Auto-Heading Controller for an Autonomous Sailboat

Nuno A. Cruz and José C. Alves
INESC Porto and FEUP–DEEC
Rua Dr. Roberto Frias; 4200-465 Porto, Portugal
Email: {nacruz, jca}@fe.up.pt

***Abstract*— This paper addresses the design and implementation of feedback controllers for the direction of autonomous robotic sailboats. In order to design such a controller, it is important to determine a model for the sailboat dynamics during turns. However, there are many uncontrollable factors that may affect the direction of the sailboat, which make it difficult to obtain an accurate model and require a lot of sensors to feed a proper controller. Instead, we assume a rather simple model relating the most important variables and concentrate on data that can easily be available with simple low-cost sensors, compensating the lack of accuracy of the model with the robustness of the controller. We describe our approach to extract the parameters of such a dynamic model using data obtained in field experiments and we show how to use this model to tune a PI controller. As a case study, we use the FASt vehicle, a 2.5 m long robotic sailing boat capable of fully autonomous navigation through a set of predefined marks. Experimental results show the performance of the designed controller.**

I. Introduction

Autonomous sailboats are an emerging technology available for the ocean community, with the ability to sample vast regions of the oceans. Even though they can only navigate at moderate velocities, these systems can transport a wide variety of sensors and transmit data to shore in real time, providing valuable ocean data in spatial and temporal scales complementary to the other technologies already in use.

In conventional sailing boats, the sailor controls the rudder according to the desired course and adjusts the sails to maximize velocity. The overall sailing performance depends on unpredictable environmental conditions such as wind and sea state. For a given course, boat speed, wind speed and wind direction, there is an optimum angle between the sail and the direction of the wind that maximizes the speed of the boat. Autonomous (or *robotic*) sailboats also use the wind for propulsion, but they rely on electronic systems to set the course, control the sails and act on the rudders, using power harvested from the environment, and without any form of human intervention. There have been already several examples of fully autonomous sailboats demonstrated in the field, and, naturally, the emphasis is being shifted towards performance. Current implementations are aiming at a continuous presence in the ocean [1], however, for the autonomous sailboats to be most useful in a wide range of practical scenarios, it is also important to ensure that they can keep the trajectory within a reasonable distance from what is specified in a given *mission script*. One of these scenarios is a search mission using sonar data, for example, in which a sailboat has to follow specific trajectories to ensure full coverage of the ocean floor.

In autonomous vessels using more conventional propulsion (electrical or combustion), the heading controllers are designed together with velocity controllers, since the velocity determines the turning behavior. In the case of autonomous sailboats, however, there is very limited control on velocity. Instead, since the energy required for propulsion is virtually *free*, the sail is usually set in such a way as to try to maximize the velocity. However, this velocity can change dramatically in short spacial and temporal scales, both due to the wind itself and to vessel maneuvers such as turning. For example, the driving force acting on the vessel changes during a turn, as the apparent wind rotates, yielding complex variations on speed, largely depending on local wind conditions. More, the heel angle affects the hydrodynamics of the sailboat, becoming asymmetric and inducing an additional turning torque. Therefore, the design of feedback controllers for auto-heading has to take into account the high variability of the environment and their consequences in terms of vessel behavior during turning maneuvers.

The FAST project

The FEUP[1] autonomous sailboat (FASt) is a small unmanned sailing yacht capable of fully autonomous navigation through a predefined set of marks (Fig. 1). This boat was custom designed and built by a team of faculty and students at FEUP, and it is a flexible autonomous navigation platform, being able to carry a few kilograms of payload equipment [2]. Although the work described in this paper was developed for this specific boat, the same methodology may be applied to design heading controllers for other sailboats.

The main dimensions of FASt are presented in table I. The design length was set to 2.5 meters, after scaling down, in length and displacement, some modern designs of oceanic sailing boats and keeping the total weight relatively low to facilitate the launch and transportation, either by towing or on the top of a car. To increase stability, the boat has a deep keel with a lead ballast. The rig is a conventional Marconi configuration with a headsail rigged on a small boom, as used in smaller RC sailing boats.

[1]Faculty of Engineering at the University of Porto, in Portugal

Fig. 1. FASt – the FEUP Autonomous Sailboat.

TABLE I
MAIN DIMENSIONS OF THE FEUP AUTONOMOUS SAILBOAT - FASt

Total length (LOA)	2.50 m
Maximum width (beam)	0.67 m
Draft	1.25 m
Mast height	3.40 m
Displacement	50 Kg

Paper organization

This paper is organized as follows. Section II provides some background regarding heading controllers for sailboats. In Section III we describe the model derived for the turning maneuver. Next, in Section IV, we show how to extract the parameters of the model, based on data obtained during field tests. In Section V, we use the model for the turning maneuver to design a PI controller based on classic frequency analysis, and finally, Section VI presents the main conclusions and the plans for the near future.

II. HEADING CONTROLLERS

From several decades sailors have been using mechanisms to provide the automatic steering of sailing boats. The autopilot is now a valuable accessory in modern sailing yachts, allowing the automatic control of helm according to some reference direction (GPS, compass or wind). First popular autopilots were fully mechanical, providing an effortless way to maintain a constant angle with the apparent wind without requiring any electric power for its operation [3]. However, these *windvane* autopilots exhibit important limitations because the reference is always the wind direction and the course stability may be severely affected in downwind legs due to large fluctuations of the apparent wind direction. With the advent of low cost and low power electronics, sophisticated autopilots became available for almost any type of sailing yacht, including features as automatic tacking or dynamic adjustment of control parameters to adapt to varying sea conditions.

While in downwind legs the course control objective is to maintain a given heading irrespective to the wind direction, for close hauled navigation a helm controller must be able to keep the apparent wind angle as close as possible to a desired value. Because the apparent wind angle varies constantly due mainly to wind gusts and boat speed variations induced by waves, trying to keep a course to maintain that angle would require excessive helm actuation, which translates into unwanted power consumption. Modern autopilots actually navigate close hauled by maintaining an heading reference and adjusting it periodically based on long term averaging of the wind direction.

Although commercial autopilots already exist, physical aspects like their size, weight and power consumption are not adequate for integration in autonomous (and small) sailing boats. Besides, the dynamics of small sailboats differ significantly from real size sailing boats and are much more sensitive to external perturbations due to waves or wind gusts. In order to mimic the human operation of sailing tasks, several works have explored non-classic control techniques to tackle the non-linear nature of a sailing boat, based on fuzzy-set theory [4], [5], neural networks [6], [7] or other artificial intelligence techniques [8].

The heading control procedure for a sailing autopilot must be decoupled from sailing maneuvers and used only during stable navigation, when the objective is to keep a constant heading reference. Maneuvers like tacking, gibing or fast course changing require a close coordination with sail trimming and may expose the boat to high dynamic forces, resulting in abrupt changes in the heading, speed and heeling (lateral inclination). While these maneuvers cannot be handled by classic linear controllers (thus requiring other non-linear techniques as described above), a simple proportional-integral controller may be effective for heading control, exhibiting very low computational requirements. This is an important issue in the scope of fully autonomous sailing boats for long term missions because computing power requirements translates to energy consumption and most of the time a sailing boat is navigating under heading control.

III. A MODEL REPRESENTATION OF THE TURNING MANEUVER

In order to design feedback controllers for the direction of the sailboat, it is important to have a realistic model for the relationship between the actuation (rudders) and the resulting yaw angle (or heading). The performance of the feedback controllers depends on the accuracy of this dynamic model. However, it should be stressed that the implementation on a real vessel also has to take into account some practical issues, such as the difficulty in measuring or estimating some of the model variables. Therefore, our idea is to build a sufficiently accurate model, for which a set of robust controllers may be designed, but also for which it will be possible to measure or estimate the required data, preferably using only low cost, although reliable, sensors. We assume that the adjustment of the sail is performed automatically, through a separate process, in order to maximize the sailboat velocity. Therefore, the heading controller only has to deal with variations with respect to the desired (or *reference*) heading.

As a first approximation, the vehicle dynamics are dominated by an inertia term and a drag term, when subjected to a given torque, $T(t)$, resulting in a simple relationship:

$$J\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} = T(t) \tag{1}$$

where J is the moment of inertia, B is the viscous term and θ is the yaw angle.

The main torque input is provided by the rudders and it is a function of the rudder angle and the vehicle speed with respect to the water, $T_r(t) = f(\delta, v_w)$. There are other sources of torque, such as the asymmetries of the hull, the wind and wave effects on the hull, the heel angle, the sail motion, etc. However, if the vehicle speed is high enough and the rudders are being actuated, then these other sources can be seen as disturbances and neglected. Therefore, the above relationship can be approximated as a first order equation relating the torque provided by the rudders with yaw *rate*, *i.e.*:

$$J\frac{d\omega}{dt} + B\omega \approx T_r(t) \tag{2}$$

where $\omega = \frac{d\theta}{dt}$.

The effect of rudder actuation in the rotation of the sailboat depends on the velocity of the boat with respect to the surrounding water, v_w, but this measurement is not available in our sailboat. The velocity of the boat can also be indirectly estimated based on sail position, wind speed and direction, heel angle, etc., for which there are usually sensors available on board. Note that this relationship is extremely complex, particularly during turns, because as soon as the vehicle changes orientation, the wind effect on the sails also changes and all the dynamics are affected, even if the wind remains constant. Instead, a readily available information is absolute velocity, v_g, provided by even the simplest GPS receivers. This velocity is typically calculated using differences in GPS position updates, which yields long delays due to filtering and also coarse accuracies, specially for low velocities. In our case, we consider that the local currents are neglectable and that the sailboat velocity is very smooth, and so, we assume $v_w \approx v_g$, with v_g being given by the GPS outputs. An additional source of error occurs during turns, since the GPS estimate of velocity usually assumes that the vehicle navigates in a straight line, instead of describing a circumference. Since the tangential velocity is higher in a curve than if it were in a straight line, because it has a longer path for the same difference in time, the velocity estimation will be lower than the true value. However, typical angular velocities account to a few tens of degrees per second, for which this error may be neglected (simple geometry calculations show a 2% error for 40 deg/s).

From our experience with the FASt sailboat, we've found that a step in the rudders produces a rapid change in heading, with a nearly constant angular velocity, which indicates that equation 2 may be a good approximation. However, since we are neglecting many factors affecting the dynamics during turns, it is wise to consider a more complex model for the turning maneuver, that uses a second order relationship between torque and yaw *rate*. In order to get yaw, this output has to be integrated, resulting in a 3^{rd} order model to relate the torque and the output heading. Although there seems to be no significant gains in terms of model accuracy, the fact that we use a more realistic higher-order model has consequences in the design of the feedback controllers, stressing the need to account for stability issues.

In other robotic vessels using electrical or combustion motors, it is possible to produce constant thrust and, independently, test different levels of rudder actuation to verify the torque produced. In sailboats, the velocity can be estimated in real time, but it cannot be controlled. However, if we consider a simple model for its influence on the torque, then it is possible to compensate for it in real time. We linearize the combined effects of rudder angle and vessel velocity, assuming that $T_r(t) \approx k_{rv}\delta(t)v_g(t)$. Our approximate model for the turning maneuver is then:

$$\alpha\frac{d^2\omega}{dt^2} + \beta\frac{d\omega}{dt} + \gamma\omega = k_{rv}\,\delta(t)\,v_g(t) \tag{3}$$

Note that for the design of the feedback controllers, the above relationship allows for a *normalized* transfer function between rudder angle and yaw angle, considering a unity velocity. Since the velocity acts as an additional gain, all that is required is that the overall controller gain be adjusted to compensate for the velocity measured in real time. This will be detailed further when discussing the design of the controller, in section V.

IV. Parameter Identification

Further to derive the dynamic relationships between rudder and heading, as modeled in the previous section, it is paramount to find the specific parameters for a given vessel. In other dynamic systems, the model parameters are usually obtained by the use of instrumentation on board and performing tests under controlled conditions. However, sailboat hulls are highly hydrodynamic and any external devices installed on the hull may severely affect the results. Since the vehicle dynamics are greatly affected by the environment (sea state, winds, etc.), it is preferable to design identification tests in the field, rather than relying on theoretical parameters or unrealistic data obtained in tow tanks (typical sailboat sizes also make it hard the use of tow tanks for parameter identification). The main difficulty in this approach is that these tests have somehow to encompass the possible variations on the environment. This can be accomplished if the data set includes enough transitions and different levels of rudder actuation. More, the relationship between rudder angle and yaw depends on the speed through the water, which is difficult to estimate with accuracy, as explained earlier. It is therefore important to use a sufficiently large set of data and to avoid local currents, so that $v_g \approx v_w$.

In order to find the best parameters for the model, we use an exhaustive search procedure where we test different values and produce a measure of adequacy. In practice, for each set of parameters, we compare the real data obtained in the field with the output from a model implemented in Simulink, resulting in a vector of differences. An overview of this procedure can be seen in figure 2.

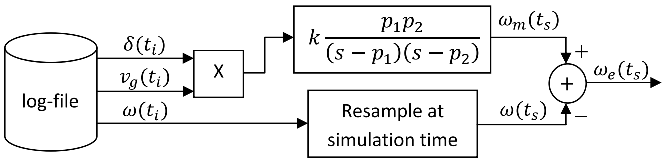

Fig. 2. Parameter identification method. The logged values of rudder angle and velocity are the input of the model. The values of angular velocity obtained using the model, $w_m(t_s)$, are compared to the values logged by the vehicle, $w(t_s)$, in order to estimate the model errors, $w_e(t_s)$.

Note that the inputs for the Simulink model are the same commands given to the sailboat rudders, $\delta(t_i)$, together with the velocity logged at the same time instant, $v_g(t_i)$. The Simulink model produces a vector of angular velocities at time samples t_s, $\omega_m(t_s)$. The values of angular velocity logged on the on board computer, $\omega(t_i)$, are resampled in the same simulation, to produce a similar sized vector for comparison, $\omega(t_s)$.

Our figure of merit, M, is given by the squared difference between the values obtained using the model and the values logged during the field test:

$$M = \sum_{\text{all } t_s} \omega_e^2(t_s) = \sum_{\text{all } t_s} (\omega_m(t_s) - \omega(t_s))^2 \tag{4}$$

Naturally, the best parameters for the model are the values of p_1, p_2, and k that minimize such figure of merit.

The choice of an exhaustive search was motivated by the simplicity of the model. For each set of values of p_1, p_2, and k, each simulation takes only a few seconds to complete, even for thousands of logged values. More, this approach ensures that the absolute minimum for M can always be found, providing we choose a sufficiently detailed resolution for the search space.

To extract the parameters for the FASt sailboat, we started by simple measurements of heading reaction to steps in the rudders, at approximately constant values of speed over ground. This allowed us to estimate a time constant in the order of a few seconds, which was a good starting point for the iterative process of finding the open loop poles (p_1 and p_2) and the gain of the transfer function, k. Using the method described above, the parameters which resulted in the minimum value for M were $p_1 = -0.7$, $p_2 = -0.6$, and $k = 0.8$. In order to have a visual confirmation of these parameters, we plotted both the model output and the logged data in the same plot, as shown in fig. 3. Note that the values used for angular velocity are very noisy, since they were computed using discrete differentiation of yaw angles obtained at sea. This differentiation amplifies the noise of the digital compass and also suffers from small inaccuracies in time measurements. In any case, it can be seen that the model approximation seems to be quite accurate.

V. Design and validation of the controller

Using the dynamic model previously described, we used classic control theory to design a stable feedback controller for auto-heading, both for compass heading reference or apparent wind angle reference. The only difference between these two cases is in the calculation on the *error* term that feeds the controller, coming either from the difference between the reference and the instantaneous compass heading, or from the reference and the direction of the apparent wind, respectively. Note that even if the steady state error is zero in heading, the sailboat may describe a trajectory parallel to the desired path, for example in the case of lateral currents. In this case it is useful to complement these controllers, taking into account not only an error in heading, but also a cross-track error between the vessel position and a given trajectory.

The model identified for the turning maneuver scales approximately linearly with the velocity, but the sailboat velocity is greatly dictated by external conditions (mainly wind speed and direction) and cannot be controlled. In terms of transfer function, however, this value acts simply as an additional gain. Since we can estimate this gain by measuring v_g, then it is possible to *normalize* the transfer function for a standard value of velocity (1 m/s), just by multiplying any standard controller for the proper scaling factor, $\frac{1}{v_g}$. From the model parameters described above, the *normalized* transfer function is then:

$$\frac{\Theta(s)}{\Delta(s)} = \frac{k\, p_1\, p_2}{s(s-p_1)(s-p_2)} = \frac{0.34}{s(s+0.7)(s+0.6)} \tag{5}$$

where $\Theta(s)$ is the Laplace Transform of the yaw angle, $\theta(t)$, and $\Delta(s)$ is the Laplace Transform of the rudder angle, $\delta(t)$.

The design of the controller will assume a standard configuration, such as presented in fig. 4. The controller input, $e(t)$, is the difference between the reference, $\theta_r(t)$, and the value of yaw, $\theta(t)$. The output of the controller is a rudder angle.

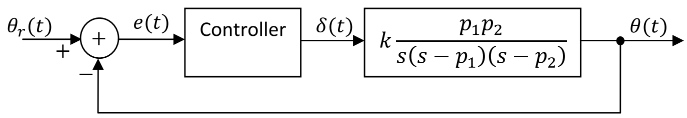

Fig. 4. Standard controller configuration.

Note that the transfer function of equation (5) above has a pole at the origin, since we are now considering the yaw angle as the output (which is the integral of the angular velocity). This means that the turning behavior can be represented as a type-1 system, *i.e.* a system which closed-loop response is able to suppress any disturbances at the output, even with a simple proportional controller. However, if there is an error in the input of the model, then the response of the feedback system

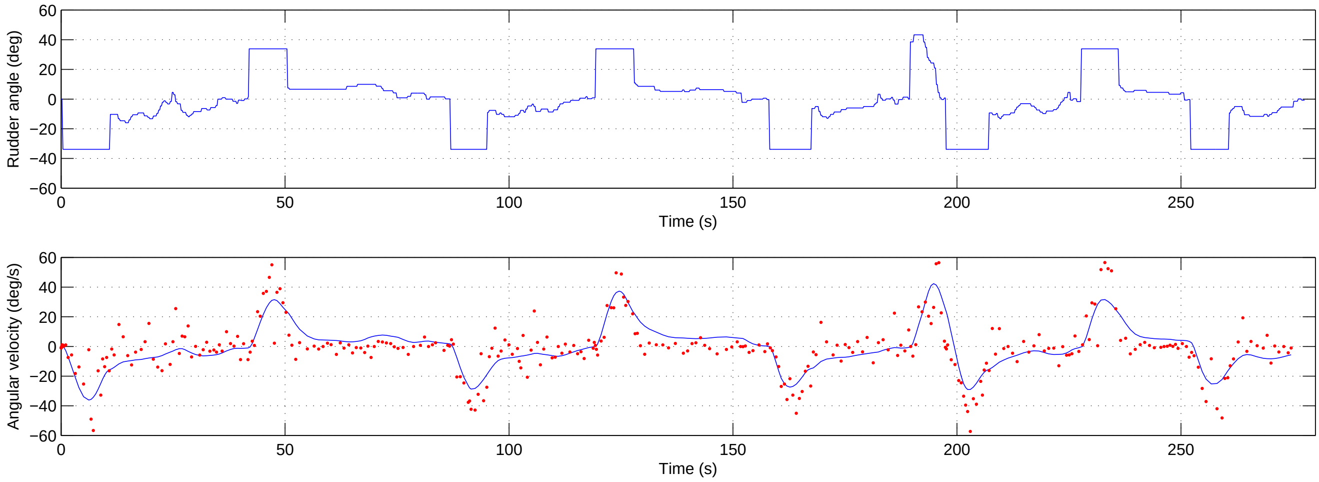

Fig. 3. Results of the procedure for parameter extraction. The bottom plot shows the values of angular velocity logged on board (dotted plot, in red) and the model approximation (solid line, in blue). The commands issued for the rudders can be seen in the top plot .

to a step input results in a steady state error. In our case, this happens because the sails are at the side of the sailboat, therefore producing a turning moment since the resulting forces are not properly aligned with the center of mass. Even though this moment is reduced, it results in an error in yaw. The same applies for any asymmetries in the hull or the rudders, which stresses the need for a controller with an integral component to remove any steady state error. The behavior of a derivative component in the controller is important to shape the response of the system, avoiding *overshoots*, but it requires a good measurement of the derivative of the control variable. That is not the case in our system, as could be seen in the previous section, so we decided to simply design a PI controller, which implements the following relationship:

$$\delta(t) = k_p\, e(t) + k_i \int e(t) \tag{6}$$

In the Laplace domain, the transfer function of the controller is given by:

$$G_{PI} = \frac{\Delta(s)}{E(s)} = k_p + \frac{k_i}{s} = k_c \frac{s + z_c}{s} \tag{7}$$

where $\Delta(s)$ is the Laplace Transform of $\delta(t)$, and $E(s)$ is the Laplace Transform of $e(t)$.

Tuning of the controller

There are several methods of tuning a PI controller and this last transfer function shows that such tuning can be seen as a choice of a zero, z_c, and a gain, k_c, for some criteria of performance. One way of choosing these parameters is based on frequency analysis. In figure 5, we can see in solid blue the Bode diagram of the plant before adding any controller, *i.e.* $\frac{\Theta(j\omega)}{\Delta(j\omega)}$. The PI controller adds a pole at the origin, which results in the phase being reduced in 90 degrees at all frequencies, as can be seen as the dotted-green plot in fig. 5. This pole at the origin is required to eliminate steady state errors in yaw, as described earlier, but this reduction of phase makes the system less stable. In order to have a stable feedback system, the phase margin has to be positive, which can be achieved with the proper choice of z_c and k_c. The zero of the controller (z_c) will affect both the phase and the gain, while the parameter k_c only affects the gain. For this reason, we start by choosing the zero, so that we add the required phase to ensure stability. We chose z_c to be around two decades less than p_1 and p_2, which ensures that the maximum value of phase goes to almost 90 degrees. Typically, phase margins are chosen between 45 and 60 degrees, so such a high value is not really necessary. However, this provides more flexibility for choosing the final phase margin, by setting the proper gain. The Bode diagram plotted in red with the (+) signs in fig. 5 shows the result of adding the pole at the origin and $z_c = 0.005$ rad/s to the original plant. With such a plot, we can decide on the value for the phase margin, up to about 75 degrees, by using the proper value of k_c. We decided for a phase margin of 45 degrees, which means the gain plot should cross the 0dB line at the frequency where the phase is -135^o. From the frequency response, we find this frequency to be 0.26 rad/s and, at this frequency, the gain is 8.4dB. Therefore, the value of k_c will have to decrease the gain by 8.4dB, which means $k_c = 10^{\frac{-8.4}{20}} = 0.38$. The complete PI controller is then given by:

$$G_{PI}(s) = 0.38\, \frac{s + 0.005}{s} \tag{8}$$

In order to implement this PI controller in the onboard software of the FASt sailboat, we compute the integral part of the error as a discrete summation, taking into account the control period, which was set to 200 ms. The command sent to the rudders is the result of the control law of equation (6), divided by v_g given by the onboard GPS. This serves to compensate for the extra gain provided by the velocity of the sailboat during turns. In practice, we only scale down the command if the velocity is greater than a certain threshold (1 m/s, in the current implementation), to avoid large values of

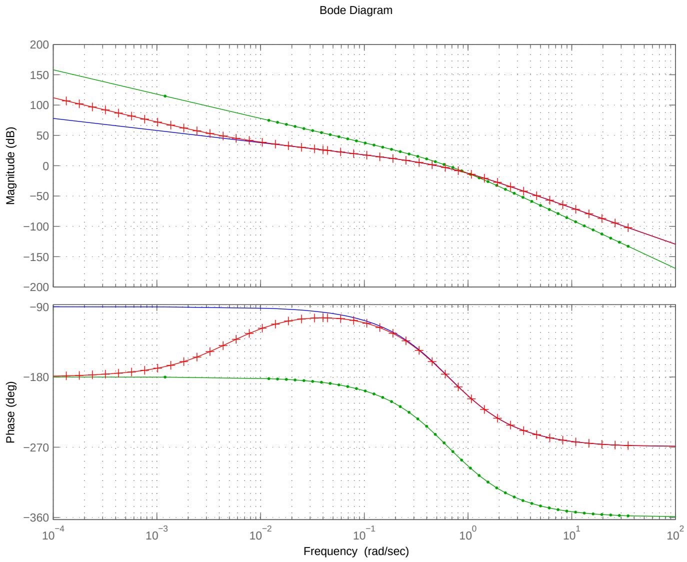

Fig. 5. Bode plots for the plant $\frac{0.34}{s(s+0.7)(s+0.6)}$. In solid blue, before adding any controller; in dotted-green, showing the influence of adding an extra pole at the origin; in red with the (+) signs, showing the influence of adding a pole at the origin and a zero, $z_c = 0.005$ rad/s.

rudder actuation in the case of the vehicle moving too slow. Finally, we are also aware of the possible effects of *integral wind-up* due to large errors in heading and, therefore, we limit the growth of the integral component.

One example of the performance of the above controller can be seen in fig. 6, taken from a field mission with FASt. For a big step in heading, we force the heading reference to change in small steps, in order to ensure that the sailboat rotates in the proper direction. Note there is a small overshoot at the end of the transition, motivated by the integral action, but the system is able to follow the heading reference (in blue).

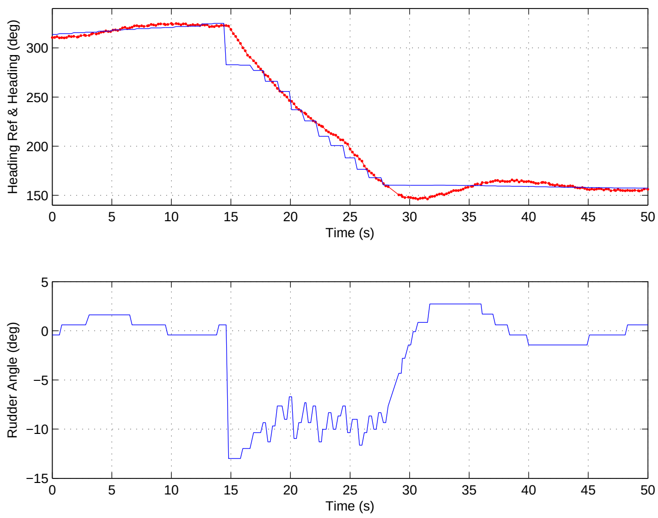

Fig. 6. Example of the performance of the heading controller. The first plot shows the heading reference, in solid blue, and the measured heading, in dotted-red. The bottom plot shows the commands sent to the rudders.

VI. Conclusions and Future Work

Until very recently, most of the efforts in developing autonomous sailboats were conducted towards the engineering challenges, either in electronics, mechanics or software programming. Now, the first prototypes are showing this is already an important tool to ensure a full time presence at the ocean, and the new challenges are addressing improvements in performance, such as the ability to follow specific directions with accuracy. In order to design auto-heading controllers, it is important to have good models for the sailboat dynamics, but sailboats are very complex systems.

In this paper, we've shown how to identify the parameters of a model for the turning maneuver of a sailboat, based on a search process and a comprehensive data set from field experiments. Since the rotation is affected both by the rudder angle and velocity, we compensate for the velocity measured in real time, in order to use a single normalized transfer function relating the rudder angle with yaw.

Using the model for the turning maneuver, we've tuned a PI controller based on frequency analysis and the performance of this controller was demonstrated in field tests. Even though the same controller can be used to maintain an absolute heading or a constant apparent wind angle, the measurement of wind direction is very noisy and therefore we typically define an absolute heading for a certain period of time.

Our main focus for the near future will be to improve the accuracy of the model, by integrating new low cost sensors, like gyroscopes and accelerometres. In addition to contribute to a better model, these sensors will provide real time data required to feed controllers with enhanced performance.

Acknowledgements

The authors would like to acknowledge the support from the Department of Electrical and Computer Engineering at FEUP and the companies sponsoring the construction of FASt.

References

[1] N. A. Cruz and J. C. Alves, "Autonomous sailboats: an emerging technology for ocean sampling and surveillance," in *Proceedings of the MTS-IEEE Conference Oceans'2008*, 2008.

[2] J. C. Alves and N. A. Cruz, "Fast - an autonomous sailing platform for oceanographic missions," in *Proceedings of the MTS-IEEE Conference Oceans'2008*, 2008.

[3] B. Belcher, *Wind-vane Self-Steering*. International Marine Publishing Company, 1982.

[4] R. Stelzer, T. Prol, and J. Robert I, "Fuzzy logic control system for autonomous sailboats," in *Proceedings of the IEEE International Fuzzy Systems Conference*, 2007, pp. 1–6.

[5] Y. Briere, F. L. C. Ribeiro, and M. A. V. Rosa, "Design methodologies for the control of an unmanned sailing robot," in *8th Conference on Manoeuvring and Control of Marine Craft (MCMC'2009)*, September 2009.

[6] A. Tiano, A. Zirilli, C. Yang, and C. Xiao, "A neural autopilot for sailing yachts," in *Proceedings of the 9th Mediterranean Conference on Control and Automation*, 2001, pp. 27–29.

[7] F. Fossati and M. Mauri, "Neural network control of sailboat dynamics: hardware real time implementation experience," in *ISC '07: Proceedings of the 10th IASTED International Conference on Intelligent Systems and Control*. Anaheim, CA, USA: ACTA Press, 2007, pp. 35–43.

[8] M. L. van Aartrijk, C. P. Tagliola, and P. W. Adriaans, "AI on the ocean: the robosail project," in *Proceedings of the European Conference on Artificial Intelligence*, 2002, pp. 653–657.