Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-23 (2016) 524–531

Construction and Control of an Autonomous Sail Boat

Henning Seeberg Stenersen *

* *Department of Engineering Cybernetics, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway (e-mail: henning.stenersen@gmail.com)*

Abstract: A small scale autonomous sail boat has successfully been designed, built and tested. It is capable of sailing a feasible predefined track with no further input from the user, using only the sails as propulsion.
The physics of sailing are tamed by the means of navigation, control and guidance. The boat has been equipped with flexible and powerful hardware for computerized control. This hardware gathers measurements, runs actuators and execute both high and low level control algorithms. A control allocation scheme for the sail boat has been proposed. The scheme ensurers progress in all feasible points of sail and has proved efficient in tests. A controller uses the scheme to close the heading loop and in turn the position loop is closed by a waypoint tracker.

Keywords: Autonomous sailing, autonomous control, embedded systems, path planning, sensors, autonomous mobile robots, autonomous vehicles.

1. INTRODUCTION

Compared to field robots, like UAVs, ROVs or AUVs, little research have been done in the field of robotic sailing. According to the *World Robotic Sailing Championship* (World Robotic Sailing Championship 2014) nobody yet has managed to cross the Atlantic ocean with a sailing robot. In 2011 it was stated that automated sail trim was not well covered by scientific publications (Stelzer and Jafarmadar. (2011)).

However, in resent years there has been some research on several aspects within the field, like: hardware design, system development, collision avoidance, path planning, rig design, controllers, modelling, stability and power management (Alexander Schlaefer (2011), Finnis (2012) , Fabrice Le Bars (2013)). Many prototypes of autonomous sail boats exists out there. These prototypes use a broad variety of boat designs, sail rigs used and control system designs.

The idea of transferring rules and common sayings from sailors into a controller is not a new idea. The use of *fuzzy controllers* to implement these rules has been tried by Abril J. (1997), Stelzer R. (2007) and Y. Briere (2009). These have all made real experiments that have showed working results. The control allocation scheme in the proceeding is also based on experience from practice in sailing, but the approach is different from fuzzy controllers.

The work presented in this paper is above all, personally, viewed as an interesting, challenging and relevant problem to solve. Previous work has already been conducted in (Stenersen (2015)). However, the autonomous sail boat may have some interesting applications. An autonomous sail boat can be constructed, not only to use the wind as propulsion, but to harvest energy for its internal systems. Hence, it can be completely self-sufficient and it has no crew cost. It can even carry other autonomous vehicles and supply them with energy, such as a flying drone. In the future we may see autonomous sail boats used for offshore inspection, surveillance and coast guard tasks, ice berg detection, scientific data gathering and more.

Main contributions are:

- Design and building of a small scale sail boat with necessary hardware to test and achieve autonomous sailing in relevant conditions.
- Design, make and enable a computer framework in order to:
  - Gather, log and make available a variety of measurements required for the application.
  - Execute high level control algorithms and provide a readable and modular implementation environment for such algorithms.
  - Interact with a smart phone user interface which enables efficient testing and running of the system.
- Suggestion of a control allocation scheme that ensures progress and controllability in all feasible points of sail, using only the sails as propulsion.

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.10.489

- Suggestion of simple control algorithms for autonomous sailing.
- Execution of several field tests where basic autonomous sailing was achieved successfully.

2. EMBEDDED COMPUTERIZED CONTROL

During the development, a computerized hardware framework was realized. The computerized hardware was developed with the following criteria in mind:

- Enable running a variety of high level control algorithms for sailing/navigation/etc. that require a higher amount of computational power.
- Enable running a variety of low level control laws prone to real-time requirements.
- Manage a variety of sensors and actuators in a modular, flexible and expandable manner.
- Allow for runtime user input.
- Provide performance results to the user, both in real-time and post operation.
- Remain operational and operate safely.

2.1 Realization

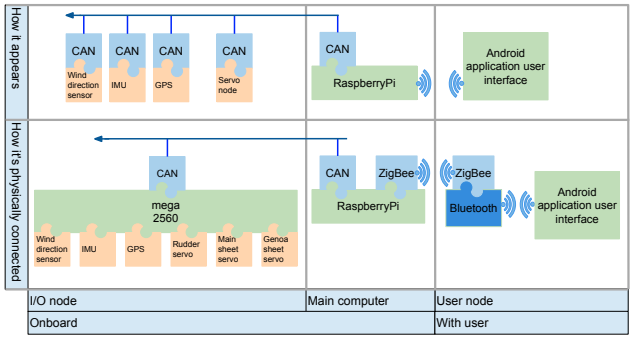

Fig. 1. Final system overview

In order to meet the system requirements, it was decided to use a high level computer running a common operating system. Hence, high level software are available to speed up the implementation of control algorithms. To achieve a modular, maintainable and expandable system, actuators and sensors were connected as nodes on an internal bus. The main computer would act as a master controlling the other nodes. Figure 2 shows the design overview of this design. The user interacts with the system through a wireless user interface.

The system was realized using a *Raspberry-Pi* computer with a common *Linux* distribution called *Rasbian* (Raspberrypi). The Raspberry-Pi computer was expanded with CAN bus (CAN bus). This was done by milling out a circuit board containing a *MCP2515* CAN-driver chip (Microchip), connecting it to the Raspberry-Pi and writing drivers for CAN-bus functionality. The drivers were installed as a module within the interpreted programming language*Python*.

It became somewhat cumbersome to implement one physical node for each sensor or actuator. Hence, it was thought better combine the basic sensors and actuators into a single physical node based on a powerful micro-controller with

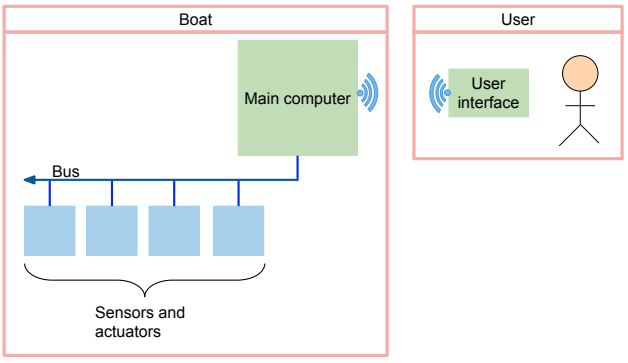

Fig. 2. Design overview

sufficient input/output/external peripherals. This would become the *ATmega2560* micro controller (Atmel). Each sensor and actuator would still appear as different nodes on the CAN-bus. Hence the system remained expandable as further physical nodes still could be connected.

In order to obtain a modular implementation of the shared resources on the ATmega2560, basic scheduling was achieved by modifying a port of *freeRTOS* onto it (FreeRTOS). This required more data memory and external RAM was fitted.

Figure 1 shows the final system overview, physically and how it appears.

3. NAVIGATION

The boat is equipped with an accelerometer, a magnetometer, a GPS and a wind direction sensor. Together, these instruments provide roll, pitch, heading, position, speed, course and wind direction for use in the control algorithms.

3.1 Roll and pitch

The accelerometer provides estimates for roll and pitch by assuming that the measured force vector equals the gravitational vector. This assumption holds well for this application as the sail boat is not subject to rapid acceleration.

3.2 Heading

Heading is obtained by calibrated magnetometer readings. A calibration routine is implemented to eliminate static hard and soft iron interference and scale factor errors. The resulting calibrated field vector is then rotated into the NED-frame and heading is obtained as follows:

$$\psi = 2\pi - atan2(X_{calRot}, Y_{calRot}), \quad (1)$$

where ψ is the heading and X_{calRot}, Y_{calRot} are the magnetic field components in north and east directions, respectively.

Filtering the heading measurement In order to filter out a noise component of ± 2-3 °, a low pass filter was considered. However, the phase lag became too large and an adaptive filter was designed in the following way:

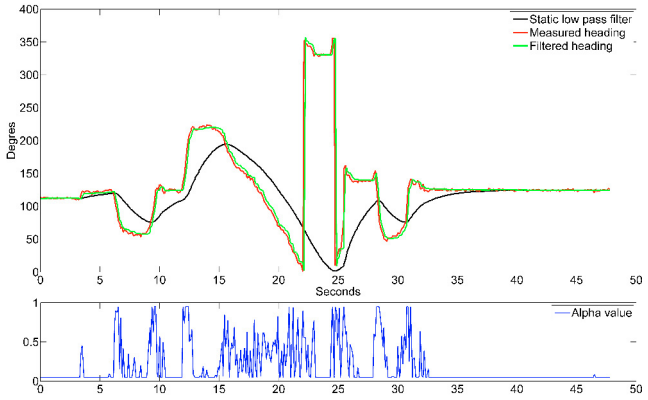

Fig. 3. The heading filter (green) applied to a heading sample (red). The filter's *alpha* value is shown below in blue. The black line is a static low pass filter with $alpha = 0.05$, which slowly converges to the heading filter.

$$\Delta[k] = circularDist(\psi[k-1], \psi) \tag{2}$$

$$\alpha_{\psi,var}[k] = sat(\beta(|\Delta[k]| - \Delta_{thres}), [0, \alpha_{\psi,max} - \alpha_{\psi,min}]) \tag{3}$$

$$\alpha_{\psi}[k] = \alpha_{\psi,min} + \alpha_{\psi,var}[k] \tag{4}$$

$$\gamma = \alpha_{\psi}[k](\psi[k-1] + \Delta[k]) + (1 - \alpha_{\psi}[k])\psi[k-1] \tag{5}$$

$$\psi[k] = circularMap(\gamma), \tag{6}$$

where $\alpha_{\psi,min} = 0.05$ and $\alpha_{\psi,max} = 0.9$ are the minimum and maximum smoothing factors, respectively. $\Delta_{thres} = 4$ is a threshold for increasing the smoothing factor, with respect to the difference in the measurement and the filtered value. $|\Delta[k]|$ is the absolute value of $\Delta[k]$. $\beta = 0.1$ decides the slope of the smoothing factor increase. Note that a high smoothing factor implies less smoothing. $sat(a, b)$ returns a saturated to the range b.

In Figure 3 and Figure 4 the filter is tested on a sample set. It is clear that the phase lag is low during large and rapid changes in heading, but still the smoothness remains high when the heading variation is small. The sample set was generated by bench testing, but the filter has significantly decreased actuator jitter during field tests.

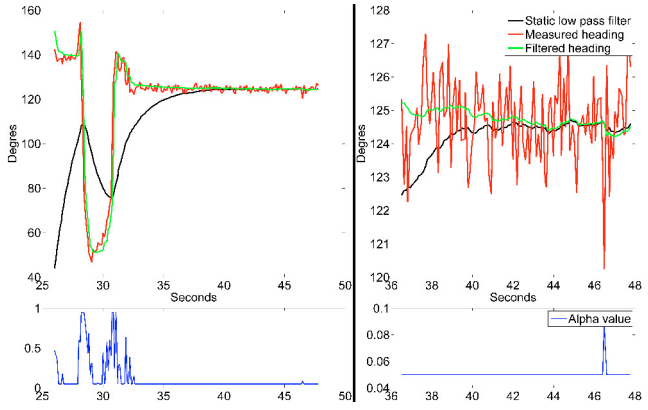

Fig. 4. The same heading sample and filters as in Figure 3.

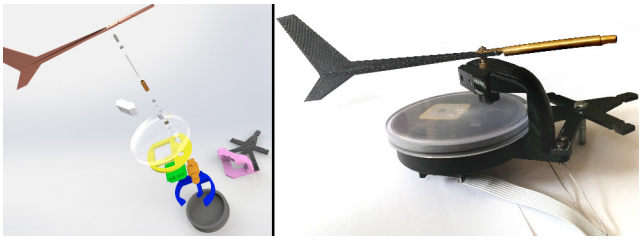

Fig. 5. Wind direction sensor and GPS housing assembly.

3.3 Wind direction

The relative wind direction ω_{rel} is obtained by reading the angle of a fin located on top of the mast, see Figure 5.

3.4 Position, speed and course

Position, speed and course are simply obtained directly from the GPS module.

4. CONTROL

4.1 Control allocation

Control allocation is the action of transforming, or mapping, control forces or moments into set-points for actuators. Hence the controllers at the lowest level runs the actuators towards the set-points and in conjunction with the craft the actuators produce the forces and moments ordered.

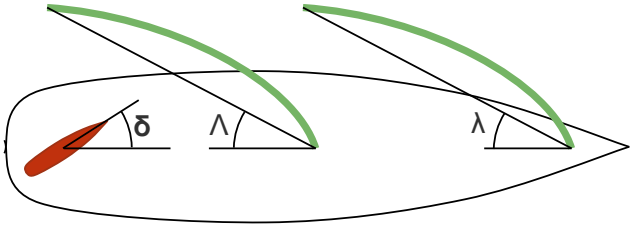

Fig. 6. Actuator angles.

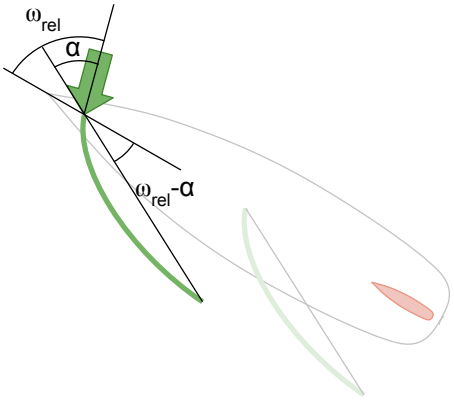

Fig. 7. Using the angle of attack to set desired sail angles.

Sail angles Λ and λ determines the set point angles for the main sail and the Genoa sail according to Figure 6. The higher level algorithms have little interest in Λ and λ themselves, but rather the generated lift. The lift is

dependent on the angle of attack, α, hence we define Λ and λ as:

$$\Lambda = \lambda = \omega_{rel} - \alpha \in [0°, 90°]. \tag{7}$$

It is no obvious choice for α. The optimal α value is dependent on the wind speed, rig configuration and other variables. Hence the α value remain as a tuning parameter available for live tuning through the user interface for now. Testing have shown that the α value only needs to be updated if there is a significant change in the wind speed.

Sail moment and rudder angle Higher level controllers will ask for a certain control moment which in turn makes the boat converge to the desired heading. There are two controllable components that generate such a moment:

- Rudder angle
- Difference in generated force by the main sail and the Genoa

In traditional sailing, the rudder is only used for fine tuning the heading. The greatest moment is set by adjusting the sails. In fact, the rudder may be omitted completely as on a wind surfing board. Hence, mapping a desirable moment should involve alterations to the main and Genoa sail angle. Thus a moment mapping function should include a main sail and Genoa sail angle contribution; Λ_{moment} and λ_{moment} respectively.

The new sail angles are altered in the following way to accommodate this:

$$\Lambda_{tot} = \Lambda + \Lambda_{moment} \in [0°, 90°] \tag{8}$$
$$\lambda_{tot} = \lambda + \lambda_{moment} \in [0°, 90°] \tag{9}$$

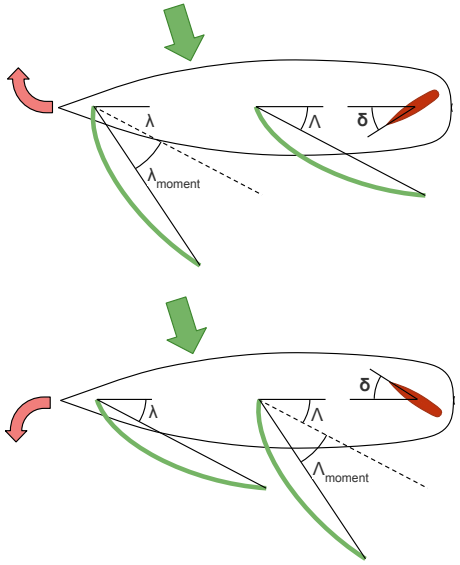

Fig. 8. Sail moment map. Green arrow indicates the wind direction.

In the following, a scheme is suggested to obtain Λ_{moment}, λ_{moment} and the rudder angle, δ:

Assuming the desirable moment is given as:

$$N_{des} \in [-1, 1], \tag{10}$$

where N_{des} is a virtual moment with no unit. The suggested moment map follows as:

$$\delta = \frac{N_{des}}{2}(\delta_{max} - \delta_{min}) \tag{11}$$

$$\epsilon = abs(N_{des}) - abs(N_{sail\ threshold}) \in [0, 1] \tag{12}$$

$$starboardTack = \begin{cases} true & \text{if } \omega_{rel} \in [0°, 180°) \\ false & \text{if } \omega_{rel} \in [180°, 360°) \end{cases} \tag{13}$$

$$\Lambda_{moment} = xor(starboardTack, isPositive(N_{des})) \epsilon \Lambda_{max\ contribution} \tag{14}$$

$$\lambda_{moment} = xnor(starboardTack, isPositive(N_{des})) \epsilon \lambda_{max\ contribution} \tag{15}$$

where $\Lambda_{max\ contribution} \geq 0$, $\lambda_{max\ contribution} \geq 0$ and $N_{sailthreshold} \geq 0$ are tuning parameters. These are available for live tuning through the user interface. $\omega_{rel} \in [0°, 360°)$ is the relative wind direction.

This moment map can more easily be described in words. The rudder is moved proportionally with the desired moment. After the certain threshold, $N_{sailthreshold}$, the respective sail eases out away from the wind proportionally to the desired moment by a scale of $\Lambda_{max\ contribution}$ for the main sail or $\lambda_{max\ contribution}$ for the Genoa sail. Figure 8 shows example scenarios.

This scheme has good effect in points of sail from and including close hauled down to and including beam reach. However, when running downwind there is no moment contribution from the sails as these are then both completely extended.

During testing of the boat, $\Lambda_{max\ contribution}$ and $\lambda_{max\ contribution}$ were both typically set to 45°and $N_{sailthreshold}$ to 0, but different configurations were tested. As the boat has a main sail that is approximately the same area as the Genoa sail, it makes sense for $\Lambda_{max\ contribution}$ to be equal to $\lambda_{max\ contribution}$.

Another way of doing this would be to tighten and shorten the two sails equally. However, it is not desirable to make a sail angle smaller as this may cause turbulence around that sail, which in turn lessens the sail efficiency.

As the rudder is proportional to the desired torque and the rudder dynamics are very fast, a torque is almost immediately produced when the desired torque changes. The response from the sails are of several magnitudes slower due to low-power actuators.

As the sail angles are limited to $\in [0°, 90°]$, the sails will remain at 90° when running downwind, which is desirable.

4.2 Modes of operation

Four modes of control operation have been defined:

- *Semi-manual*
- *Heading hold*
- *Waypoint tracking*
- *Path following*

These modes are described below:

Semi-manual In this mode the, user manipulates N_{des} through the graphical user interface. Hence, the user can operate the boat without considering the sails. Thus, as

long as the user directs the boat in a feasible direction, the sailing can be done with as much ease as operating a radio controlled car, where N_{des} corresponds to turning and α corresponds to throttling.

This mode of operation has proved itself very useful while testing not only the control allocation algorithm, but different aspects about the system. It has also been an effective tool for manually positioning the boat prior to testing of higher level algorithms.

Heading hold The *heading hold* mode uses a basic proportional controller to maintain a certain heading, ψ_{ref}:

$$e = \psi_{ref} - \psi \tag{16}$$
$$N_{des} = K_p e, \tag{17}$$

where $\psi \in [0°, 360°)$ is the measured heading, see Section 4.1, and N_{des} the desired torque, which at a lower level is saturated to $[-1, 1]$. $K_p > 0$ is the proportional tuning parameter. K_p and $\psi_{ref} \in [0°, 360°)$ are variables that can be changed live from the user interface.

Waypoint tracking The *waypoint tracking* mode is the simplest automated mode that closes the position loop. The overlying guidance system, see Section 5, passes waypoints (the desired position) down to this controller. The controller ensures progress towards the respective waypoint.

Progress towards the waypoint is simply ensured by passing the direction towards the waypoint as ψ_{ref} to the same heading controller as in *Heading hold* mode. This is a very simple approach that does not take sideslip into account.

The heading reference, ψ_{ref}, is calculated in the following way:

$$\mathbf{P}^n_{\Delta} = \mathbf{P}^n_{Ref.Waypoint} - \mathbf{P}^n_{boat} \tag{18}$$
$$\psi_{ref} = 90° - atan2(\mathbf{P}^n_{\Delta}), \tag{19}$$

where $\mathbf{P}^N_{Ref.Waypoint}$ and $\mathbf{P}^N_{boat}$ is the NED position of the waypoint and boat, respectively. $atan2(\mathbf{a}) \in [-\pi, \pi]$ returns the angle between the vector argument and the east axis, which is equivalent to $atan2(a_1, a_2)$.

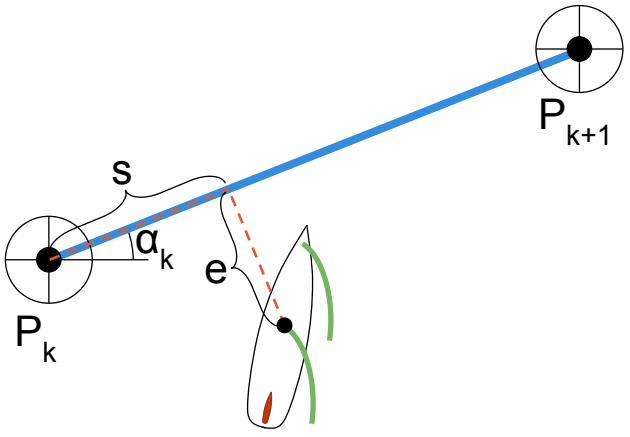

Fig. 9. Lookahead-based steering based on Fossen (2011).

Path following The *path following* mode is another implemented approach to close the position loop. The mode tries to not only ensure progress towards the next waypoint, but also remain on the line defined by the next waypoint and the previous waypoint. Hence, sideslip is taken into account.

The technique used to achieve this is called *Lookahead-based steering*(Breivik and Fossen (2009)). This technique is based on calculating the boat's distance perpendicular to the line, also known as the *cross-track error* e:

$$\alpha_k = atan2(\mathbf{P}_{k+1} - \mathbf{P}_k) \tag{20}$$
$$\begin{bmatrix} s \\ e \end{bmatrix} = \mathbf{R}(\alpha_k)^T(\mathbf{P}^n_{boat} - \mathbf{P}^n_k), \tag{21}$$

where α_k is the line heading, $\mathbf{P}_{k+1}$ and $\mathbf{P}_k$ is the next and previous waypoint respectively, s is the distance travelled along the line. $\mathbf{R}$ is the 2-dimensional rotation matrix for rotations about the z/down-axis:

$$R(\psi) = \begin{bmatrix} cos(\phi) & -sin(\phi) \\ sin(\phi) & cos(\phi) \end{bmatrix}, \tag{22}$$

The cross-track error e is used as negative feedback to the same heading controller as in *Heading hold* mode:

$$\chi_r = sat(K_p e + K_i \int_0^t e),\ [-90°, +90°] \tag{23}$$
$$\psi_{ref} = \alpha_k + \chi_r, \tag{24}$$

where $\psi_{ref} \in [0°, 360°)$ is the desired heading, $K_p > 0$ the proportional tuning parameter and $K_i > 0$ the integral tuning parameter. $sat(a, b)$ returns a saturated to the range b. The tuning parameters are set by the user on the user interface at runtime.

Unfortunately, this mode of operation still remains untested.

5. GUIDANCE

Previously, it is established low level controllers that either ensures progress towards a feasible waypoint or ensures progress along a line defined by two waypoints. Hence, the main task of the guidance part of the system, is to pass feasible waypoints down to these controllers in order to conduct the mission. In this case, the mission consists of sailing through a set of waypoints or follow a certain path. The next two sections describes how the two mission types are addressed.

5.1 Waypoint tracking

When the mission is to sail through a set of waypoints, the *Waypoint tracking* mode is applied. The definition of sailing through a waypoint is to bring the boat within a certain radius $r_{waypoint} > 0$ of the waypoint. It is assumed that the predefined set of waypoints are feasible and that the boat is able to sail through the set. The following algorithm is used to achieve waypoint tracking:

Algorithm 1 Waypoint tracking

$P \leftarrow P_1, P_2, ..., P_N$
$i \leftarrow 0$
repeat
  $P_{Ref.Waypoint} \leftarrow P_i$
  if $|P^n_{boat} - P^n_i| < r_{waypoint}$ **then**
    $i \leftarrow i + 1$
  end if
until $i == N$

where P is the set of waypoints and $P_{Ref.Waypoint}$ is the reference waypoint from Section 4.2.3.

If a waypoint is infeasible, or becomes infeasible due to high ocean currents or high side-slip, the boat fails to complete the route. It is trivial to determine if that is the case and the guidance system should attempt to generate a new path.

5.2 Path following

When the mission is to follow a path, the *Path following* mode is used. A similar algorithm to that of *Waypoint tracking* is used for this purpose:

Algorithm 2 Path following

$P \leftarrow P_1, P_2, ..., P_N$
$i \leftarrow 0$
repeat
  $P_k \leftarrow P_i$
  $P_{k+1} \leftarrow P_{i+1}$
  if $s > |P^n_{boat} - P^n_i|$ **then**
    $i \leftarrow i + 1$
  end if
until $i == N$

where P consists of waypoints that makes up the path from the lines defined by coincident waypoints. P_{k+1} and P_k define the current line as in Section 4.2.4, while s is the distance travelled along the current path.

As with waypoint tracking, the guidance system must generate a new path if the previous becomes infeasible.

5.3 Path generation

From a practical perspective, any path or waypoint route may be divided into a finite set of lines. For curved paths, there is a trade off between the representation accuracy and number of line segments. Hence, using the previous control algorithms, the boat may follow any feasible path.

Triangle test path During field tests, a frequently used test path was the *triangle test path*. This is a generated path shaped as an equilateral triangle. When the triangle path is rotated such that one side is parallel to the wind, all sides of the triangle are feasible lines with some margin. Hence, the boat may sail along the triangle.

From the user interface, the user can adjust the center position T_c, the triangle radius T_r, the triangle heading T_ψ, and if applicable; the waypoint radius $r_{waypoint}$.

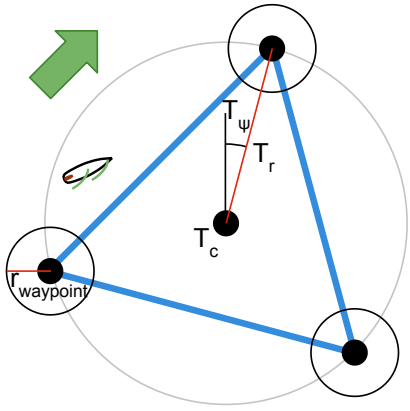

Fig. 10. Equilateral triangle test path.

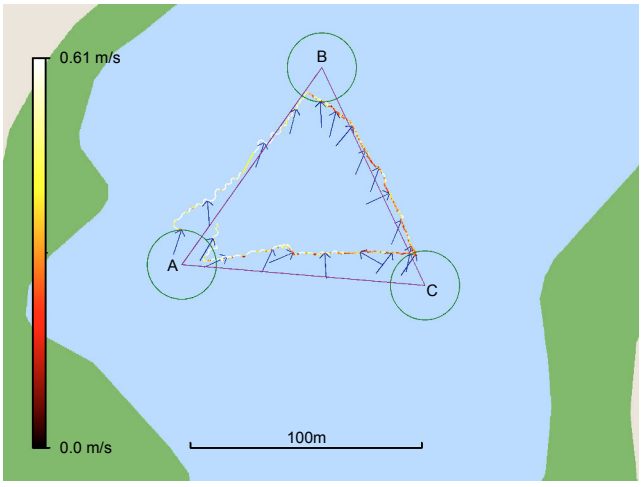

Fig. 11. *Waypoint tracking* around a triangle test path. Arrows indicate instantaneous wind direction measurement at the point the arrow is pointing at.

6. EXPERIMENTAL RESULTS

Through field tests conducted at Kyvannet in Trondheim during April, May and June 2015, results and experience has been gained. In particular:

- The boat has proved good sailing capabilities and satisfactory operation of sail trim and rudder.
- The embedded computerized control system, data link and user interface has worked in a stable and satisfactory fashion.
  - The user is able to remote control the boat and the controls feel responsive.
  - The data logger has stored timestamped information on sensor readouts, filtered values, user input and information about the inner workings of the system itself. Hence, system behaviour and performance can be studied post testing.
  - Allowed for calibration routines and live tuning of the system to be conducted.
  - Higher level control algorithms have been running successfully allowing for autonomous operation of the boat.
- The boat has sailed autonomously through a predefined feasible path using the suggested control alloca-

tion scheme together with navigation measurements, heading control and waypoint tracking.

- Pure accelerometer estimates for roll and pitch has been used to compensate for attitude in the calibrated magnetometer reading in order to obtain the heading. The filtered heading has proved sufficient for heading hold control.
- The control allocation scheme has proved satisfactory for semi-automated operation as well as full autonomous operation in all points of sail.
- The waypoint tracking scheme has worked satisfactory for feasible paths.

Fig. 12. The same *Waypoint tracking* as in Figure 11, but with an additional round. At the second round the wind stopped for a moment resulting in a crocked path along one of the sides. Imagery and mapping by Google-Earth^{TM}

Figure 11 and Figure 12 show the boat path when sailing autonomously along a triangle test path using the waypoint tracking scheme.

Fig. 13. The boat sailing in close reach, seen from upwind.

Figure 14 and 13 show the boat as it sails around manually.

Figure 15 shows the boat sailng in *heading hold* mode, accompanied by Figure 16 which shows the heading reference and measurements.

[1] Notice that heading values are limited to the range [0 °,360 °). In this figure the heading values have been allowed to go outside this range for illustrative purposes. For values outside the range, true values are obtained by subtracting or adding a complete round.

Fig. 14. The boat sailing in close reach, seen from downwind.

Fig. 15. The boat in "heading hold mode". The main sail angle is clearly released outwards by the controller in order to make the boat turn right away from the wind.

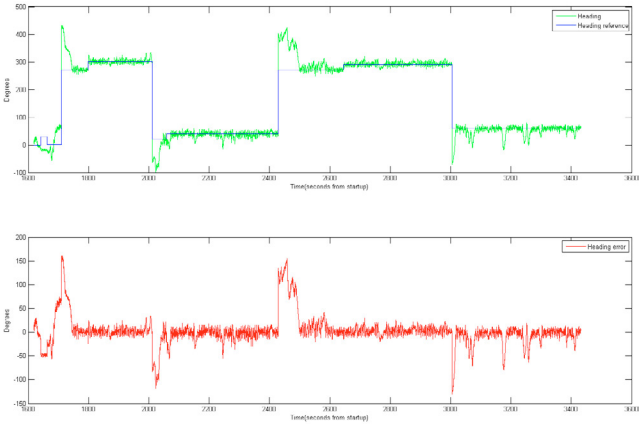

Fig. 16. Heading reference, measured heading and heading error during "heading hold" based on magnetometer readings [1] .

7. CONCLUSION

A small scale sail boat has been designed, built and automated. It is capable of sailing a predefined feasible track with no user interaction, using only the wind as propulsion.

The boat has been equipped with flexible and powerful hardware for computerized control. This hardware gathers measurements, runs actuators and execute both high and low level control algorithms in a structured and expandable environment. Additionally, a smart phone user

interface was made that proved to be a powerful tool in field tests.

A control allocation scheme for the sail boat has been proposed. A wind direction sensor was made and its measurements are used to control the lift in the sails. This scheme has proved efficient in tests.

The control allocation scheme has been used to achieve a semi-manual mode which lets the user sail the boat in all feasible directions without having to keep sail trim in mind. A controller uses the control allocation scheme to close the heading loop. Another controller are used to close the position loop by a waypoint tracker.

Further possible work include:

- It is believed that a little more tuning and small refinements quickly can further improve the boat's performance. This can be done by conducting more field tests and does not require any further implementation.
- After succeeding with waypoint tracking, there was no time left to verify the path following controller. This should be a first choice for further work. It is expected that this should work out well and that only minor changes may be necessary.
- Work on making a complete and optimal path planner for sail boats have already been started and shows promising results, see Figure 17. This path planner takes the constraints of sailing and the geographical constraints into account. The path planner algorithm also accommodates ocean currents and moving obstacles. However some work remain before this path planner can be integrated into the rest of the system. At that point, the user can click a selected location on the user interface map, and the boat will autonomously sail to that location.

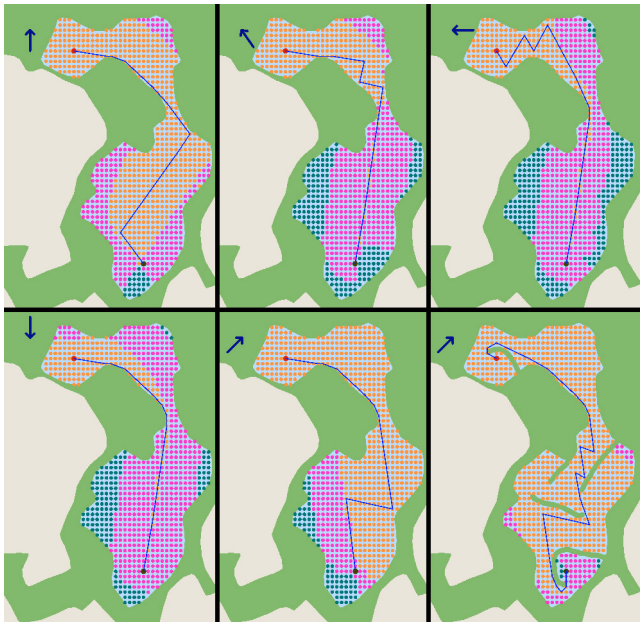

Fig. 17. Results from the path planner mentioned. The arrow indicates wind direction. Red dot indicates the start point, while dark blue indicates the finish point.

8. ACKNOWLEDGEMENTS

I would like to thank the Norwegian University of Science and Technology for 5 good years as a student.

I would also like to thank Morten Breivik for supervising my master thesis and for nominating it for *thesis of the year 2015* by *NFA-Norsk Forening for Automasjon*.

REFERENCES

Abril J., Salom J., S.O. (1997). Fuzzy control of a sailboat. *International Journal of Approximate Reasoning*.

Alexander Schlaefer, O.B. (ed.) (2011). *Proceedings of the 4th International Robotic Sailing Conference*. Springer.

Atmel (2016). Atmega2560. `http://www.atmel.com/devices/atmega2560.aspx`. Accessed: 03-04-2016.

Breivik, M. and Fossen, T.I. (2009). Guidance laws for autonomous underwater vehicles. *InTech*.

CAN bus (2016). Controller area network (can) overview. `http://www.ni.com/white-paper/2732/en/`. Accessed: 03-04-2016.

Fabrice Le Bars, L.J. (ed.) (2013). *Proceedings of the 6th International Robotic Sailing Conference*. Springer.

Finnis, C.S.J. (ed.) (2012). *Proceedings of the 5th International Robotic Sailing Conference*. Springer.

Fossen, T.I. (2011). *Handbook of marine craft hydrodynamics and motion control*. Wiley.

FreeRTOS (2016). Freertos home page. `http://www.freertos.org/`. Accessed: 03-04-2016.

Microchip (2016). Stand-alone can controller with spi interface: Mcp2515. `www.microchip.com/MCP2515`. Accessed: 03-04-2016.

Raspberrypi (2016). Raspberrypi home page. `https://www.raspberrypi.org/`. Accessed: 03-04-2016.

Stelzer, R. and Jafarmadar., K. (2011). History and recent developments in robotic sailing. *Proceedings of the 4th International Robotic Sailing Conference*.

Stelzer R., Prll T., J.R.I. (2007). Fuzzy logic control system for autonomous sailboats. *IEEE International Conference on Fuzzy Systems,*.

Stenersen, H. (2015). *Construction and Control of an Autonomous Sail Boat*. Master's thesis, Norwegian Univercity of Science and Technology.

World Robotic Sailing Championship 2014 (2014). `http://wrsc2014.com/`. Accessed: 14-12-2014.

Y. Briere, F.L. Cardoso Ribeiro, M.V.R. (2009). Design methodologies for the control of an unmanned sailing robot. *IFAC*.