Safe Navigation for Autonomous Sailboats under Uncertain Wind Conditions

Ben Wolsieffer

June 9, 2020

1 Introduction

This paper develops a method of path planning for unmanned autonomous sailboats under shifting and uncertain wind conditions. It consists of two main components: a method of estimating uncertainty in the wind, and a path planner that can take this uncertainty information into account and respect a sailboat's motion constraints.

Unmanned autonomous sailboats can be self-sufficient, operating solely on wind and solar power for long periods, while also remaining mobile. This makes them an ideal alternative to stationary buoys for data collection. For example, unmanned sailboats can use a wind lidar to measure turbulence caused by turbines within an offshore wind farm. This information could increase the efficiency of wind farms by allowing more optimal placement of the turbines. Currently, these measurements are performed by meteorological masts or lidars on fixed platforms or buoys, all of which are difficult to move to survey a new location [1]. Operating within wind farms requires the ability to avoid collisions with wind turbines. Obstacle avoidance for propulsion powered unmanned surface vehicles has been studied by many researchers [2–6], but sailboats have restricted maneuverability that requires a different approach.

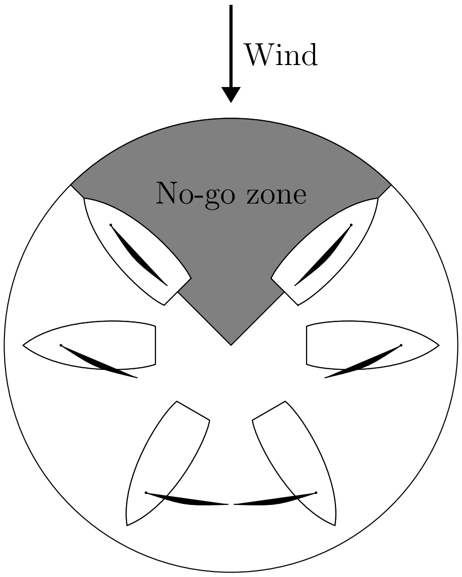


Figure 1: Diagram of no-go zone and sail trim at various headings relative to the wind. The black line on the boat represents the sail position.

Implementing obstacle avoidance for sailboats is more difficult than for propulsion powered vehicles, primarily because sailboats do not have the freedom to move in any direction

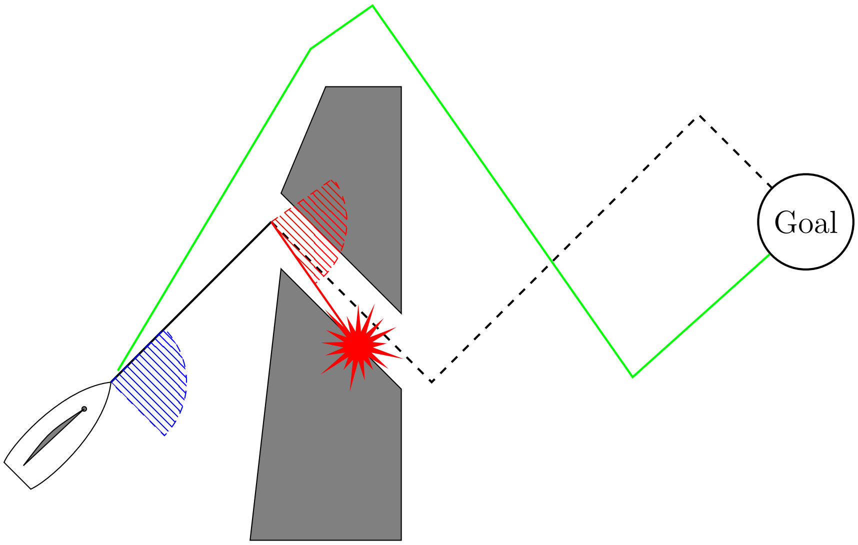


Figure 2: Overview of the problem. A path planner that only takes into account the current wind could lead the boat into a environment where slight shifts in wind could force the boat into a collision (red). Anticipating these shifts allows the planner to pick a path (green) that is more likely to be safe in the face of uncertain wind.

at a particular time. The motion of a sailboat is limited by the current direction of the wind. The speed of a sailboat depends primarily on the boat's heading relative to the apparent wind. The apparent wind is the wind observed on the moving boat, and is equal to the true wind vector minus the boat's velocity. Sailboats must not enter the *no-go zone*, the range of headings around the apparent wind direction where the sail loses lift and the boat's Velocity Made Good (VMG) begins to decrease. VMG is the rate at which the boat approaches a target point, and the boat's heading and sail trim must be continually adjusted to maximize VMG. On most sailboats, the no-go zone is around 45 degrees to either side of the wind [7, p. 72]. A diagram showing the no-go zone and various sail trim examples can be seen in Figure 1. If the target point is inside the no go zone, the boat cannot travel in a straight line toward the target. Instead, the boat travels on a heading that maximizes VMG, normally right on the edge of the no-go zone. After some time, the boat may need to tack, which refers to the act of crossing the no-go zone, either because the VMG of the new tack is significantly better or there is an obstacle along the current heading. The boat's momentum carries it through the no-go zone, but this maneuver temporarily reduces the boat's speed, making it advantageous to reduce the number of tacks performed while traveling toward a goal.

This paper proposes a path planner that takes into account the constraints the no-go zone places on the sailboat's motion. To produce the best possible plan, the planner must know the apparent wind direction at each point along the path. This is impossible, because the wind, while having a prevailing direction, shifts slightly over time. However, it is possible to estimate the probability that the wind will blow in a particular direction at a particular time. This prediction can then be used to guarantee with a certain probability that the boat will be outside the no-go zone at all points along the planned path. Figure 2 illustrates an example of how incorporating wind uncertainty can allow the boat to avoid a collision.

2 Related Work

The research performed for this work can be divided into two main topic areas: estimating the wind and its uncertainty, and planning a path based on this information. Wind estimation draws on existing work in the fields of signal processing and statistics. While path planning for sailboats is not fully studied, there is extensive literature in robotics for many different systems and environments.

2.1 Forecasting

A number of techniques have been used to develop models for stochastic processes. Kalman filters are commonly used for state and uncertainty estimation of linear systems perturbed by Gaussian noise [8]. A standard Kalman filter requires manual selection of the measurement and process covariance matrices through a characterization process, and the resulting state covariance matrix depends on the selection of these parameters, not the measurements. In this application to wind uncertainty estimation, process covariance depends on the wind conditions, and therefore has to be determined online from the wind measurements. Adaptive Kalman filter methods have been proposed that use the filter innovation sequence to calculate both the measurement and process covariance matrices [9, 10].

Autoregressive Moving-Average (ARMA) models are commonly used to generate forecasts in economics and other fields. ARMA model parameters are normally selected by performing maximum likelihood estimation on historical data. This approach is ill-suited to online adaptation, but some papers have described methods for online ARMA parameter fitting [11, 12]. For this wind uncertainty modeling application, it is necessary to determine the confidence interval for the wind state. Given a training dataset, the confidence interval can be calculated from the variance of the model residuals, but the online optimization methods cited previously do not provide this capability.

Generalized Autoregressive Conditional Heteroskedastic (GARCH) models have been proposed as a complement to ARMA to model heteroskedastic processes [13]. GARCH models use ARMA to predict the residuals of a separate mean model (often also ARMA). Applying a GARCH model to the wind would allow the estimator to adapt to changing wind conditions, particularly changes in volatility, without being fitted offline. The online optimization methods discussed earlier could also be applied to the GARCH model, enabling all aspects of the estimator to be adapted online.

One of the key differences between a Kalman filter and an ARMA approach is the asymptotic behavior of the predicted variance. The Kalman filter predicts a variance that grows linearly and without bound as time increases, while an ARMA model predicts a variance that approaches the variance of the underlying process (under the assumption that the process is stationary). An adaptive Kalman filter method was chosen to estimate the wind uncertainty in this project and may be more computationally efficient than an ARMA-based method, but an ARMA-based method could be explored in future work.

2.2 Path Planning

Various projects have tackled the problem of obstacle avoidance and navigation for autonomous sailboats or other types of marine vehicles. Given the unpredictability of the aquatic environment, many approaches do not incorporate any forward planning and instead rely on reactive local control algorithms that allow the boat to make progress toward a target while staying away from obstacles. One such technique is to have the boat follow the gradient of a potential field that repels the boat from obstacles and the no-go zone and attracts the boat toward the goal [14,15]. This technique is limited by the possibility of the boat getting stuck in a local minimum of the potential field.

An even simpler technique chooses the optimal heading based on a cost function. The cost of each heading is determined by a balance between the Velocity Made Good and the distance to the nearest obstacle. In the case where there are two optimal headings (e.g., port and starboard tacks), a cost hysteresis can be used to prevent oscillation [16], or the opposite tack can be penalized in the cost function [17]. In general, reactive techniques are limited by their inability to plan ahead and optimize the entire path. To keep the boat safe, they often need to be conservative.

Path planning algorithms have seen limited use with autonomous sailboats. This may be due to the difficulty of predicting wind and other environmental conditions. Siegwart et al. use an A* planner with a sailboat specific cost function [18]. The wind is assumed to remain constant throughout the path, which is likely an unrealistic assumption. This is worked around by regenerating the path if the conditions change significantly. Saoud et al. use a Probabilistic Roadmap (PRM) and Dijkstra's algorithm [19]. Wind information is provided by an external forecast. The wind forecast is coarse, but this is acceptable because the paths spanned distances on the order of 1000 km. Local control was performed using a reactive potential field controller.

Other work has used partially observable Markov decision processes (POMDP) to solve robotic planning problems under uncertainty. For example, Ragi and Chong use a POMDP to perform collision avoidance and target tracking for unmanned aerial vehicles (UAVs) while accounting for wind disturbances [20]. Rajendran et al. apply POMDPs to a unmanned surface vehicle planning problem in the presence of other boats and their wakes [21]. Markov decision processes require discretizing the state space and may be computationally intensive for large state spaces.

There are many sources of uncertainty while sailing, but no existing work incorporates an uncertainty model into control or planning for sailboats. There exist algorithms for planning under uncertainty that have been applied to other robot planning problems. Missiuro and Roy [22] extend PRM [23] to model uncertain obstacles, while Luders et al. developed chance constrained rapidly exploring random trees (CC-RRT) [24] and CC-RRT* [25,26], which are extensions of the well known RRT [27] and RRT* [28] algorithms respectively. These algorithms can incorporate uncertainty in localization, motion and obstacle positions and plan paths that trade off between risk of collision and path efficiency, maintaining a desired probability of success.

3 Methods

The proposed method uses a wind uncertainty estimation algorithm to produce a dynamically tightened no-go zone constraint in the path planner.

3.1 Wind Uncertainty Estimation

Incorporating wind uncertainty into the planning problem first requires a way to estimate wind characteristics. It is assumed that the boat carries sensors that can measure wind speed and direction. All models in this paper were developed and tested using a cup anemometer and wind vane. These sensors have mechanical properties that may make them behave differently from other types of sensors such as ultrasonic anemometers. This work assumes the wind speed and direction follow a Gaussian distribution with constant mean and variance over short time scales. The parameters of this distribution are estimated using an innovation-based adaptive Kalman filter, based on [10]. The Kalman filter estimates the wind state, which is a two element vector defined as follows:

$$\hat{\boldsymbol{w}}_t = \begin{pmatrix} \hat{s}_t \\ \hat{\phi}_t \end{pmatrix} \tag{1}$$

where $\hat{s}_t \in \mathbb{R}_+$ is the wind speed in meters per second and $\hat{\phi}_t \in (-\pi, \pi]$ is the wind direction in radians. The measurement covariance matrix R is assumed constant and determined through characterization of the anemometer, while the process covariance matrix Q_t is adapted using a moving innovation window. The wind speed and direction are assumed to be Gaussian random variables with constant mean and variance over the estimation window. This means the expected value of the next state is the same as the current state, making the state update matrix A an identity matrix. A more complex model may be derived in the future through analysis of the wind's behavior, or based on external forecasts, but lacking such information, the only assumption that can be made is that the expected wind does not change between time steps. The measurement matrix H is also an identity matrix because the wind sensor directly measures the state variables. The filter also has no inputs. This results in the following simplified filter equations:

$$\begin{align} P_t^- &= P_{t-1} + Q_t \tag{2} \\ K_t &= P_t^-(P_t^- + R)^{-1} \tag{3} \\ \boldsymbol{v}_t &= \boldsymbol{z}_t - \hat{\boldsymbol{w}}_{t-1} \tag{4} \\ \hat{\boldsymbol{w}}_t &= \hat{\boldsymbol{w}}_{t-1} + K_t \boldsymbol{v}_t \tag{5} \\ P_t &= (I - K_t)P_t^- \tag{6} \end{align}$$

At each time step t, the filter first uses (2) to predict the new state covariance P_t^- from the covariance P_{t-1} at the previous timestep. The predicted covariance is used to calculate the Kalman gain K_t in (3). The innovation $\boldsymbol{v}_t$ is calculated in (4) as the difference between the real wind measurement $\boldsymbol{z}_t$ and the predicted measurement $\hat{\boldsymbol{w}}_{t-1}$, which in this case is the wind state from the previous step. The Kalman gain and innovation are used to update the state in (5), and the Kalman gain is again used in (6) to produce a corrected covariance P_t from the predicted covariance.

The process covariance matrix Q is then adapted as follows:

$$Q_t = K_t C_{v_t} K_t^T \tag{7}$$

where C_{v_t} is calculated using a moving window of the past N innovations:

$$C_{v_t} = \frac{1}{N} \sum_{j=j_0}^{t} \boldsymbol{v}_j \boldsymbol{v}_j^T \tag{8}$$

where $j_0 = t - N + 1$ is the first index inside the moving estimation window. The window size must be large enough to capture the full variation of the wind. Larger windows make the estimator respond more slowly to changes in the wind's variance, and increase memory usage.

Measurements from the anemometer are used to update the conditional state and covariance of the filter, producing a constantly updated estimate of the wind conditions. During planning, the filter uses open loop forward propagation to predict the wind conditions at each point along the planned path. This causes the predicted covariance to grow linearly with time.

3.2 Path Planning

Path planning is performed using a variant of the Rapidly-Exploring Random Trees (RRT) algorithm, which incorporates chance constraints to find a probabilistically safe path given uncertain wind conditions. The planner attempts to find a sequence of feasible states in configuration space that allows the boat to reach the goal state. In this case, the boat uses the SE(2) configuration space, which consists of position in 2D space and heading. The boat's state is constrained by obstacles such as land masses or islands, as well as the no-go zone, which acts like an obstacle in the heading dimension. The location of the no-go zone depends on the wind direction; therefore, any uncertainty in the wind results in uncertainty in the edges of the no-go zone. This uncertainty motivates the extension of RRT to include probabilistic constraints. Other obstacles are assumed to be static with known positions.

The traditional RRT algorithm builds a tree of feasible states, starting from a root at the robot's current state [27]. A state is sampled randomly from configuration space and the nearest existing state in the tree is found. A steering law is used to find a sequence of controls that steers the robot from the nearest state toward the new state, while respecting differential constraints on the robot's motion. If an infeasible state is found along the path, the last valid state is added to the tree instead of the randomly sampled state. This process repeats until the goal state is reached. RRT makes no attempt to minimize the cost of the path, and therefore often results in paths of low quality.

RRT* extends RRT to include a cost function, which allows the path to be optimized [28]. When a new state is sampled, an attempt is made to form a connection from each of the k-nearest neighbor states. The connection that results in the lowest cost is kept. Then, the rewiring step replaces each of the neighbors' parents with the new state if that would reduce their cost.

CC-RRT [24] and CC-RRT* [25] extend RRT and RRT*, respectively, to include uncertainty in the robot's state as well in the obstacle locations. These uncertainties are represented using Gaussian distributions, and obstacle boundaries are dynamically expanded to make sure all paths stay below a certain threshold of probability of collision. CC-RRT* adds a cost function that minimizes the risk of the path.

This work uses a simplified model of sailboat dynamics and control. The boat travels in a straight line between points along a desired heading. It is assumed that the boat can turn in place to reach a new heading. This is a reasonable assumption because the distance required to turn is small compared to the normal length of straight line paths between turns.

The current project ignores any uncertainty in the boat's state and the position of cartesian obstacles. The boat's state is assumed to be known with low, nearly constant uncertainty, through the use of GPS and an inertial measurement unit (IMU). Obstacle positions are assumed to be accurately specified by nautical charts or other sources. On the other hand, the no-go zone has greater uncertainty, and the uncertainty varies depending on wind conditions.

The proposed planner does not implement the full chance constraint model or risk based cost function used in CC-RRT*. CC-RRT* places a bound on the estimate of the total risk along the path, which is neglected in this work. This work only places a bound on the maximum risk along the path, which Luders et al. [25] state usually results in less conservative plans. CC-RRT* also includes a cost that minimizes accumulated risk along the path as well as the maximum cost of any state in the path. These cost terms are not used in the proposed planner.

At a particular time step t, the wind direction is defined as follows:

$$\phi_t \sim \mathcal{N}\left(\hat{\phi}_t, P_{\phi_t}\right) \tag{9}$$

where $\mathcal{N}(\hat{a}, P_a)$ denotes a Gaussian random variable with mean $\hat{a}$ and covariance P_a. The mean ϕ_t and variance P_{ϕ_t} are provided by the adaptive Kalman filter tailored for this application, described in Section 3.1.

Let $\boldsymbol{x}_t$ be the state vector at time t. This vector consists of three components:

$$\boldsymbol{x}_t = \begin{pmatrix} x_{xt} \\ x_{yt} \\ x_{\theta t} \end{pmatrix} \tag{10}$$

where $p_x \in \mathbb{R}$ and $p_y \in \mathbb{R}$ correspond to the boat's Cartesian coordinates, while $\theta \in (-\pi, \pi]$ is the boat's heading in radians. $\boldsymbol{x}_t$ is assumed to be perfectly known, but subject to a set of time-varying, probabilistic constraints $\mathcal{X}_t \subset \mathbb{R}^3$ due to the no-go zone. These constraints are defined as follows:

$$\mathcal{X}_t = \mathcal{X}^0 + \boldsymbol{c}_t, \tag{11}$$

$$\boldsymbol{c}_t \sim \mathcal{N}\left(\hat{\boldsymbol{c}}_t, P_{\boldsymbol{c}_t}\right) \tag{12}$$

where the + operator denotes set translation. $\mathcal{X}^0 \subset \mathbb{R}^3$ is a convex polyhedron representing the shape of the no-go zone, while $\boldsymbol{c}_t \in \mathbb{R}^3$ represents its time-varying and uncertain translation.

One of the goals of the path planner is to ensure that the following constraint is satisfied:

$$\mathbb{P}\left(\boldsymbol{x}_t \in \mathcal{X}_t\right) \geq \delta_s, \ \forall t \in \mathbb{Z}_{0,t_f} \tag{13}$$

where $\delta_s \in [0.5, 1]$ is the desired probability that the boat will not enter the no-go zone and t_f is the final time step of the path.

The no-go zone can be represented as a conjunction of two linear inequalities, one for each edge:

$$\bigwedge_{i=1}^{2} \boldsymbol{a}_i^T \left(\boldsymbol{x}_t - \boldsymbol{c}_{it}\right) < 0, \ \forall t \in \mathbb{Z}_{0,t_f} \tag{14}$$

where $\boldsymbol{c}_{it}$ is a point nominally on the edge of the ith constraint, assuming no uncertainty. The specific values of $\boldsymbol{a}_i$ and $\boldsymbol{c}_{it}$ that define the no-go zone constraint are as follows:

$$\boldsymbol{a}_1 = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}, \boldsymbol{c}_{1t} = \begin{pmatrix} 0 \\ 0 \\ \hat{\phi}_t + \theta_{ng} \end{pmatrix}$$

$$\boldsymbol{a}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \boldsymbol{c}_{2t} = \begin{pmatrix} 0 \\ 0 \\ \hat{\phi}_t - \theta_{ng} \end{pmatrix}$$

where $\hat{\phi}_t$ is the expected wind direction in radians.

As shown by Luders et al. [25], the no-go zone constraints are satisfied with probability δ_s at a particular time step t if the following modified constraints are satisfied:

$$\bigvee_{i=1}^{2} \boldsymbol{a}_i^T \left(\boldsymbol{x}_t - \boldsymbol{c}_{it}\right) \geq \bar{b}_{it} \equiv \sqrt{2} P_v \operatorname{erf}^{-1}\left(1 - 2\delta_s\right), \tag{15}$$

$$P_v = \sqrt{\boldsymbol{a}_i^T \cdot P_{\boldsymbol{c}_t} \cdot \boldsymbol{a}_i} \tag{16}$$

where $\bar{b}_{it}$ is the amount of tightening (in radians) that will be applied to a particular side of the no-go zone and $\operatorname{erf}^{-1}()$ is the inverse error function. Because $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ are additive inverses, the same constraint tightening is applied to each side of the no-go zone, so this tightening can simply be referred to as $\bar{b}_t$.

Algorithm 1 Sailboat steering function

```
    **Input**: $\boldsymbol{x}$ current state, $\boldsymbol{y}$ target state
    **Output**: $\bar{\sigma}$ Sequence of states that approach $\boldsymbol{y}$
 1: $\Delta_x \leftarrow y_x - x_x$
 2: $\Delta_y \leftarrow y_y - x_y$
 3: $\theta_y \leftarrow \text{atan2}(\Delta_y, \Delta_x)$
 4: **if** $|\theta_y - \phi_t| >= \theta_{ng} + \bar{b}_t$ **then**
 5:     $u \leftarrow \theta_y$
 6: **else**
 7:     $u_s \leftarrow \hat{\phi}_t + \theta_{ng} + \bar{b}_t$        ▷ Starboard tack no-go zone heading
 8:     $u_p \leftarrow \hat{\phi}_t - \theta_{ng} - \bar{b}_t$        ▷ Port tack no-go zone heading
 9:     $\gamma_s \leftarrow \cos(u_s - \theta_y)$
10:     $\gamma_p \leftarrow \cos(u_p - \theta_y)$
11:     **if** $\gamma_s \geq \gamma_p$ **and** $\gamma_s > 0$ **then**
12:         $u \leftarrow u_s$
13:     **else if** $\gamma_p \geq \gamma_s$ **and** $\gamma_p > 0$ **then**
14:         $u \leftarrow u_p$
15:     **else**
16:         Can't make any progress toward $\boldsymbol{y}$
17: **while** distance to $\boldsymbol{y}$ decreases **do**
18:     Propagate $\boldsymbol{x}$ in heading $u$
19:     Append propagated state to $\bar{\sigma}$
```

To determine the control required to connect a pair of states, the planner uses a steering function, described in Algorithm 1. In the downwind case, the steering function simply calculates the heading required to point the boat at the target state (line 5). If the target state is located in the no-go zone, the steering function points the boat toward the tack with the highest VMG (line 11), and continues along that tack until it is no longer making progress toward the goal (line 17) (i.e., the heading toward the target is perpendicular to the boat's heading). Propagation (line 18) is performed assuming a constant forward speed v, ignoring any dependence on the wind for simplicity. At each time step, the boat's position is advanced along the commanded heading u by a fixed amount.

The planner cost function is described in Algorithm 2. It comprises a cost for distance traveled and a cost for turning. The costs are calculated in terms of the time required to complete the path, using estimates for the boat's forward speed v in meters per second and turn rate ω in radians per second. Again, these speeds are assumed to be constant and independent of the wind speed and direction for simplicity, but future work will look at optimizing the path by calculating the boat's speed based on the wind conditions.

Algorithm 2 Sailboat cost function

```
**Input**: $\bar{\sigma}_t$ Sequence of states in path from time step 0 to $t$
**Output**: $f$ Path cost
**for** $i \in [1, t]$ **do**
    $f \leftarrow f + \sqrt{(\bar{\sigma}_{i,x} - \bar{\sigma}_{i-1,x})^2 + (\bar{\sigma}_{i,y} - \bar{\sigma}_{i-1,y})^2}/v + |\bar{\sigma}_{i,\theta} - \bar{\sigma}_{i-1,\theta}|/\omega$
```

4 Experiments

Experiments were conducted to validate the effectiveness of the proposed method both as a basic path planner for sailboats, as well as to assess whether incorporating wind uncertainty into the no-go zone improves navigation safety. The planner was tested offline, and was also integrated into a simulator.

To collect wind data and perform experiments, an off-the-shelf 1 meter Horizon Hobby Ragazza remote control sailboat [29] (Figure 3) was modified to be autonomously controlled, using a Pixhawk flight controller [30] running the open source ArduPilot firmware [31]. ArduPilot has support for basic reactive control of sailboats, as well as data logging from its sensors. The Pixhawk contains many sensors, including a gyroscope, accelerometer, and magnetometer. A GPS receiver was connected to the Pixhawk to provide position information, and a telemetry radio was used to communicate with a computer on shore. These components were mounted to the bottom of the hatch cover, as shown in Figure 3 (bottom). As this boat was only used for short duration tests, this was found to be sufficient to keep the components out of contact with water. In this paper, the boat was only used for wind data collection, but future work will perform real world tests of the proposed path planner.

The sail is controlled with a winch servo that tightens or slackens both the jib and main sheets simultaneously. The rudder is actuated with a standard hobby servo. The boat was fitted with a cup anemometer and wind vane manufactured by Davis Instruments [32], mounted on a pole extending forward in front of the jib, about 1 meter above the deck (see Figure 3 (top)). The area of the wind vane was increased to make it more responsive in low winds. These sensors have significant mechanical dynamics that affect their measurements. For example, the friction in the wind vane's bearing can cause it to stick in low winds, coupling it with the boat's motion. The anemometer's inertia causes under- and over-speeding, resulting in a measurement delay when the wind speed changes.

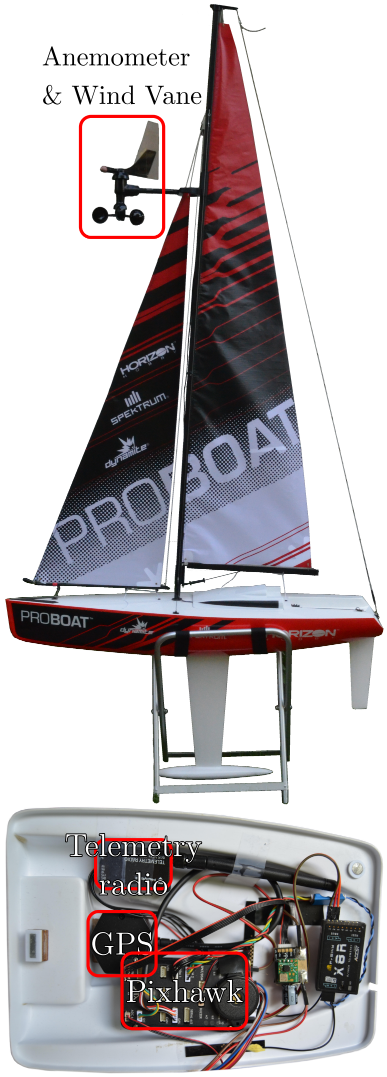


Figure 3: Autonomous sailboat modified for this project (top) and electronics mounted under hatch (bottom).

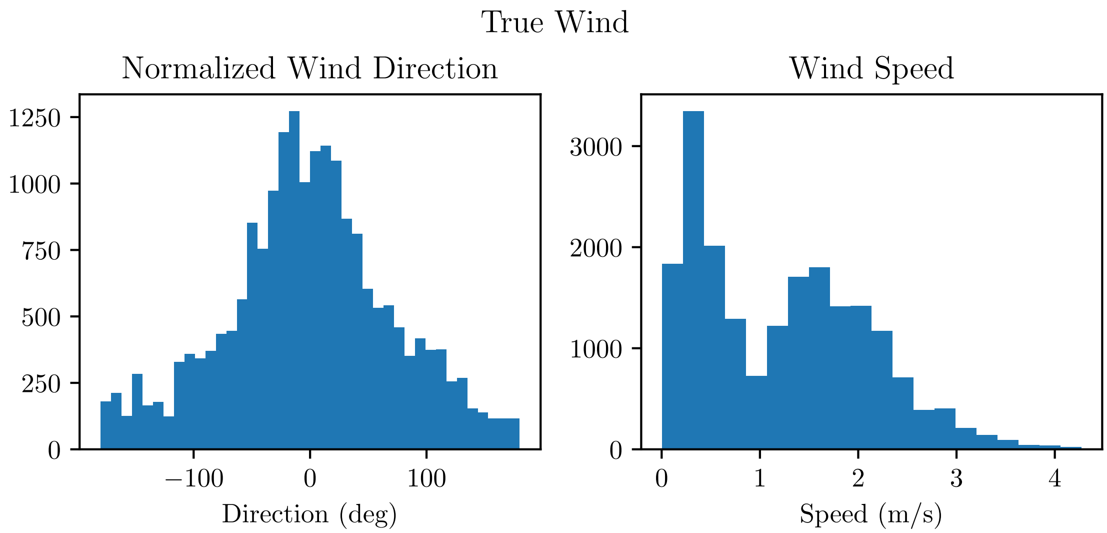


Figure 4: Histograms of wind speed and direction. The wind direction appears normally distributed, as does the wind speed except near zero.

4.1 Wind Uncertainty Estimation

The assumption of the normal distribution of the wind direction was validated using data collected from the sailboat. Figure 4 shows a histogram of wind speed and direction collected over an approximately one hour period. The wind direction is normalized around the mean for visualization purposes, and appears roughly normally distributed. The wind speed histogram only appears roughly normal at high speeds, due to limitations of the anemometer. The anemometer is unable to measure speeds below 1 m/ sec, causing the spike in measurements near zero. A low pass filter was applied to the data as it was collected, causing some readings to appear between 0 and 1 m/ sec. This has no effect on the path planner, which does not use wind speed information.

The fact that the distribution appears Gaussian does not necessarily imply that the wind is a Gaussian random variable. Samples may not be independent with respect to time, which can be seen on an autocorrelation plot, shown in Figure 5. As may be expected of a physical process, this plot indicates that each sample has some dependence on previous samples, which motivates potential future wind estimator design using models that can account for this dependence, such as ARMA.

Despite the evidence that wind direction is not a true Gaussian random variable, the Kalman filter based wind estimator is successfully able to identify changes in the variance of the wind. The adaptive Kalman filter can account for the autocorrelation by using a moving window that is larger than the time window that exhibits significant autocorrelation.

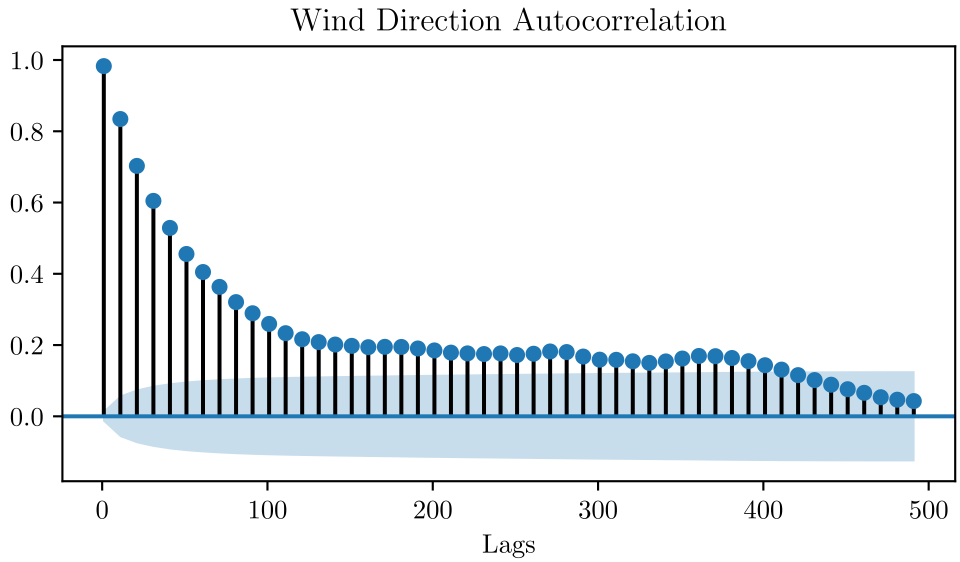


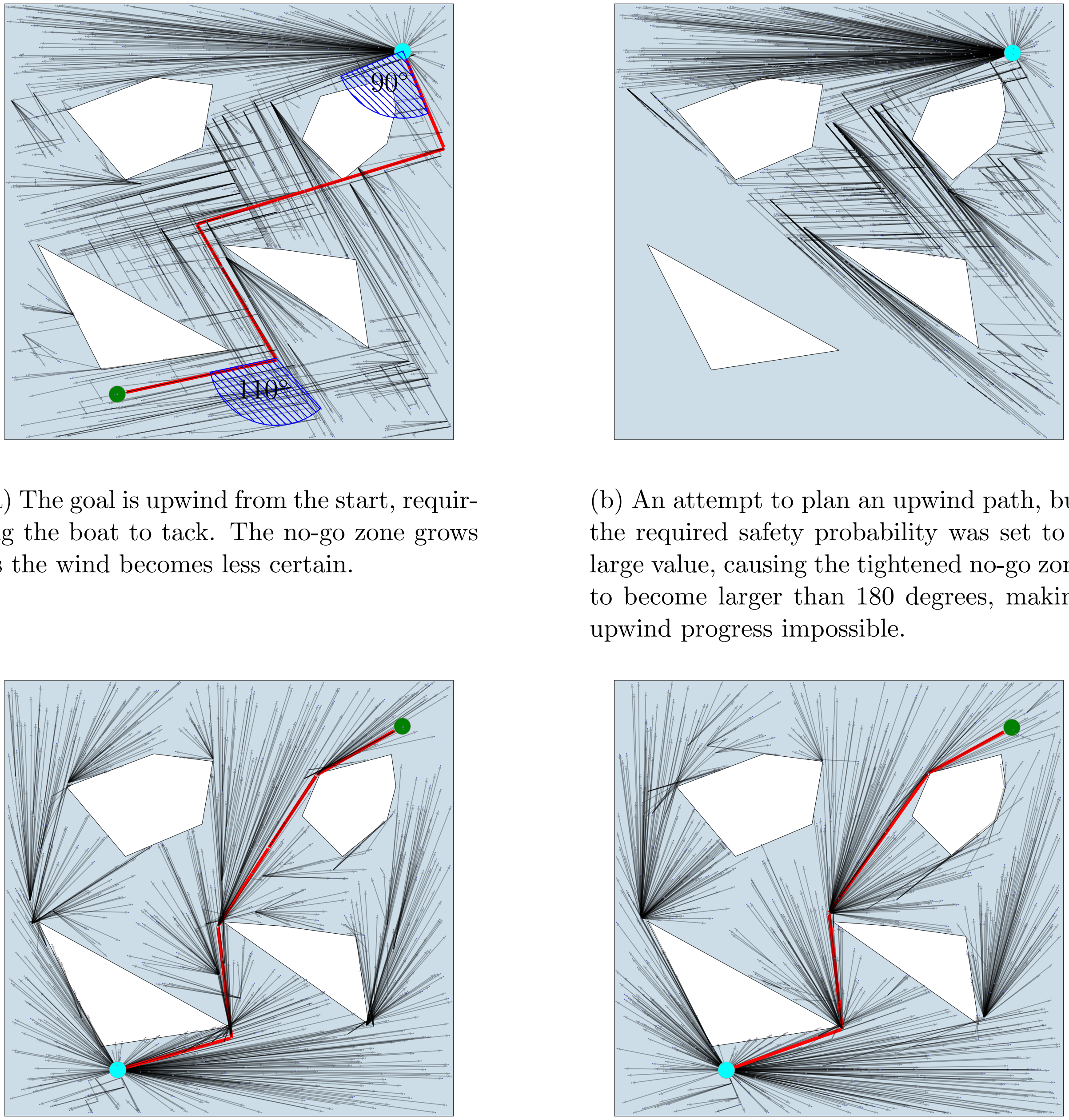
Figure 5: Autocorrelation plot of wind direction. The decaying autocorrelation indicates that samples of the wind direction are not independent.

4.2 Path Planning

The planner was implemented using the Open Motion Planning Library (OMPL) [33]. This library provided a framework for developing a sampling based motion planner.

The sailboat was used to collect samples of wind on a lake under different wind conditions. A test setup was developed that used samples of wind data collected by the boat to initialize the wind estimator, and then the planner was used offline to solve various planning problems with different parameters. Figure 6 shows example state trees that were generated by the planner operating in an environment containing four polygonal obstacles. The planner starts at the cyan vertex and attempts to find a path to the green vertex. The lowest cost path found by the proposed algorithm is highlighted in red. Figure 6a shows an upwind path, where the boat tacks a number of times to reach the goal. Initially the no-go zone is 90 degrees wide, but grows to 110 degrees near the end of the plan as the uncertainty increases. If the safety probability δ_s is increased by a relatively small amount, from 60% to 66%, the planner becomes unable to find a solution because the no-go zone grows larger than 180 degrees, as shown in Figure 6b. If the start and goal points are swapped, as shown in Figure 6c, the no-go zone constraint no longer affects the path, as the boat is travelling downwind.

The planner was then integrated with ArduPilot to test its effectiveness online in simulated or real environments. ArduPilot provides a simple sailboat simulator that has a nearly identical programming interface to a real ArduPilot based boat. Obstacle information can be loaded using a well-known text (WKT) file, a common format for representing geographic data. Geographic points are converted into a local coordinate system using a equirectangular projection centered around the boat's starting position, and all planning takes place assuming Euclidean geometry.



(a) The goal is upwind from the start, requiring the boat to tack. The no-go zone grows as the wind becomes less certain.

(b) An attempt to plan an upwind path, but the required safety probability was set to a large value, causing the tightened no-go zone to become larger than 180 degrees, making upwind progress impossible.

(c) The goal and start are reversed, resulting in a simpler downwind path.

(d) The safety probability was increased, but this had no effect on the downwind path.

Figure 6: State trees generated by the planner under different conditions. The start state is highlighted in cyan, while the goal is in green. The best path is shown in red.

The wind estimator is continually updated using data provided by the simulator or a real sensor. In the simulator, the wind direction is generated using a first order autoregressive model (AR(1)) [34, p. 53], represented as follows:

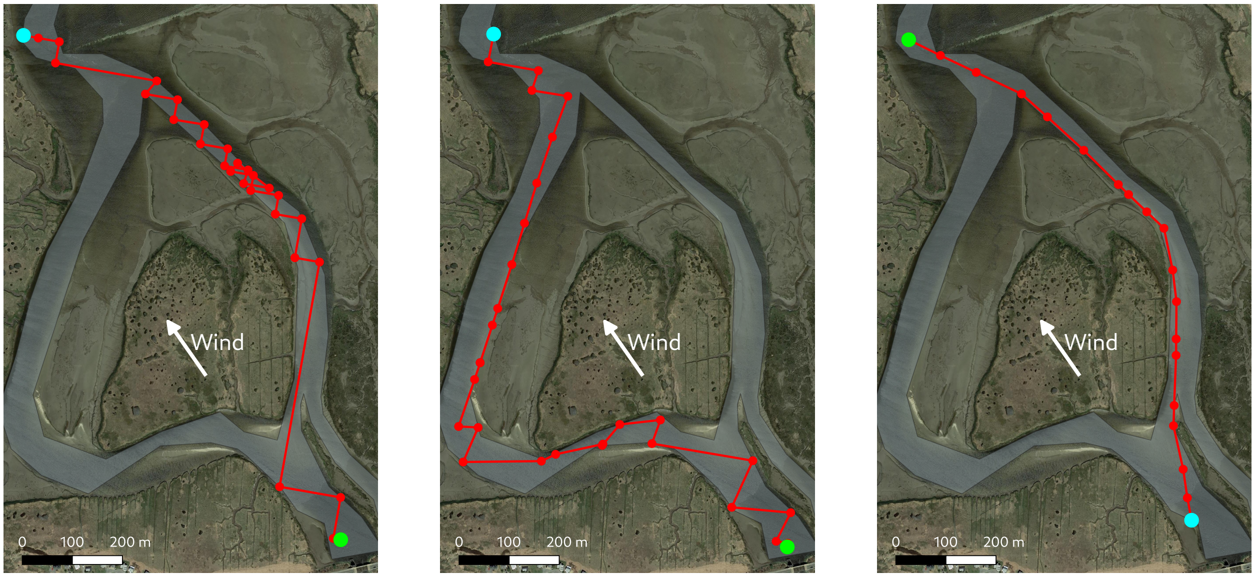
$$\begin{gathered}\hat{\phi}_t = \hat{\phi}_{t-1} + \Delta_t \left(-\rho \hat{\phi}_{t-1} + w_t \right), \qquad (17) \\ \text{s.t. } \rho = 0.036\,\text{Hz},\ w_t \sim \mathcal{N}\left(0\,\text{rad/ sec}, 0.087\,\text{rad/ sec}\right)\end{gathered}$$

where Δ_t is the simulation time step. This representation was chosen because the parameters have a physically relevant units and were independent of the chosen time step. (17) can easily be shown to be equivalent to the standard AR(1) model equation.

The AR(1) model provided a good balance between model complexity and correspondence with empirical wind data. The model parameters were fitted to a wind dataset collected by the RC sailboat by maximizing the log-likelihood using a Kalman filter [34, p. 372]

The planner runs for a fixed period of time, and then commands the boat to begin following the calculated plan. The planning time was determined experimentally, by observing how long it took before the rate of solution optimization tailed off. A planning time of 10 seconds was used for all simulator experiments. ArduPilot uses an L1 navigation controller to travel between the points on the path [35]. This controller causes the boat to travel toward a point on the line between the previous and next waypoint. The point is located a certain distance ahead of the boat, which causes the boat to move toward the next waypoint while also reducing cross-track error. Each time the boat reaches a waypoint, the path is generated again, starting from the next waypoint and incorporating new wind measurements. This compensates for the conservative nature of the planner. Even if the wind was very uncertain near the goal when the planner first ran, it will become more certain as the boat approaches the goal. Re-planning allows this increased certainty to be used to create better, less conservative paths.

Preliminary experiments have been performed using the autonomous sailboat. On the water tests were completed to tune its waypoint navigation and control algorithms, and full validation of the path planner will be part of future work.



(a) The wind is positioned such that the boat can just barely traverse the right hand path without tacking for a significant portion of the path.

(b) Increasing the wind noise tightens the no-go zone constraint, making the left hand path have a lower cost.

(c) In the downwind case, the boat plans a simple route along the right hand path.

Figure 7: Paths followed by the boat in simulation under different conditions. The start point is highlighted in cyan, while the goal is in green. As each waypoint was reached the path ahead was re-planned to take into account new wind data and reduce uncertainty. This environment offers two distinct paths to the goal.

5 Discussion

This paper successfully develops a novel sampling based planner for sailboats. This planner enables safe and efficient navigation around static obstacles while accounting for the most important constraint on sailboat motion, the no-go zone. This work also introduces a method for increasing the probability of successful path traversal by estimating the uncertainty in the wind direction and using this information to prevent the boat from becoming stuck in irons, potentially unable to maneuver away from obstacles. In realistic scenarios on open water, a planner that neglects wind uncertainty may succeed; however, there exist certain scenarios such as navigation in tight spaces or around dynamic obstacles where the proposed planner may result in significantly safer behavior. Such examples have yet to be demonstrated in simulation, but this may be due in part to the simplicity of the simulator. Analysis of more complex scenarios in simulation and the real world will be performed in future work.

Current online planning experiments use continual re-planning to produce less conservative paths over time as more wind information becomes available. Re-planning allows even a non-uncertainty aware planner to respond to changes in the wind, but is computationally expensive. Incorporating wind uncertainty may allow re-planning to occur less often, only when the wind uncertainty changes significantly, resulting in increased efficiency.

It may be possible to improve the planner's performance and stability by using a different wind estimator. The asymptotic behavior of the Kalman filter's open loop covariance prediction results in unstable planning results that depend greatly on the chosen parameters. Slight changes in the minimum allowable safety probability can mean the difference between the planner finding a solution, or failing because the tightened no-go zone grew too large. A wind estimator based on an ARMA model would certainly result in more stable results because the predicted uncertainty estimate approaches a constant as time increases, rather than increasing linearly without bound as in the case of the Kalman filter. An additional improvement in performance may be achieved in the future by implementing the full CC-RRT* algorithm, including the bound on the accumulated risk of the path and the risk based cost function.

The CC-RRT* based planner offers a useful foundation for future avenues of research, including incorporating dynamic obstacles, such as other boats. CC-RRT* may provide a framework for incorporating the uncertain motion of these obstacles into the planner, and allow the sailboat to navigate in a manner that minimizes the risk of collision.

6 Acknowledgements

Thank you to my thesis advisor, Alberto Quattrini Li, and to Mingi Jeong for their help and advice. Thank you to my parents for helping me with deploying and recovering the sailboat, and for putting up with my frustration when things didn't work.

References

[1] B. H. Bailey and D. Green, "The Need for Expanded Meteorological and Oceanographic Data to Support Resource Characterization and Design Condition Definition for Offshore Wind Power Projects in the United States," American Meteorological Society, Boston, MA, Tech. Rep., May 2013. [Online]. Available: https://www.ametsoc.org/index.cfm/cwwce/boards/board-on-enterprise-strategic-topics/offshore-wind-energy-annual-partnership-topic-committee/apt-final-report/

[2] M. R. Benjamin, M. Defilippo, P. Robinette, and M. Novitzky, "Obstacle Avoidance Using Multiobjective Optimization and a Dynamic Obstacle Manager," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 2, pp. 331–342, Apr. 2019.

[3] M. Jeong, E.-B. Lee, Y.-S. Park, and A. Quattrini Li, "A Risk Visualization Technique based on Static and Dynamic Data for Maritime Mobility," in *OCEANS 2019 MTS/IEEE SEATTLE*, Seattle, WA, Oct. 2019, pp. 1–5, iSSN: 0197-7385.

[4] T. A. Johansen, T. Perez, and A. Cristofaro, "Ship Collision Avoidance and COLREGS Compliance Using Simulation-Based Control Behavior Selection With Predictive Hazard Assessment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3407–3422, Dec. 2016.

[5] J. Park and J. Kim, "Predictive Evaluation of Ship Collision Risk Using the Concept of Probability Flow," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 4, pp. 836–845, Oct. 2017.

[6] K. Woerner, M. R. Benjamin, M. Novitzky, and J. J. Leonard, "Quantifying protocol evaluation for autonomous collision avoidance," *Autonomous Robots*, vol. 43, no. 4, pp. 967–991, Apr. 2019. [Online]. Available: https://doi.org/10.1007/s10514-018-9765-y

[7] G. Jobson, *Sailing Fundamentals.* Simon and Schuster, Sept. 2008.

[8] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960, publisher: American Society of Mechanical Engineers Digital Collection. [Online]. Available: https://asmedigitalcollection.asme.org/fluidsengineering/article/82/1/35/397706/A-New-Approach-to-Linear-Filtering-and-Prediction

[9] R. K. Mehra, "Approaches to adaptive filtering," in *1970 IEEE Symposium on Adaptive Processes (9th) Decision and Control*, Dec. 1970, pp. 141–141.

[10] A. H. Mohamed and K. P. Schwarz, "Adaptive Kalman Filtering for INS/GPS," *Journal of Geodesy*, vol. 73, no. 4, pp. 193–203, May 1999. [Online]. Available: https://doi.org/10.1007/s001900050236

[11] O. Anava, E. Hazan, S. Mannor, and O. Shamir, "Online Learning for Time Series Prediction," in *Proceedings of the 26th Annual Conference on Learning Theory*, S. Shalev-Shwartz and I. Steinwart, Eds., vol. 30. PMLR, June 2013, pp. 172–184. [Online]. Available: http://proceedings.mlr.press/v30/Anava13.html

[12] C. Liu, S. C. H. Hoi, P. Zhao, and J. Sun, "Online ARIMA algorithms for time series prediction," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. Phoenix, Arizona: AAAI Press, Feb. 2016, pp. 1867–1873.

[13] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, p. 327, 1986.

[14] S. Chanu, "Collision Avoidance for an Autonomous Sailing Boat," Åland University of Applied Science, Tech. Rep., Oct. 2016. [Online]. Available: https://www.ensta-bretagne.fr/jaulin/rapport2016_chanu.pdf

[15] C. Petres, M.-A. Romero-Ramirez, F. Plumet, and B. Alessandrini, "Modeling and reactive navigation of an autonomous sailboat," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Sept. 2011, pp. 3571–3576.

[16] R. Stelzer, K. Jafarmadar, H. Hassler, and R. Charwot, "A reactive approach to obstacle avoidance in autonomous sailing." 2010.

[17] Y. Guo, M. Romero, S.-H. Ieng, F. Plumet, R. Benosman, and B. Gas, "Reactive path planning for autonomous sailboat using an omni-directional camera for obstacle detection," in *2011 IEEE International Conference on Mechatronics*, Apr. 2011, pp. 445–450.

[18] R. Siegwart, D. C. Pradalier, M. Rufli, and G.-A. Busser, "Design and Implementation of a Navigation Algorithm for an Autonomous Sailboat," Bachelor's Thesis, Swiss Federal Institute of Technology Zurich, 2009.

[19] H. Saoud, M.-D. Hua, F. Plumet, and F. Ben Amar, "Routing and course control of an autonomous sailboat." Lincoln, United Kingdom: IEEE, Sept. 2015, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/7324218/

[20] S. Ragi and E. K. P. Chong, "UAV Path Planning in a Dynamic Environment via Partially Observable Markov Decision Process," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2397–2412, Oct. 2013, conference Name: IEEE Transactions on Aerospace and Electronic Systems.

[21] P. Rajendran, T. Moscicki, J. Wampler, B. C. Shah, K. von Ellenrieder, and S. K. Gupta, "Wave-Aware Trajectory Planning for Unmanned Surface Vehicles Operating in Congested Environments," in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Aug. 2018, pp. 1–7, iSSN: 2475-8426.

[22] P. Missiuro and N. Roy, “Adapting probabilistic roadmaps to handle uncertain maps,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, May 2006, pp. 1261–1267, iSSN: 1050-4729.

[23] L. Kavraki, M. Kolountzakis, and J.-C. Latombe, “Analysis of probabilistic roadmaps for path planning,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 1, pp. 166–171, Feb. 1998.

[24] B. Luders, M. Kothari, and J. How, “Chance Constrained RRT for Probabilistic Robustness to Environmental Uncertainty.” American Institute of Aeronautics and Astronautics, Oct. 2010. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2010-8160

[25] B. D. Luders, S. Karaman, and J. P. How, “Robust Sampling-based Motion Planning with Asymptotic Optimality Guarantees,” in *AIAA Guidance, Navigation, and Control (GNC) Conference.* Boston, MA: American Institute of Aeronautics and Astronautics, Aug. 2013. [Online]. Available: http://arc.aiaa.org/doi/10.2514/6.2013-5097

[26] B. B. D. Luders, “Robust sampling-based motion planning for autonomous vehicles in uncertain environments,” Thesis, Massachusetts Institute of Technology, 2014. [Online]. Available: https://dspace.mit.edu/handle/1721.1/90727

[27] S. M. LaValle and J. J. Kuffner, “Randomized Kinodynamic Planning,” *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001. [Online]. Available: https://doi.org/10.1177/02783640122067453

[28] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, June 2011. [Online]. Available: https://doi.org/10.1177/0278364911406761

[29] “Ragazza 1 Meter Sailboat V2.” [Online]. Available: https://www.horizonhobby.com/product/boats/boats-14532--1/boats-scale-and-sail-/sailboats/ragazza-1-meter-sailboat-v2--rtr-prb07003

[30] “3DR Pixhawk 1,” Mar. 2020. [Online]. Available: https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk.html

[31] “ArduPilot.” [Online]. Available: https://ardupilot.org/

[32] “Anemometer for Vantage Pro2™ & EnviroMonitor®.” [Online]. Available: https://www.davisinstruments.com/product/anemometer-for-vantage-pro2-vantage-pro/

[33] I. A. Sucan, M. Moll, and L. E. Kavraki, “The Open Motion Planning Library,” *IEEE Robotics Automation Magazine*, vol. 19, no. 4, pp. 72–82, Dec. 2012.

[34] J. D. Hamilton, *Time Series Analysis.* Princeton, New Jersey: Princeton University Press, 1994.

[35] S. Park, J. Deyst, and J. How, “A New Nonlinear Guidance Logic for Trajectory Tracking,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, Aug. 2004. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2004-4900