. This path corresponds to a path that should be possible for a normal sailboat robot for crosswind conditions. We take a vector field which corresponds to a pole placement strategy. For instance, we want the error satisfies $\dot{e}=-0.1\,e$, so that it will converge to zero in about 10 sec. Thus

$$\underbrace{\cos\left(\frac{p_1}{10}\right)\dot{p}_1-\dot{p}_2}_{\dot{e}(\mathbf{p})}=-\frac{1}{10}\underbrace{\left(10\sin\left(\frac{p_1}{10}\right)-p_2\right)}_{e(\mathbf{p})} \tag{22}$$

We take $\dot{p}_1=1$, to go to the right. As a consequence, we get the following field:

$$\boldsymbol{\psi}(\mathbf{p})=\begin{pmatrix}\dot{p}_1\\ \dot{p}_2\end{pmatrix}=\begin{pmatrix}1\\ \cos\left(\frac{p_1}{10}\right)+\frac{1}{10}\left(10\sin\left(\frac{p_1}{10}\right)-p_2\right)\end{pmatrix} \tag{23}$$

which is attracted by the curve $p_2=10\sin\left(\frac{p_1}{10}\right)$.

We have

$$\boldsymbol{\psi}(\mathbf{g}\left(\mathbf{x}\right))=\begin{pmatrix}1\\ \cos\left(\frac{x_1}{10}\right)+\frac{1}{10}\left(10\sin\left(\frac{x_1}{10}\right)-x_2\right)\end{pmatrix} \tag{24}$$

and

$$\frac{\partial\mathbf{g}}{\partial\mathbf{x}}(\mathbf{x})\cdot\mathbf{f}\left(\mathbf{x},\mathbf{u}\right)=\begin{pmatrix}\cos x_3\\ \sin x_3\end{pmatrix}. \tag{25}$$

Thus

$$\begin{array}{rcl} a & = & 1\\ b & = & \cos\left(\frac{x_1}{10}\right)+\sin\left(\frac{x_1}{10}\right)-\frac{1}{10}x_2\end{array} \tag{26}$$

and

$$\begin{array}{rcl} \dot{a} & = & 0\\ \dot{b} & = & -\dot{x}_1\frac{1}{10}\sin\left(\frac{x_1}{10}\right)+\dot{x}_1\frac{1}{10}\cos\left(\frac{x_1}{10}\right)-\frac{1}{10}\dot{x}_2\\ & =\frac{1}{10} & \cos x_3\cdot\left(\cos\left(\frac{x_1}{10}\right)-\sin\left(\frac{x_1}{10}\right)\right)-\frac{1}{10}\sin x_3.\end{array} \tag{27}$$

From (13), we get that the desired angular velocity should be

$$\hat{\omega}\;=\;-\left(\text{sawtooth}(\theta-\text{atan2}(b,a)\right)-\frac{\left(b\cdot\dot{a}-a\cdot\dot{b}\right)}{a^2+b^2}\;. \tag{28}$$

Now, since the true angular velocity is $\dot{\theta}=-\rho_2 v\sin 2u_1$, we take

$$u_1=-\frac{1}{2}\text{arcsin}\left(\tanh\left(\frac{\hat{\omega}}{\rho_2 v}\right)\right). \tag{29}$$

The saturation function *tanh* is needed since the rudder cannot respond to any required $\hat{\omega}$. Indeed, if our controller ask to turn too fast for the boat, $\frac{\hat{\omega}}{\rho_2 v}$ will be more than 1, and the rudder can only do its best. The behavior of our controller is illustrated by Figure 4, where the sailboat has to slalom tightly between doors. We can see that the trajectory follows exactly the sine path (magenta).

The Python source codes associated to the simulation can be found at:

https://www.ensta-bretagne.fr/jaulin/slalompy.zip

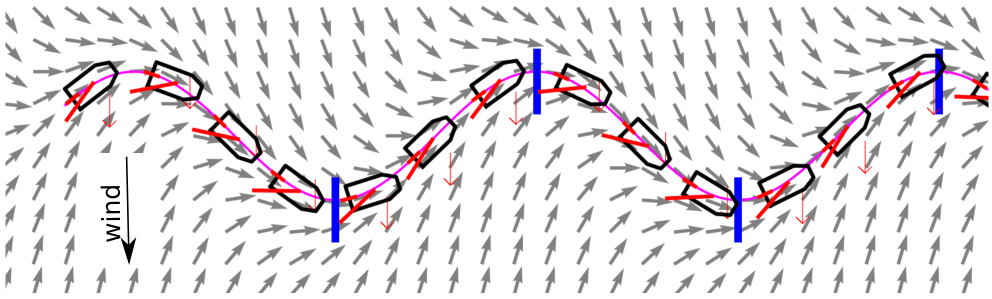

Figure 4: The sailboat robot slaloms through the blue doors

4 Conclusion

In this paper, we have proposed a new controller for sailboat robots which allows to take into account the curvature of the required field in order to anticipate as much as possible the required trajectory. To our knowledge, this is not considered by existing controllers [1] which are devoted to straight lines [3]. It has been shown that the required vector field could be followed exactly. This anticipation is crucial if we want to maneuver quickly and precisely as needed when we want to avoid an obstacle. This has been illustrated on a simulated test-case where a tight slalom is performed by a sailboat robot.

References

[1] F. Le Bars and L. Jaulin. An experimental validation of a robust controller with the VAIMOS autonomous sailboat. In *5th International Robotic Sailing Conference*, pages 74–84, Cardiff, Wales, England, 2012. Springer.

[2] N.A. Cruz and J.C. Alves. Ocean sampling and surveillance using autonomous sailboats. In *International Robotic Sailing Conference*, Austria, 2008.

[3] F. Le Bars F. PLumet, Y. Briere. Les voiliers robotisés. *Les Techniques de l'Ingénieur*, 2018.

[4] A. Isidori. *Nonlinear Control Systems: An Introduction, 3rd Ed.* Springer-Verlag, New-York, 1995.

[5] L. Jaulin. *Automation for Robotics*. ISTE editions, 2015.

[6] L. Jaulin. *Mobile Robotics*. ISTE editions, 2015.

[7] L. Jaulin and F. Le Bars. An Interval Approach for Stability Analysis; Application to Sailboat Robotics. *IEEE Transaction on Robotics*, 27(5), 2012.

[8] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

[9] H. Klinck, R. Stelzer, K. Jafarmadar, and D. K. Mellinger. AAS Endurance: An Autonomous Acoustic Sailboat for Marine Mammal Research. In *2th International Robotic Sailing Conference*, Matosinhos, Portugal, 2009.

[10] P. H. Miller, M. Hamlet, and J. Rossman. Continuous improvements to USNA sailbots for inshore racing. In *5th International Robotic Sailing Conference*, pages 49–60, Cardiff, Wales, England, 2012. Springer.

[11] T. Neumann and A. Schlaefer. Feasibility of basic visual navigation for small sailboats. In *5th International Robotic Sailing Conference*, pages 13–22, Cardiff, Wales, England, 2012. Springer.

[12] C. Petres, M. Romero Ramirez, and F. Plumet. Reactive Path Planning for Autonomous Sailboat. In *IEEE International Conference on Advanced Robotics*, pages 1–6, 2011.

[13] S. Schmitt, F. Le Bars, L. Jaulin, and T. Latzel. Obstacle Avoidance for an Autonomous Marine Robot - A Vector Field Approach. In *7th International Robotic Sailing Conference*, Irland, 2014. Springer.

[14] R. Stelzer, T. Proll, and R. John. Fuzzy Logic Control System for Autonomous Sailboats. In *in Proceedings of IEEE International Conference on Fuzzy Systems*, London, UK, 2007.