View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Digitale Bibliothek Thüringen

PROCEEDINGS

55. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium

13 - 17 September 2010

Crossing Borders within the ABC

Automation,

Biomedical Engineering and

Computer Science

Faculty of
Computer Science and Automation

www.tu-ilmenau.de

TECHNISCHE UNIVERSITÄT
ILMENAU

Home / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=16739

Impressum
Published by

Publisher: Rector of the Ilmenau University of Technology
Univ.-Prof. Dr. rer. nat. habil. Dr. h. c. Prof. h. c. Peter Scharff

Editor: Marketing Department (Phone: +49 3677 69-2520)
Andrea Schneider (conferences@tu-ilmenau.de)

Faculty of Computer Science and Automation
(Phone: +49 3677 69-2860)
Univ.-Prof. Dr.-Ing. habil. Jens Haueisen

Editorial Deadline: 20. August 2010

Implementation: Ilmenau University of Technology
Felix Böckelmann
Philipp Schmidt

USB-Flash-Version.

Publishing House: Verlag ISLE, Betriebsstätte des ISLE e.V.
Werner-von-Siemens-Str. 16
98693 Ilmenau

Production: CDA Datenträger Albrechts GmbH, 98529 Suhl/Albrechts

Order trough: Marketing Department (+49 3677 69-2520)
Andrea Schneider (conferences@tu-ilmenau.de)

ISBN: 978-3-938843-53-6 (USB-Flash Version)

Online-Version:

Publisher: Universitätsbibliothek Ilmenau
ilmedia
Postfach 10 05 65
98684 Ilmenau

© Ilmenau University of Technology (Thür.) 2010

The content of the USB-Flash and online-documents are copyright protected by law.
Der Inhalt des USB-Flash und die Online-Dokumente sind urheberrechtlich geschützt.

Home / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=16739

ON PATH GENERATION AND FEEDFORWARD CONTROL FOR A CLASS OF SURFACE SAILING VESSELS

Lin Xiao and Jerome Jouffroy

Mads Clausen Institute
University of Southern Denmark (SDU)
Alsion 2, DK-6400, Sønderborg, Denmark
e-mail: {xiao,jerome}@mci.sdu.dk

ABSTRACT

Sailing vessels with wind as their main means of propulsion possess a unique property that the paths they take depend on the wind direction, which, in the literature, has attracted less attention than normal vehicles propelled by propellers or thrusters. This paper considers the problem of motion planning and controllability for sailing vehicles representing the no-sailing zone effect in sailing. Following our previous work, we present an extended algorithm for automatic path generation with a prescribed initial heading for a simple model of sailing vehicles, together with a feedforward controller guiding these vessels along desired trajectories of bounded curvatures. Further, this method immediately adapts to varying wind conditions. Simulation results are hereby presented to illustrate the approach.

Index Terms— Sailing vessels, motion planning, path generation, on-line planning, feedforward control.

1. INTRODUCTION

With the current worldwide concern on environmental issues, many solutions have been proposed recently to survey, explore, or monitor different parts of the environment around us, such as ocean surface or layers of the atmosphere. Among them, a series of mobile platforms is currently being introduced to perform these survey and exploration missions. For cost and human safety reasons, these platforms are typically unmanned and autonomous, which are mostly propelled by usual means like propellers and thrusters. There are very few implemented to use the power of the wind, and even less studies were dedicated to automation of this kind of sailing vehicles, i.e. our well-known sailboats, ships equipped with a kite or landyachts (see e.g. [3][1] or [4]).

In a previous study [8], we saw that both controllability and motion planning issues could be addressed with a single perspective, which took the form of a Boundary Value Problem (BVP). As an extension, we propose in this paper a general approach to deal with path planning for any couple of points in the plane with prescribed initial headings, considering wind conditions at the same time, which refers to on-line motion planning. This method reacts to changes of the wind in real-time by recalculating the path and the control signals, see also [9] for replanning issues, but in which the algorithm is recalculating the heading periodically and works with the aid of a simplified polar diagram.

Following the introduction, a strategy for suitable path generation in order to reach a specific target point is presented. Starting from the kinematic sailing vehicle model from [4][5], together with its controllability property, to propose a path generation algorithm. The basic principle of on-line path planning is also discussed. Finally, the generation of sequences of control inputs necessary to feedforward control is briefly explained and the particularities of the proposed strategy are illustrated by simulations using a computer model of a surface sailing vehicle.

2. PATH PLANNING STRATEGY

2.1. A nonlinear dynamical model

Consider the following simple model, taken from [4][5], which represents the behaviour of a surface sailing vehicle.

$$\begin{cases} \dot{x}(t) = v(t)\cos\theta(t), \\ \dot{y}(t) = v(t)\sin\theta(t), \\ \dot{\theta}(t) = v(t)\tan\delta(t)/L, \\ m\dot{v}(t) + dv(t) = g(\theta(t))F(t), \end{cases} \quad (1)$$

where function $g(\theta(t))$ is such that

$$g(\theta(t)) = \begin{cases} 0 & \text{if } \underline{\theta} \leq \theta(t) \leq \overline{\theta} \\ 1 & \text{otherwise} \end{cases} . \quad (2)$$

$(x(t),\ y(t))$ are the Cartesian coordinates of a reference point on the longitudinal axis of the vehicle, and the heading $\theta(t)$ is the angle formed by such axis with a direction in the plane, while $v(t)$ is the vessel's longitudinal velocity. Coefficients m and d are strictly positive constants. Variable $\delta(t)$ is the control input representing the steering action coming from e.g. a rudder, and takes values on the interval $[-\overline{\delta}, \overline{\delta}]$. Similarly, $F(t)$ is assumed to be the other control input accounting for the propulsion of the vehicle, which is limited to lie within the set $[-\overline{F}, \overline{F}]$ (for simplicity, we assume this interval to be symmetric around the origin, which implies a braking action for the sailing vehicle. This is possible for some vehicles if the boom angle can be directly controlled or with the presence of brakes, as on a landyacht. However, our discussion is also valid for interval $[\underline{F}, \overline{F}]$ with minor changes). Parameter L accounts (together with $\delta(t)$) for the bounded curvature in the plane of the trajectories followed by system (1). From the way the function $g(\theta(t))$ is defined, it is clear that the control input $F(t)$ has no effect on system (1) while $\theta(t)$ belongs to $[\underline{\theta}, \overline{\theta}]$, representing the loss of propulsion when the vehicle is in the no-sailing zone.

System (1)-(2) can be directly put into a state-space form $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$, where the state vector is $\mathbf{x}(t) = (x(t), y(t), \theta(t), v(t)) \in \mathbb{R}^3 \times \mathbb{S}^1$, together with the control input $\mathbf{u}(t) = (\delta(t), F(t)) \in \{-\overline{\delta}, \overline{\delta}\} \times \{-\overline{F}, \overline{F}\}$.

Theoretically, model (1)-(2) is close in spirit to the work done on the nonholonomic car well-known in mobile robotics, that received considerable attention in this community (see e.g. [6]). Note, moreover, that on the practical side, and despite their relative simplicity, similar models are widely used for practical implementations, notably for guidance and collision avoidance (see [2]).

The following proposition (see [4][8]) relates to the controllability properties of system (1)-(2) by expressing the fact that the surface sailing vessels have to accumulate enough energy before entering the no-sailing zone, or use the inertia of the sytem to regain controllability and to cross the deadzone.

Proposition 1 *The state $\mathbf{x}(0) = (x(0), y(0), \theta(0) = \underline{\theta}, v(0) = v_0)$ can not be controlled to the state $\mathbf{x}(T) = (x(T), y(T), \theta(T) = \overline{\theta}, v(T))$ on the time interval [0, T] if $F = 0$ and the following holds:*

$$v_0 < \frac{dL}{m} \frac{\overline{\theta} - \underline{\theta}}{\tan \overline{\delta}} \triangleq v_{min}. \tag{3}$$

2.2. Feasible path generation

In the following, we assume any couple of points in the plane, the sailing vessel starts (with zero velocities) with set initial headings from one point to go to the destination. We also assume that final headings can be decided upon, and all angles are defined on the interval $[-\pi, \pi]$.

In Fig. 1, point p_1 indicates the starting point. If the initial heading, e.g. θ_{02}, lies in the interval $[\underline{\theta}, \overline{\theta}]$, the vehicle cannot move because it fails in capturing enough energy from the environment. Therefore, the sailing vessel should start with headings (e.g. θ_{01}) outside the no-go zone.

In some cases, the vehicle is able to go straight to reach the target. For this simple case, consider we are given $p_1(x_0, y_0)$ and $p_2(x_T, y_T)$, then the angle of the line-of-sight is $\theta_0 = atan2((y_T - y_0), (x_T - x_0))$. If θ_0 is equal to the initial heading θ_{01}, the trajectory is a straight line connecting the two established points in the plane. The length of the corresponding path is $s_T = (y_T - y_0)/\sin\theta_0$ or $s_T = |x_T - x_0|$ $(y_T == y_0)$.

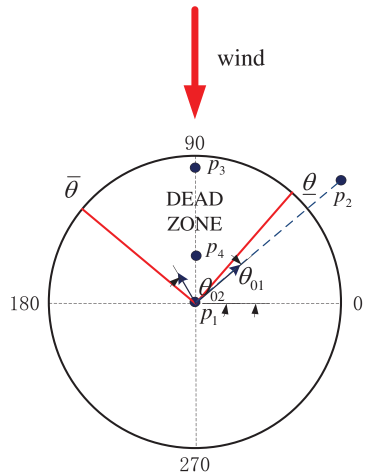

Fig. 1. Representation of starting and end points with initial angles.

Nevertheless, the end points (e.g. p_3 and p_4 in Fig. 1) usually can not be simply reached by only one straight line motion, the vehicle can also reach the destination but by zigzagging (tacking and wearing are the two main maneuvers to go upwind in sailing [4][5]) or by jibing (i.e. sailing before the wind and the vessel turns such that the wind direction changes from one side to the other). Conventionally, one would typically alternate straight lines and circles to build a path, the path generated by both tacking and wearing (or jibing) maneuvers could be constructed by doing so (see also [8]). If $\overline{\delta}$ or $-\overline{\delta}$ is applied during the turn, the radius of the circle followed by vehicle is $r = L/\tan\overline{\delta}$. Fig. 2 lays out the constructions of paths for tacking and wearing maneuvers.

However, whether we are tacking or wearing, both these maneuvers can be considered as the same geometric task, while crossing the no-sailing zone which is specific to tacking, hence more of a dynamic flavor. Therefore, for the geometric task, define a path in the plane by $x(s), y(s)$, where $s(t)$ is a path variable, according to (1) without considering forces, the path is:

$$\begin{cases} dx_i/ds(s) = \cos\theta_i(s), \\ dy_i/ds(s) = \sin\theta_i(s), \\ d\theta_i/ds(s) = \tan\delta_i(s)/L, \end{cases} \tag{4}$$

with $i = 1, 2, 3, \ldots$ representing the ith segment on the path. Since $p_1(x_0, y_0, \theta_0)$ and $p_3(x_T, y_T)$ (or $p_4(x_T,$

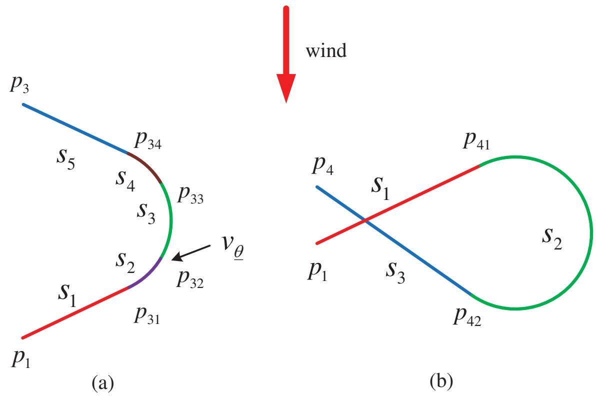

Fig. 2. Desired path for (a) tacking and (b) wearing.

y_T)) are known, together with the final headings θ_T which are supposed to be decided upon, solving for the path can be seen as a two-point BVP.

A standard form required for several BVP codes (e.g. *bvp4c* in MatLab) is (see also [7]) $\dot{y} = f(x, y(x), p), a \le x \le b$, subjects to boundary conditions $\mathcal{C}(y(a), y(b), p) = 0$, where p is a vector of unknown parameters. But (1) is not in the standard form because the independent variable t belongs to $[0, T]$ and T is also an unknown. However, If we change t to $\tau = \frac{t}{T}$ such that the problem is now posed on the fixed interval $\tau \in [0, 1]$, the new BVP form becomes $dy/d\tau(\tau) = Tf(\tau, y(\tau), T), 0 \le \tau \le 1$ with constraints $\mathcal{C}_\tau(y(0), y(1), T) = 0$. Take tacking trajectory in Fig. 2(a) for example, formulate ODEs with a constraint function:

$$
\frac{d\mathbf{x}}{d\tau}(\tau) = \begin{bmatrix} T_1 \cos\theta_1(\tau) \\ T_1 \sin\theta_1(\tau) \\ 0 \\ T_2 \cos\theta_2(\tau) \\ T_2 \sin\theta_2(\tau) \\ T_2 \tan\bar{\delta}/L \\ T_3 \cos\theta_3(\tau) \\ T_3 \sin\theta_3(\tau) \\ T_3 \tan\bar{\delta}/L \\ T_4 \cos\theta_4(\tau) \\ T_4 \sin\theta_4(\tau) \\ T_4 \tan\bar{\delta}/L \\ T_5 \cos\theta_5(\tau) \\ T_5 \sin\theta_5(\tau) \\ 0 \end{bmatrix}, \mathcal{C}_\tau = \begin{bmatrix} x_1(0) - x_0 \\ x_1(1) - x_2(0) \\ y_1(0) - y_0 \\ y_1(1) - y_2(0) \\ \theta_1(0) - \theta_0 \\ x_2(1) - x_3(0) \\ y_2(1) - y_3(0) \\ \theta_2(0) - \theta_0 \\ \theta_2(1) - \underline{\theta} \\ x_3(1) - x_4(0) \\ y_3(1) - y_4(0) \\ \theta_3(0) - \underline{\theta} \\ \theta_3(1) - \bar{\theta} \\ x_4(1) - x_5(0) \\ y_4(1) - y_5(0) \\ \theta_4(0) - \bar{\theta} \\ \theta_4(1) - \theta_T \\ x_5(1) - x_T \\ y_5(1) - y_T \\ \theta_5(0) - \theta_T \end{bmatrix} \tag{5}
$$

Consequently, the length for each segment (i.e. s_i) on the path can be solved automatically and numerically by a BVP solver.

Table 1. Path generation algorithm.

Name : PATH GENERATION ALGORITHM
Goal : To build a feasible continuous path between two prescribed points in the plane with a given initial heading belonging to $[-\pi, \pi]$.

1: enter boundary conditions (x_0, y_0, θ_0), (x_T, y_T)
2: **if** $\underline{\theta} \le \theta_0 \le \bar{\theta}$
3: warning('Can not start with $\theta_0 \in [\underline{\theta}, \bar{\theta}]$')
4: **elseif** $\theta_0 == \theta_{01}$
5: go straight
6: **elseif** $y_T \ge y_0$
7: **if** $0 \le \theta_0 \le \pi$
8: formulate tacking or wearing trajectory according to the criterions pictured in Fig. 3
9: **else** (i.e. $\theta_0 \in (-\pi, 0)$)
10: change the initial configuration such that $\theta_0' \in [0, \pi]$ (as pictured in Fig. 4 (a)) and repeat step 8
11: **end**
12: **else** (i.e. $y_T < y_0$)
13: **if** $-\pi < \theta_0 < -\pi/2$ or $-\pi/2 < \theta_0 < 0$
14: formulate jibing trajectories or tack if necessary (see Fig. 5)
15: **else**
16: change the initial configuration such that $\theta_0' \in (-\pi, -\pi/2) \cup (-\pi/2, 0)$ (as pictured in Fig. 4(b)(c)), then going downwind; or $\theta_0' \in [0, \pi]$, then moving into the wind
17: **end**

Fig. 3 describes criterions for tacking and wearing with threshold values on it, which takes $\theta_0 \in (0, \underline{\theta})$ for example, but the principle can easily be extended to other quadrants. Drawing a circle of radius r with the starting point (x_0, y_0) on it, the orientation of the tangent line to this circle at (x_0, y_0) is the given heading θ_0. Define the final heading θ_T, where $\theta_T \in (\bar{\theta}, \pi]$ for $\theta_0 \in [0, \underline{\theta})$ for a tacking maneuver. A tangent line with θ_T intersects the horizontal line $y = y_T$ at (x_1, y_T) (i.e. the red point in Fig. 3(a)). Since we have assumed that this surface sailing vessel should not start or end with turning maneuvers, which is true in practice, we are not capable of reaching points $x_T \le x_1$ in this case by tacking once, so wearing around is proposed as the first maneuver (there might be a combination of different maneuvers).

For these target points satisfying $x_T > x_1$, selection of the right maneuver, when beating to windward, must take into account the distance between two established points, i.e. $y_T > y_1$ (y_1 is formed in Fig. 3(b)) ensures that there is enough space for turning with ra-

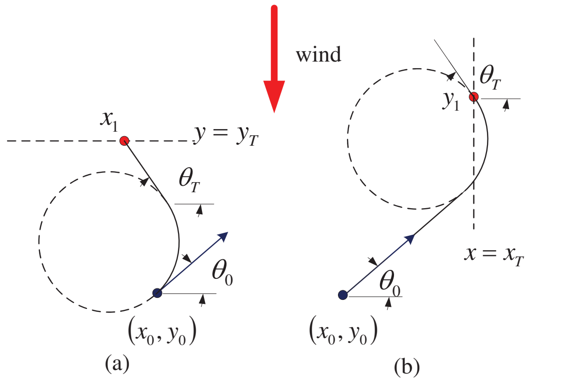

Fig. 3. Criterions for tacking and wearing. x_1 and y_1 are two threshold values.

dius r. In addition, tacking requires to check the entering velocity $v_{\underline{\theta}}$ (showed in Fig. 2(a)). If $v_{\underline{\theta}} <= v_{min}$, the sailing vessel must go wearing instead.

The algorithm we propose for generating a feasible path between two given postures (x_0, y_0, θ_0) and (x_T, y_T) is shown in Table 1. In the following, Fig. 4 is dedicated to describe the transformation of initial configurations, while (x_0', y_0', θ_0') representing the new starting points. On the other hand, as shown in Fig. 5, if the two points are too close to jibe when going down the wind, a tacking maneuver can be applied instead.

2.3. On-line path planning

As the surrounding environment is usually unstable and not perfectly known, e.g. the local wind conditions can change, we developed our algorithm to deal with uncertainties when the sailing vessels travel in the environment. In order to react to changes of the wind conditions in real-time, the algorithm should always be available with any reasonable initial velocity other than zero and any wind direction. There is only a minor change in the program by assigning $v(0) = v_0$ instead of $v(0) = 0$, where v_0 is the vehicle velocity when the environment changes.

Concerning wind direction, we assume firstly that the wind is coming from the north, i.e. the wind angle is $\alpha = -\pi/2$. Once the wind shifts (i.e. the new wind angle is α'), the Cartesian coordinates changes as well, and the position of the vehicle in the new coordinate system is $(x_0', y_0')^T = \mathbf{R}(\triangle\alpha)(x_0, y_0)^T$. The rotation matrix induced by the wind shift is

$$\mathbf{R} = \begin{pmatrix} \cos(\triangle\alpha) & -\sin(\triangle\alpha) \\ \sin(\triangle\alpha) & \cos(\triangle\alpha) \end{pmatrix}, \qquad (6)$$

in which $\triangle\alpha$ is the difference between the wind angles. Similarly, the specified target is also tranformed in a new frame as well as the initial heading. Consequently, according to terminal conditions (x_0', y_0', θ_0')

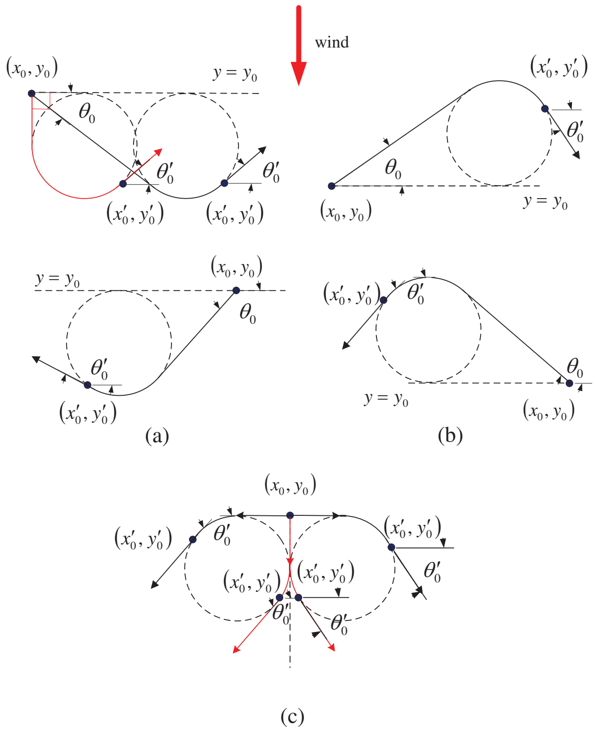

Fig. 4. Initial configuration transformation. (a) $y_T \geq y_0$ and $\theta_0 \in (-\pi, 0)$. Red line indicates that $\theta_0 = -\frac{\pi}{2}$. (b) $y_T < y_0$ and $\theta_0 \in (0, \pi)$. (c) $y_T < y_0$ and $\theta_0 = 0, \pm\pi, -\frac{\pi}{2}$.

and (x_T', y_T'), the path is replanned by using the path generation algorithm.

3. FORMULATION OF A FEEDFORWARD CONTROLLER

Once the path is defined, the objective is to find appropriate control signals that will steer the vehicle along these paths. It can be seen from the way creating path, a sequence of constant inputs is used, i.e. always using the upper or lower bounds in steering angles ($\overline{\delta}$ or $-\overline{\delta}$) during turns. In other words, the input function $\delta(t)$ is a piecewise-constant function. The problem of solving for the controls becomes to one of finding the switching times, which looks similar to bang-bang control strategies [10]. Here, the control $\delta(t) = \delta_i$, with $t \in [t_{i-1}, t_i)$, and the time-duration of δ_i being applied is $T_i = t_i - t_{i-1}$ ($t_0 = 0$), which explicitly shows that the behaviour of the system in the ith segment also depends on the previous times. Similarly, only $\overline{F}$ and $-\overline{F}$ are considered for the propulsive force $F(t)$ (0 can also be included because of the existence of the deadzone in tacking) such that $F(t) = F_i, t \in [t_{i-1}, t_i)$.

Contrary to [10], in which the bang-bang control problems were solved numerically by using of Newton's method and by solving the formulated ODEs as

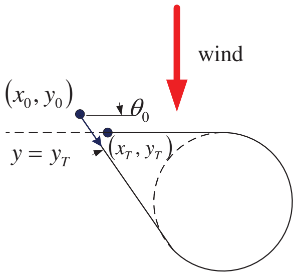

Fig. 5. Tacking maneuver when going downwind.

initial value problems, we regard this issue as TPBVP and search for appropriate switchings. To do this, as represented in Fig. 6, the basic dynamics of system (1) can be simply described as follows:

$$\begin{cases} \dot{s}(t) = v(t), & s(0) = 0, \quad s(T) = s_T, \\ m\dot{v}(t) + dv(t) = F(t), & v(0) = 0, \quad v(T) = 0, \end{cases} \tag{7}$$

where $s_T = \sum_{i=1}^{n} s_i$.

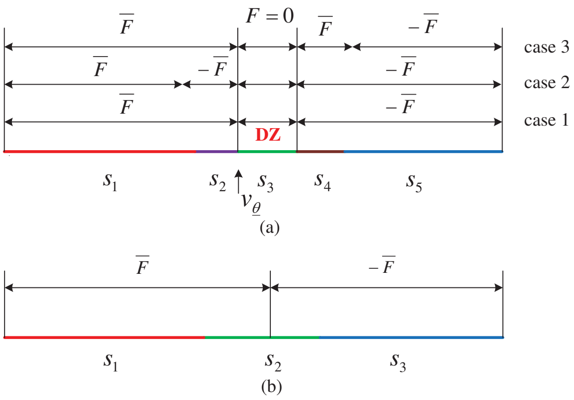

Fig. 6. Force profile for (a) tacking and (b) wearing.

Fig. 6(a) indicates that three different successive input levels ($\{\overline{F}, 0, -\overline{F}\}$, $\{\overline{F}, -\overline{F}, 0, -\overline{F}\}$, and $\{\overline{F}, 0, \overline{F}, -\overline{F}\}$) are available according to the controllability issues that are linked with the saturation levels on the input with the deadzone, i.e. (see the algorithm in Table 2) decelerating before the deadzone to avoid overshooting the target (case 2), or accelerating at the exit of the deadzone to avoid 'undershooting" (case 3). In Fig. 6(b), only $\overline{F}$ and $-\overline{F}$ are needed for the start/stop maneuvers in wearing.

Afterwards, based on the one-dimensional dynamics (7), characterize distinct cases in the first-order form together with boundary conditions similarly to (5), so that the time-durations T_i are derived and the switching times t_i for $F(t)$ are therefore figured out.

Whereafter, by using function $s(t)$ obtained from solving (7), together with segment length s_i, it is straig-

Table 2. Motion planning algorithm.

Name : MOTION PLANNING ALGORITHM
Goal : To generate a sequence of control signals for tacking maneuver.

1: enter boundary conditions $(0, 0)$, $(s_T, 0)$
2: assume only three successive input levels ($\overline{F}(T_1)$, $F = 0(T_2)$, $-\overline{F}(T_3)$) are used and compute $s(T_3)$
3: **if** $s(T_3) == s_T$
4: case 1, apply control input $F(t) = F_i, i = 1, 2, 3$.
5: **elseif** $s(T_3) > s_T$
6: case 2, compute $\overline{F}(T_1)$, $-\overline{F}(T_2)$, $F = 0(T_3)$, $-\overline{F}(T_4)$ and apply control input $F(t) = F_i$, $i = 1, 2, 3, 4$.
7: **else** (i.e. $s(T_3) < s_T$)
8: case 3, compute $\overline{F}(T_1)$, $F = 0(T_2)$, $\overline{F}(T_3)$, $-\overline{F}(T_4)$ and apply control input $F(t) = F_i, i = 1, 2, 3, 4$.
9: **end**

htforward to get switching times in the steering function $\delta(t)$ by interpolation, i.e. to find t_i at the point s_i in the function $s(t)$. Eventually, $F(t)$ and $\delta(t)$ are both derived as propulsion and steering for the surface sailing vessels.

4. SIMULATION RESULTS

As an illustration of the proposed path generation algorithm and the behaviour of feedforward controllers, a few simulation results are stated, with parameters $L = 3m$, $m = 150kg$, and $d = 135$. Furthermore, the BVP solver *bvp4c* from MatLab is adopted in order to solve TPBVPs of the present study.

Based on our algorithm, it is practical for the surface sailing vessels to start moving (i.e. with zero initial velocity) from the starting point (x_0, y_0) with any initial heading outside the deadzone to any end point (x_T, y_T) in the plane. Fig. 7 presents the trajectory from $(10, 10)$ to $(-10, 30)$ with $\theta_0 = -3\pi/4$ and the system subjects to wind from the north.

Besides, in order to show the implementation of the on-line motion planning, certain trajectories were calculated and corresponding controllers were generated to guide the system. Originally, the wind is coming from the north and the vehicle is about to go upwind from $(10, 10)$ to $(20, 40)$ with $\theta_0 = \pi/6$ and $v_0 = 0$. Accordingly, the trajectory is supposed to be as the blue line in Fig. 8. However, when the vehicle travels through the point $(22, 22.5)$, the wind suddently changes direction so that the vehicle can not follow the previous trajectory, because the system would en-

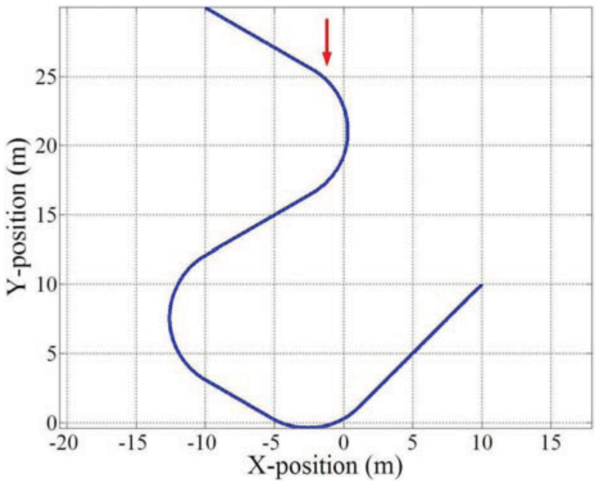

Fig. 7. Trajectory followed by the vehicle with red arrow indicating where the wind comes from.

ter the no-sailing zone under the new wind condition and could possibly get stuck in the end. After recalculation, the vehicle should follow the red route to reach the destination after the environment changes.

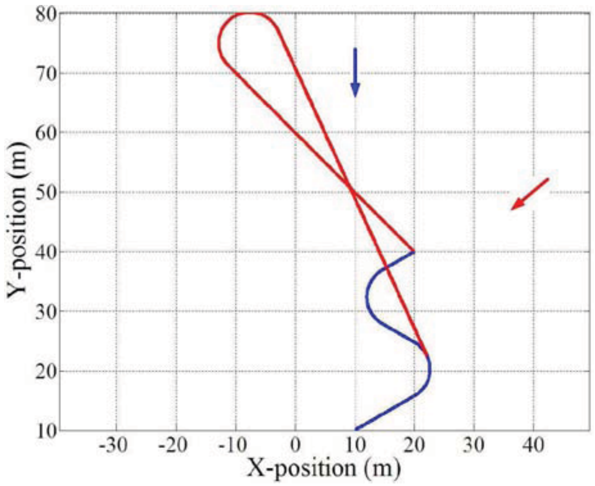

Fig. 8. Trajectories followed by the vehicle when the wind shifts. Blue and red arrows indicate that the wind direction alters.

5. CONCLUDING REMARKS

The method of motion planning for a class of surface sailing vessels is an expansion on our previous work. The introduction of wind condition changes does not impose any significant problems for the path planning, and makes the process more similar to real sailing. Each newly occurred situation can be in a natural way added to the algorithm by amending boundary conditions. Further work will include computation of optimum trajectory as well as time optimization.

6. REFERENCES

[1] B. Houska1 and M. Diehl, "Optimal control of towing kites," *Proceedings of the 45th IEEE Conference on Decision and Control*, PP. 2693 - 2697, San Diego, USA, 2006.

[2] C. Tomlin, I. Mitchell, and R. Ghosh, "Safety verification of conflict resolution maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 2, NO. 2, PP. 110-120, 2001.

[3] E. C. Yeh and J. C. Bin, "Fuzzy control for self steering of a sailboat," *Singapour International Conference on Intelligent Control and Instrumentation, 1992, Proc.*, Vol. 12, PP. 1339-1344, 1992.

[4] J. Jouffroy, "A control strategy for steering an autonomous surface sailing vehicle in a tacking maneuver," *In Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, San Antonio, Texas, 2009.

[5] J. Jouffroy, "On steering a sailing ship in a wearing maneuver," *IFAC International Conference on Maneuvering and Control of Marine Craft*, PP. 26 - 31, Brazil, 2009.

[6] J.-P. Laumond, S. Sekhavat, and F. Lamiraux, *Guidelines in Nonholonomic Motion Planning for Mobile Robots*, Lectures notes in Control and Information Sciences, Vol. 229/1998, PP. 1 - 53, Springer Berlin/Heidelberg, 1998.

[7] L. F. Shampine, J. Kierzenka, and M. W. Reichelt, "Solving boundary value problems for ordinary differential equations in MATLAB with bvp4c," *Available at ftp://ftp.mathworks.com/pub/doc/papers/bvp/*, 2000.

[8] L. Xiao and J. Jouffroy, "Reflexions on feedforward control strategies for a class of sailing vehicles," *IFAC Conference on Control Applications in Marine Systems*, Rostock, Germany, 2010.

[9] R. Stelzer and T. Rröll, "Autonomous sailboat navigation for short course racing," *Robotics and Autonomous Systems*, Vol. 56, PP. 604 - 614, 2008.

[10] S. K. Lucas and C. Y. Kaya, "Switching-time computation for bang-bang control laws," *Proceedings of the 2001 American Control Conference*, PP. 176 - 181, 2001.