Field trials of an energy-aware mission planner implemented on an autonomous surface vehicle

Thompson, Fletcher; Galeazzi, Roberto; Guihen, Damien

Published in:
Journal of Field Robotics

Link to article, DOI:
10.1002/rob.21942

Publication date:
2020

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Thompson, F., Galeazzi, R., & Guihen, D. (2020). Field trials of an energy-aware mission planner implemented on an autonomous surface vehicle. *Journal of Field Robotics*, *37*(6), 1040-1062. https://doi.org/10.1002/rob.21942

Thompson Fletcher ORCID iD: 0000-0002-0639-9871

Field Trials of an Energy Aware Mission Planner Implemented on an Autonomous Surface Vehicle

Fletcher Thompson[*]

Section for Oceans and Arctic

Technical University of Denmark

Building 201, Kemitorvet, 2800 Kgs. Lyngby, Denmark

fletho@aqua.dtu.dk

Roberto Galeazzi

Automation and Control

Technical University of Denmark

Building 326, Elektrovej, 2800 Kgs. Lyngby, Denmark

rg@elektro.dtu.dk

Damien Guihen

National Centre for Maritime Engineering and Hydrodynamics

University of Tasmania

Newnham Drive, Newnham, Tasmania, Australia

damien.guihen@utas.edu.au

Abstract

Mission planning for Autonomous Marine Vehicles (AMVs) is non-trivial due to the dynamic and uncertain nature of the marine environment. Communication can be low-

[*] Corresponding Author

bandwidth and is not always guaranteed, so the operator must rely on the vehicles to adjust their plans according to the realised state of the environment. This paper presents the improvements made to an energy-aware mission planner that allow it to generate and adjust plans for an Autonomous Surface Vehicle (ASV) operating in an uncertain environment. The energy-aware mission planning problem was redefined as a stochastic programming problem, and a two-stage solver was developed to provide an initial plan for the ASV and then adjust it during run-time according to pre-defined recourse actions. The mission planner and ASV were trialled in Lake Waverley, Tasmania. Adjusting the recourse action criteria demonstrated that the ASV could exhibit conservative or opportunistic behaviours according to the operator's preference of safety margin. In the pursuit of extending the planner's second-stage so that it can predict a suitable recourse action ahead of time, a hybrid Long Short-Term Memory energy forecaster was trained from the Waverley mission data. Comparison of the error between the forecaster and the test data shows that the forecaster has a reliable forecast horizon of about 10 seconds.

List of Acronyms

AMV Autonomous Marine Vehicle

ASV Autonomous Surface Vehicle

AUV Autonomous Underwater Vehicle

DC Direct Current

EKF Extended Kalman Filter

EUROPtus Extensible Universal Remote Operations Planning with Neptus

GPS Global Positioning System

GUI Graphical User Interface

HMI Human-Machine Interface

IMU Inertial Measurement Unit

LOS Line of Sight

LSTM Long Short-Term Memory

OP-SW Orienteering Problem with Stochastic Weights

OR Operations Research

PID Proportional Integral Derivative

PSO Particle Swarm Optimisation

PSR Point of Safe Return

RC Radio Controller

RMSE	Root-Mean-Squared Error
RNN	Recurrent Neural Networks
ROS	Robot Operating System
SAA	Sample Average Approximation
TOP	Team Orienteering Problem
T-REX	Teleo-Reactive Executive
UTM	Universal Transverse Mercator

1 Introduction

This paper considers automated mission energy planning and forecasting for Autonomous Marine Vehicles (AMVs) with electric power supplies. The energy supply of an AMV is vital to its successful operation, power failures are one of the leading causes of overall mission failure (Brito et al., 2014). Effectively managing the energy supply means not only ensuring that the AMV does not overextend itself with an overambitious mission plan, it also means making the most use of the available energy of the vehicle to achieve as many mission tasks as possible. Currently, operators balance the survival of their vehicle and its task completion effectiveness by consulting the manufacturer specifications (e.g. range, endurance, rated depth) and applying safety factors to those specifications in order to obtain safe mission planning constraints. In pursuit of refining this balance to maximise the effectiveness of the vehicle while still maintaining survivability, this paper considers energy management from an onboard mission planning and decision-making approach.

The reasoning and deliberation capabilities of AMVs have blossomed over the last decade. Prior to 2008, mission planning for AMVs was a task reserved for the human operator, who would create a scripted set of way-points, depths, speeds, control modes, and sensor payload modes. This procedure relied on the operator's experience to predict the reliability of the mission script, and to adequately prepare the vehicle with contingencies for outcomes that are likely to occur. McGann et al. (2008) ushered in a new method of planning for AMVs by implementing the deliberative Teleo-Reactive Executive (T-REX) agent onboard Autonomous Underwater Vehicles (AUVs). T-REX effectively allows vehicles to adjust their plan and behaviours during execution according to detected external changes and inferred changes to the mission state.

With AMVs able to adapt plans in order to achieve mission objectives, automated mission planning has been extended to dynamically generate and adapt mission plans for large operations. Mesoscale ($\geq$ 50 km^2) coordinated multi-AMV operations have been realised through the development and implementation of temporal planners such as Extensible Universal Remote Operations Planning with Neptus (EUROPtus) (Py et al., 2016). Temporal plans schedule and allocate tasks to the vehicles using time as the base resource constraint. A partial plan is instantiated and is refined into a complete plan as flaws are observed during execution. Temporal planning allows for easy synchronisation of individual vehicle plans, which is convenient for operators when deploying and retrieving vehicles (Ferreira et al., 2018), or for mixed-initiative missions (Ai-Chang et al., 2004). However, the environmental loadings experienced by the vehicles while deployed are not directly considered in temporal planning. Instead the planner relies on the time taken for the vehicle to perform tasks and its speed as the relevant temporal indicators.

In Thompson and Galeazzi (2020), an energy based planner was proposed to predict the energy cost for a team of vehicles to perform tasks. It then uses these predictions to schedule and allocate tasks to individual vehicles based on their available energy resources. Energy planning factors in the loadings on the vehicles traversing waypoints along an expected path (something that is not considered by temporal planners) and can be compared against the vehicle's measured power consumption during deployment.

Aspects of mission planning for autonomous vehicles can be found in the field of Operations Research (OR), where logistical planning problems are defined as optimisation problems and then solved. The Team Orienteering Problem (TOP) (Tsiligirides, 1984; Chao et al., 1996) is a good candidate for the modelling of standard AMV deployments where vehicles must visit operator-specified positions of interest in order to perform tasks (such as sampling the environment and performing intervention actions). Variants of the TOP have also been implemented for the planning of multi-AMV correlated scalar field sampling missions (Tsiogkas and Lane, 2018). Adapting the TOP formulation for deployment in uncertain environments, where the energy costs for vehicles to perform tasks is not deterministic, requires the TOP to be configured for Stochastic Weights (TOP-SW).

Two-stage solutions to the Orienteering Problem with Stochastic Weights (OP-SW) have been implemented at a simulation level (Evers et al., 2014; Shang et al., 2016). The first stage selects a route for a singular vehicle based on the expected weight costs for each transition. In the second stage these weights are realised and a ’return home’ recourse action is implemented if the realised total cost exceeds the total limit. The profit shortage cost (i.e. the number of points not visited because of the recourse action) in summation with the first stage’s profit is used as a global objective function. Maximising the global objective function creates a route that maximises points collected and minimises the expected profit-shortage consequence. The second stage method presented in Evers et al. (2014) uses a OP-SW heuristic adapted from Sample Average Approximation (SAA). SAA performs Monte Carlo simulation on the weights (which are random variables) to construct the objective function as a deterministic mixed-integer programming problem. While these two-stage solvers provide robust solutions, they are limited in their scope based on what actions the vehicle can take at any given moment. For example, the vehicle could choose to skip the current task if it is taking longer than expected to complete.

This paper continues to develop the energy-aware planner from Thompson and Galeazzi (2020) by implementing it onboard a prototype marine vehicle platform. Marine robots operate in a dynamic and uncertain environment that imparts non-linear and uncertain forces onto the vehicles. In Section 2, we propose an AMV mission planner that is inspired by the two-stage method used in OR for solving the OP-SW, but adapted for *in situ* decision-making.

The first stage (Section 2.2) computes the expected task sequence using the Monte Carlo sampling method in Thompson and Galeazzi (2020) *a priori* to vehicle deployment. The second stage (Section 2.2.1) occurs during deployment of the AMV, and is computed locally onboard the vehicle. During the mission execution, the weights for each section of the plan are revealed sequentially. This, coupled with the potential for vehicle-to-shore communication dropouts, makes the two-stage solvers difficult to implement as SAA or other solvers are too computationally expensive to execute onboard the computer of an out-of-contact AMV. Instead, we propose a supervisor agent acting onboard the AMV that decides whether to enact one of several recourse actions arranged in the subsumption architecture Brooks (1986) style:

1. Continue current plan.
2. Skip the current task.
3. Request a replan from the shore mission planning agent.
4. Return to the rendezvous (home) position.
5. Emergency power saving mode.

To enable the supervisor to decide on one of these actions, three probabilistic metrics are proposed (Section 2.3):

1. Confidence that the energy allocated for the current task has not been exceeded.

2. Confidence that the energy allocated for the current plan has not been exceeded.

3. Confidence that the energy capacity of the battery (or some fraction of it) has not been exceeded.

The confidence metrics are the result of computing the survival function of the predicted energy consumption distributions generated by the first stage planner, and using the measured energy consumption of the battery as input. In this context, the survival function provides an estimate of how likely an energy consumption measurement reading has exceeded a predicted task, plan, or battery distribution. An operator can then specify acceptable confidence thresholds for the supervisor that control the minimum confidence of the supervisor before a recourse action is activated.

A prototype Autonomous Surface Vehicle (ASV) platform was designed with the specific purpose of testing the outlined two-stage planning approach (Section 3). The ASV was deployed in a lake environment, where fluctuating winds produced uncertain external forces that were not directly available for consideration by the mission planner or the feedback control system. During trials (Section 5.1), combinations of confidence metrics were used to produce trajectories that conserved the original plan before returning home, and others that actively changed the plan to find achievable tasks.

To allow the supervisor agent to look ahead in time so that it can make recourse action decisions sooner, this paper also proposes a data-driven approach to forecasting the vehicle's energy consumption. Forecasting energy consumption has been achieved for ground robots through linear regression and Bayesian estimation (Sadrpour et al., 2013), and through encoding the mission tasks into a Long Short-Term Memory (LSTM) network (Hamza and Ayanian, 2017). Marine vehicle dynamics are non-linear, and the marine environment is much more dynamic and uncertain than terrestrial environments. In this respect, the use of non-linear regression models and probabilistic models are more likely to succeed in forecasting. LSTMs are an adaptation of Recurrent Neural Networkss (RNNs) that include input, output and forget gates in order to overcome the vanishing gradient problem experienced by RNNs. LSTMs have seen significant success in sequential data problems such as handwriting recognition (Greff et al., 2017), weather forecasting (Zaytar and Amrani, 2016), and ocean surface temperature forecasting (Caley et al., 2017) as they are able to identify and remember important features that influence the data later on.

In Section 4, we propose a hybrid LSTM network control model to predict the motion of the vehicle, output of the vehicle's thrusters, and subsequent energy consumption. The LSTM networks were trained on the data gathered from the lake trials, and analysis of the hybrid energy forecaster shows that it is capable of reliably forecasting the energy consumption of the vehicle up to 10 seconds into the future.

2 Stochastic Programming Formulation

2.1 Original Mission Planner Definition

In Thompson and Galeazzi (2020), the multi-AMV mission planning problem was modelled as the TOP (Chao et al., 1996). The following definitions of a vehicle, task, and open and closed mission plans are presented for completeness:

$$V = (e_b, I_v) \tag{1}$$

$$T = (g, s, I_t) \tag{2}$$

$$M_o = (\mathcal{T}, \mathcal{V}, \mathcal{O}, P, Q, E) \tag{3}$$

$$M_c = (\mathcal{T}, \mathcal{V}, R, S, F) \tag{4}$$

where V is a vehicle, represented by a tuple containing the energy capacity of the battery (e_b) in Watt-hours (Wh) or Joules (J) and I_v is a tuple containing additional information about the vehicle (speed, operating domain, capabilities, etc.). T is a task, represented by a tuple containing the positional information of the task (g), the operator specified reward for completing the task (s), and I_t is a tuple containing additional information about the task (e.g. payload requirements, requisite and dependent tasks). To accommodate for missions where there are N_V vehicles and N_T tasks, $\mathcal{V}$ and $\mathcal{T}$ are defined as the accumulated set of defined V and T.

The first step of the planner is to use the above information to create the open mission, M_o, which represents the complete domain that the planner searches through to obtain the closed mission M_c. P is a reference vector containing sequential integers that reference elements of $\mathcal{T}$. Q is a similar reference vector for $\mathcal{V}$. E is a zero-diagonal matrix of energy costs for transitioning between the ith and jth tasks ($\mathcal{T}_{P_{ij}}$).

The energy cost for E_{ij} is the result of performing a Monte Carlo simulation of size N on the marine vehicle dynamic model. Monte Carlo simulation of the model was necessary to capture the uncertainty of the hydrodynamic coefficients used in the model. The simulation first produces N time-varying sets of body forces required for the vehicle model to move along a reference trajectory. Each set of body forces are then decomposed into N sets of actuator allocations using a control allocation algorithm. Each set of actuator allocations are then converted to power consumptions through identified thrust-power relationships for each actuator. The summation of these actuator power consumption sets as well as the vehicle's hotel load produces N time-varying total power consumptions for the simulated vehicle models along the reference trajectory, $\mathcal{P}_k(t)$. The energy cost is the expected value (denoted by the operator $E\left[\cdot\right]$, not to be confused with the energy cost E) of the integral of these distributions with time:

$$E_{ij} = E\left[\frac{\sum_{k=1}^{N}(\int_{t_i}^{t_j}\mathcal{P}_k(t)\,\mathrm{d}t)}{N}\right] \tag{5}$$

$\mathcal{O}$ is a set of tuples that contain obstacle information necessary for collision avoidance path-planning and will not be considered further in this paper as it is tangential to the main question of energy planning. R is a N_V long set, each element of which contains a subset of P that represents the ordered sequence of tasks allocated to a vehicle. S is the set of rewards accumulated from completed tasks in R. F is a N_V long set, each element of which contains a subset of E that correspond to the energy costs for each task scheduled and allocated according to R.

The planner formulates the search for an optimum M_c into the following optimisation problem:

$$\begin{aligned} \underset{M_c}{\text{maximise}} \quad & \sum_{x_i \in S} x_i \\ \text{subject to} \quad & \sum_{y_i \in F_Q} y_i \leq e_b \in \mathcal{V}_Q \end{aligned} \tag{6}$$

where the goal is to maximise the reward collected in S while ensuring that the sum of energy costs in F do not exceed battery constraints of each corresponding vehicle (e_b).

2.2 Adaptation for Stochastic Weights

Even though the energy for each transition and task was obtained as a random variable through Monte Carlo simulation, the planner in Thompson and Galeazzi (2020) only uses the expected values (Eq. (5)) and does not consider the full distribution of possible energy consumptions. This seems like a sensible choice as the expected value is the most likely amount of energy to be consumed for a given task transition (provided the distribution is Gaussian). However, this does not remove the chance that the energy consumption is more than expected, which could jeopardise the feasibility of the entire plan. In reality, these transition weights are complex and non-trivial to determine for certain, and depend upon the following:

1. Satisfactory identification of the vehicle's dynamic model.

2. Satisfactory identification of the vehicle's propulsion thrust/power relationship.

3. An accurate model of the vehicle's mechanical and electric efficiency up to the power source.

4. An accurate model of the wind, wave, and current forces acting upon the vehicle.

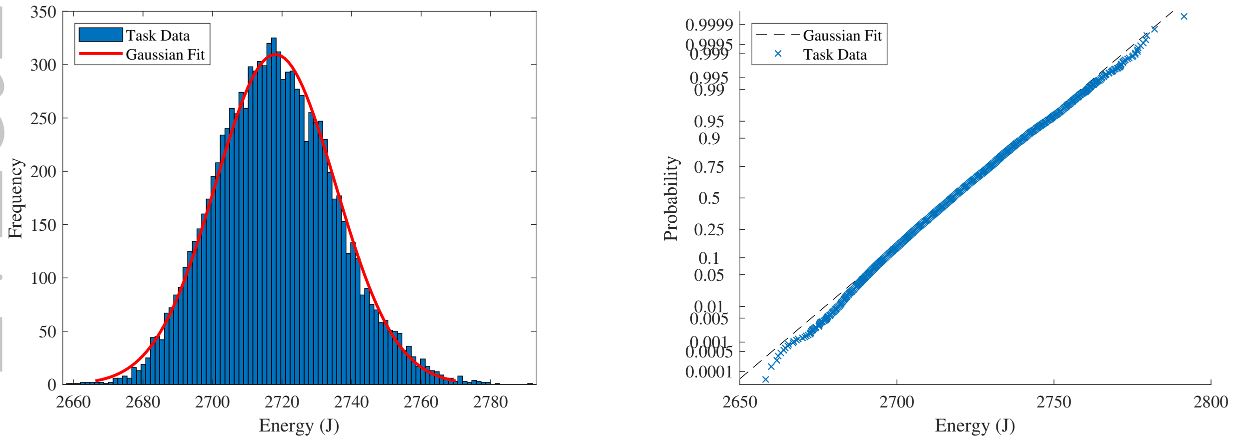
5. Well designed controllers that are able to track plan-generated reference trajectories.

In particular, variance and unknown parameters within the environment model contribute to unpredictable behaviour in the controllers, leading to a higher variance in the *a priori* mission energy consumption prediction. Therefore, the planner must in some way accommodate for situations in which the realised energy consumption for a given task transition is greater than expected. The same can be said for the reverse situation where the realised consumption is less than expected.

To account for this uncertainty, OR researchers consider solutions to the OP-SW. A successful strategy for solving the OP-SW is to first solve the OP with the expected values of the weights. Then, once the vehicle is deployed on the initial route, a second stage solver keeps track of each transition's true weight once it has been realised. It then initiates a 'go to finish' or 'return home' recourse when the remaining transition costs plus the realised costs exceed the total OP cost allowance.

The mission planner performs Monte Carlo simulation upon a sampled vehicle model to obtain the task energy requirement distributions before the vehicle is deployed (the expected values of these distributions are used to form E in the original M_o definition). For the purposes of minimising computational resources in the solving of the OP-SW and also in minimising communication overhead between the vehicle and shore, it would be advantageous to be able to parameterise the output distributions with a fitted standard distribution. The simplest fit to approximate the distribution (at least in number of parameters) is the Gaussian distribution, requiring just the mean and the variance.

Testing the output distributions of the Monte Carlo simulation for normality using the Andersen-Darling test statistic (at 5% significance) showed that the distributions are not from a Gaussian distribution. This means that the output distributions are not strictly Gaussian, and errors will have to be accepted if a Gaussian approximation is used. Consider the example distribution of the energy prediction for a transition task in Fig. 1. It is clear from Fig. 1b that the distribution loses correlation with the Gaussian fit at the upper and lower 5% boundaries ($x > 0.95$ and $x < 0.05$). On close inspection, the Gaussian fit overestimates the likelihood of the task's energy requirement towards the lower 5% boundary, and underestimates the likelihood of the requirement towards the upper 5% boundary. This means that if a Gaussian fit is used, the planner will have a tendency to use an optimistic prediction of the energy consumption due to the approximation error. Given these limitations, it must be acknowledged that approximating the distribution as Gaussian is an engineering trade-off between accuracy of the model prediction and the practical limitations of computation and communication in the field.



(a) Task energy histogram with Gaussian fit.

(b) Task energy probability plot with Gaussian fit.

Figure 1: Plots of an example task energy distribution generated from 10,000 simulations of varying vehicle dynamics models using Eq. (5). Fig. 1b shows a correlation with a Gaussian distribution between the upper and lower 5% boundaries.

In this paper, we accept the implications of using a Gaussian approximation, and model the generated distributions with only their mean and standard deviation. E is then redefined as two separate matrices, μ_E and σ_E, representing the mean and standard deviation of the ijth task transition respectively. M_c is then solved by the planner using μ_e instead of E. F is similarly redefined into μ_F and σ_F, which gives the means and standard deviations of the ordered sequence of transition weights for each vehicle respectively.

The stochastic energy prediction, H, is defined as a random variable pertaining to the task, plan or battery capacity as follows:

$$H \sim \mathcal{N}(\mu, \sigma^2) \tag{7}$$

$$H_t \sim \mathcal{N}(\mu_{F_i}, {\sigma_{F_i}}^2) \tag{8}$$

$$H_p \sim \mathcal{N}(\sum \mu_F, \sum {\sigma_F}^2) \tag{9}$$

$$H_b \sim \mathcal{N}(\mu_{e_b}, {\sigma_{e_b}}^2) \tag{10}$$

where H_t is the energy prediction for a particular task, H_p is the energy prediction for the summation of tasks to be performed, and H_b is a random variable obtained based on battery discharge/recharge data.

2.2.1 Naive Energy Consumption Certainty Estimation

The vehicle is equipped with sensors to measure the voltage and current consumption close to the battery terminals. The measured energy consumption of the vehicle is calculated by performing numerical integration using the trapezoidal rule of the measured power consumption over the time interval $\Delta t = t(k) - t(k-1)$:

$$E_m(k) = \frac{P_m(k) + P_m(k-1)}{2}\Delta t + E_m(k-1) \tag{11}$$

With the energy consumption prediction now represented as a Gaussian random variable (H), a simple metric to determine the likelihood that the vehicle has consumed more than the prediction is the survival function:

$$S_H(E_m(k)) = P(H > E_m(k)) = 1 - \int_{-\infty}^{E_m(k)} \frac{1}{\sqrt{2\pi\sigma_H^2}} e^{-\frac{(x-\mu_H)^2}{2\sigma_H^2}} \, \mathrm{d}x \tag{12}$$

The survival function metric allows the operator to specify a lower limit (δ) for the supervisor based on the likelihood that $E_m(k)$ is not greater than H. If $S_H < \delta$, then the supervisor will activate a recourse action behaviour. For $\delta > 0.5$, the operator is encouraging the supervisor to be conservative and activate recourse actions earlier and *vice versa* for $\delta < 0.5$.

2.3 Recourse Actions

By implementing S_H and the δ condition across different energy consumption scales, several levels of decision making for the supervisor can be designed based on the expected operations of a deployed vehicle. For example, when the vehicle switches to its own power supply, it must then keep track of the energy consumed when compared to the estimated energy capacity of the vehicle, e_b. When the supervisor commences a plan given to it by the mission planner, it must compare the energy consumed against the predicted total energy consumption of the plan. Finally, each plan is a sequence of tasks, each of which should be considered on the task energy prediction scale.

To formalise this we defined separate datum points for the battery, plan, and task energy scales.

1. On vehicle power source mode switch to battery (battery datum), o_b
2. On commencement of a plan (plan datum), o_p.
3. On commencement of a task (task datum), o_t.

As the vehicle progresses through a mission, task E_t, plan E_p, and battery E_b scales are simultaneously evaluated in the equations below.

$$E_b(k) = E_m(k) - o_b \tag{13}$$

$$E_p(k) = E_m(k) - o_p \tag{14}$$

$$E_t(k) = E_m(k) - o_t \tag{15}$$

These energy measurements are then compared with their respective H energy predictions (Eqs. (8) to (10)) based on the survival function activation criteria (Eq. (12)). As an additional fail-safe, we also place a hard limit on the minimum measured voltage of the battery, V_{lim}. The recourse actions and their activation conditions are listed below:

1. $S_{H_t}(E_t(k)) < \delta_t$: skip task heuristic.
2. $S_{H_p}(E_p(k)) < \delta_p$: replan heuristic.
3. $S_{H_b}(E_b(k)) < \delta_b$: return to rendezvous (home) position.
4. $V_m(k) < V_{lim}$: emergency power saving mode.
5. Otherwise: continue current plan.

The first and second recourse action activations are described in the following subsections. The third activation commands the vehicle to travel to the home point. The fourth activation is an emergency fail-safe mode triggered when the voltage of the battery has dropped below the minimum voltage requirement of the thrusters. The vehicle shuts down the motors and is stranded.

2.3.1 Task Skip Heuristic

The task skip heuristic enables when the task survival function is below the task survival threshold. The supervisor performs a naive linear estimate of the energy remaining for the current task, E_T' by:

$$E_t' = \frac{\mathcal{S}_{bc}}{\mathcal{S}_{ac}} E_t \tag{16}$$

where $\mathcal{S}_{bc}$ is the distance remaining on the predicted trajectory from the vehicle's current position to the goal, and $\mathcal{S}_{ac}$ is the total distance of the predicted trajectory. A process to decide whether to skip or not skip the remainder of the current task is given in Alg. 1.

Algorithm 1: Task skip heuristic.

```
input : Energy remaining for current task $E_t'$
        Reward of current task $s_t$
        Set of rewards of tasks remaining $S$
        Set of energy costs of remaining tasks $F$
output: True: skip task or False: keep task

$e = 0$;
$s = 0$;
$[F', I] =$sort($F$);
$S' = S(I)$;
for $i \leftarrow 1$ to $dim_F(F')$ do
    $e \leftarrow e + F'(i)$;
    $s \leftarrow s + S'(i)$;
    if $e < E_t'$ then
        if $s > s_t$ then
            return True;
    else
        return False
```

The task skip heuristic sorts in ascending order the remaining tasks for the current plan by predicted energy cost (F_{ij}), and iteratively aggregates the energy cost of this sorted list until it exceeds E_t'. If the accumulated reward of the sorted, remaining tasks exceeds the reward for the current before this condition then the algorithm returns true (i.e. skip the task). Otherwise the algorithm returns false (i.e. keep the task).

2.3.2 Replan Action

During deployment, the vehicle keeps a track of the tasks that were completed and the tasks that were skipped due to the task skip recourse action. Upon activation of the replan recourse action, the vehicle first must determine if it can communicate with the first stage of the mission planner. If communication is successful, it sends the request $\mathcal{R}$ to the mission planner, containing the following information:

$$\mathcal{R} = (\mathcal{C}, \mathcal{D}, \mathcal{E}, L, E(k)) \tag{17}$$

where $\mathcal{C} \subset R$ is the set of completed tasks, $\mathcal{D} \subset R$ is the set of skipped tasks, $\mathcal{E}$ is the set of final energy measurements for each completed task (E_t) and L is the location of the vehicle at the time of replan request. Some of the elements in $\mathcal{E}$ will contain the consumed energy from tasks that were skipped previous to it. The vehicle then holds its current position while the mission planner generates a new solution from $\mathcal{R}$ and updates M_c. Once the vehicle receives the updated M_c, it begins the new plan. If the vehicle is not within communication range, it activates the 'return home' recourse action. Ideally, the replanning recourse would happen entirely onboard the vehicle. However, due to the computational constraints of current small form factor embedded computers that typically run the software of AMVs, the replanning steps must be outsourced to an external computer (such as a shoreside system) that can handle the planning requirements.

One potential method for enabling online replanning on a low-cost embedded system would be to create a lightweight planning agent that only uses the task information given in the original plan to generate a replan solution. This means that not all potential tasks will be considered, but the vehicle would then be able to create a new plan based on a subset of the old. This also ensures, in a multi-AMV deployments, that each vehicle would be guaranteed not to create conflicts with other vehicles by allocating itself an already allocated task. This comes with the caveat of restricting each vehicle's knowledge of the global mission state, meaning that vehicles will be unable to act on tasks that weren't initially given to them. This increases the risk of mission failure due to local vehicle failures. A distributed planning architecture, such as described in Zlot (2006) and Sotzing et al. (2007), would enable vehicles to actively give, take and swap tasks according to their replan actions.

Upon receiving $\mathcal{R}$ from the vehicle, the first stage planner formulates a new M_o based on the tasks in the old M_c that were neither skipped nor completed. This reduces the size of E that has to be searched through, because the rows and columns of the previous E that reference starting from, or moving to completed or skipped tasks can be deleted. The energy constraint (e_b from Eq. (6)) is replaced with the previous plan's energy prediction minus the energy consumed during deployment (replace e_b with $\mu_{H_p} - E(k)$ in Eq. (6)). Additionally, the planner must redefine E_{1j} and E_{i1} with an energy distribution prediction based on the provided L, which is the new starting point of the vehicle in the new plan. By reducing the size of E and only performing Monte Carlo simulation on the subset of trajectories that start at L, the replanning process time is a fraction of the initial plan generation time.

3 System Description

The full system is comprised of two components: the shoreside mission planner and Human-Machine Interface (HMI), and the ASV[1]. The shoreside systems and ASV communicate with each other over

[1] The ASV system framework is open-source and has been made available at https://github.com/FletcherFT/asv_framework

a Wi-Fi network, with a Radio Controller (RC) included as a manual override backup. Both systems were built using the Robot Operating System (ROS) middleware.

3.1 Shoreside Subsystem

Fig. 2 depicts the major components of the shoreside system. The first stage of the mission planner described in Sections 2.1 to 2.2 was developed in MATLAB and extended with the robotics system toolbox to allow it to interface with the ROS network. The operator provides the mission planner with $\mathcal{T}$ and $\mathcal{V}$, and the mission planner computes an optimal plan that it sends to the ASV via a ROS message. During mission run-time, the planner listens for planning requests from the ROS communication layer.

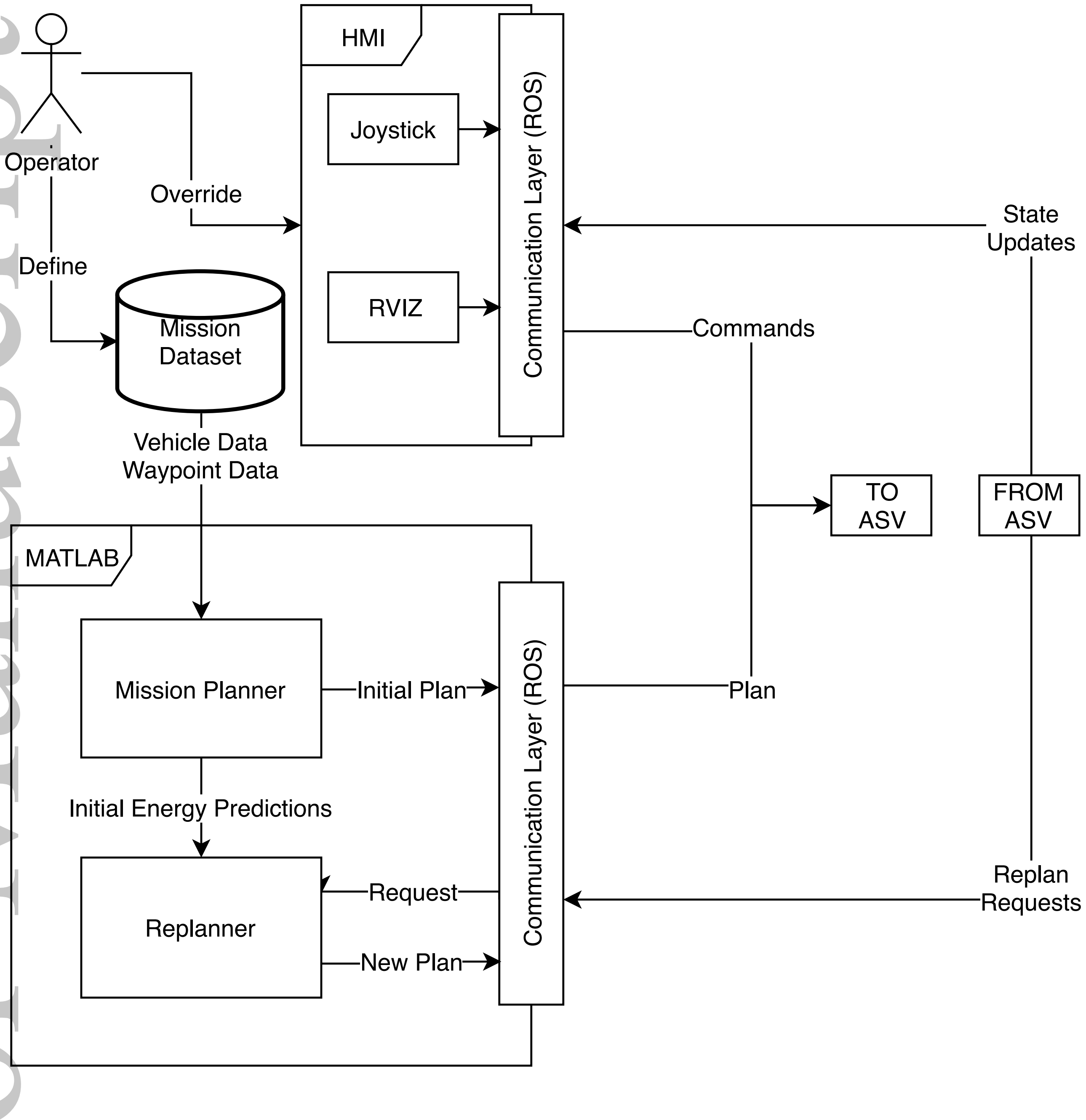
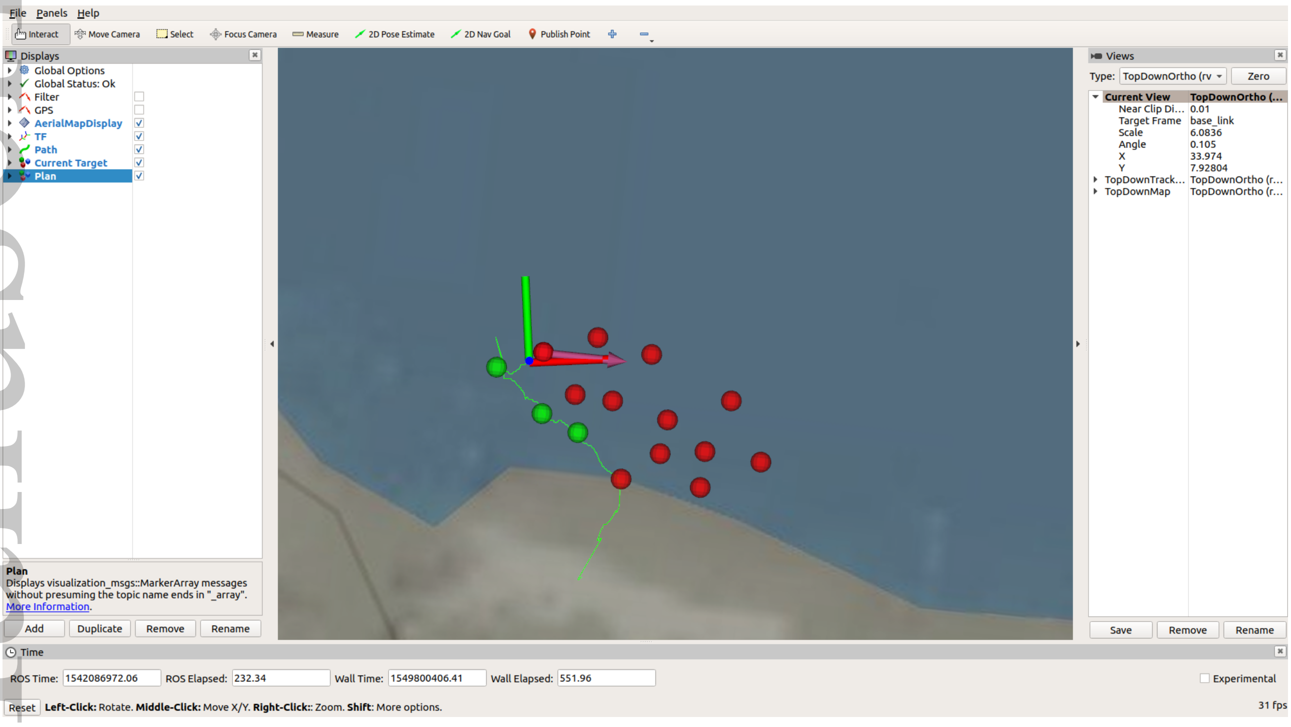


Figure 2: Layout of the shoreside system.

The HMI (see Fig. 3) is used by the operator to sanity check the navigation of the ASV by seeing on a map the vehicle's own estimate of its position and orientation. The HMI was built in ROS using RViz and a satellite map plugin. Override waypoint commands can be sent to the vehicle by clicking and dragging a position and orientation on the map, causing the vehicle to pause the current mission and navigate to the specified position. The operator can also see the vehicle's progress on the current mission red and green markers that indicate if a waypoint has been visited or not. A simple Graphical

User Interface (GUI) to send start/pause mission commands, calibration commands to the navigation system, and Proportional Integral Derivative (PID) tuning commands to the control system was also implemented.



Plan
Displays visualization_msgs::MarkerArray messages without presuming the topic name ends in "_array".
More Information.

Type: TopDownOrtho (rv

Current View	TopDownOrtho (...
Near Clip Di...	0.01
Target Frame	base_link
Scale	6.0836
Angle	0.105
X	33.974
Y	7.92804
TopDownTrack...	TopDownOrtho (r...
TopDownMap	TopDownOrtho (r...

ROS Time: 1542086972.06 ROS Elapsed: 232.34 Wall Time: 1549800406.41 Wall Elapsed: 551.96

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click::** Zoom. **Shift**: More options.

31 fps

Figure 3: Screenshot from the HMI component of the shoreside system. The vehicle's position and orientation is provided by the red/green/blue axis, incomplete waypoints are shown as red spherical markers, completed tasks are shown in green. The selected waypoint for completion is shown with a red marker and fuchsia arrow.

3.2 ASV Subsystem

A simple model-scale box-barge hull was converted into a prototype autonomous platform for the purposes of testing planning, guidance, navigation and control algorithms. The vehicle, pictured in Fig. 4, has three actuators: one tunnel thruster located towards the bow and two azimuth thrusters located at the transom. This configuration means the vehicle can move independently in forwards (surge), sideways (sway), and heading (yaw) motions. To take advantage of this, an autopilot and a dynamic positioning controller were designed to control the vehicle in two different operating modes:

transition mode (autopilot), and hold position mode (dynamic positioning). The ASV is equipped with a u-blox LEA-6H GPS for position feedback, and a Redshift UM7 IMU for orientation, angular velocity, and acceleration feedback. Velocity measurements from the GPS, such as course-over-ground information, were not included for state estimation as the LEA-6H obtains velocity information from the derivative of the position (not through an independent measurement such as Time-Differenced Carrier Phase).

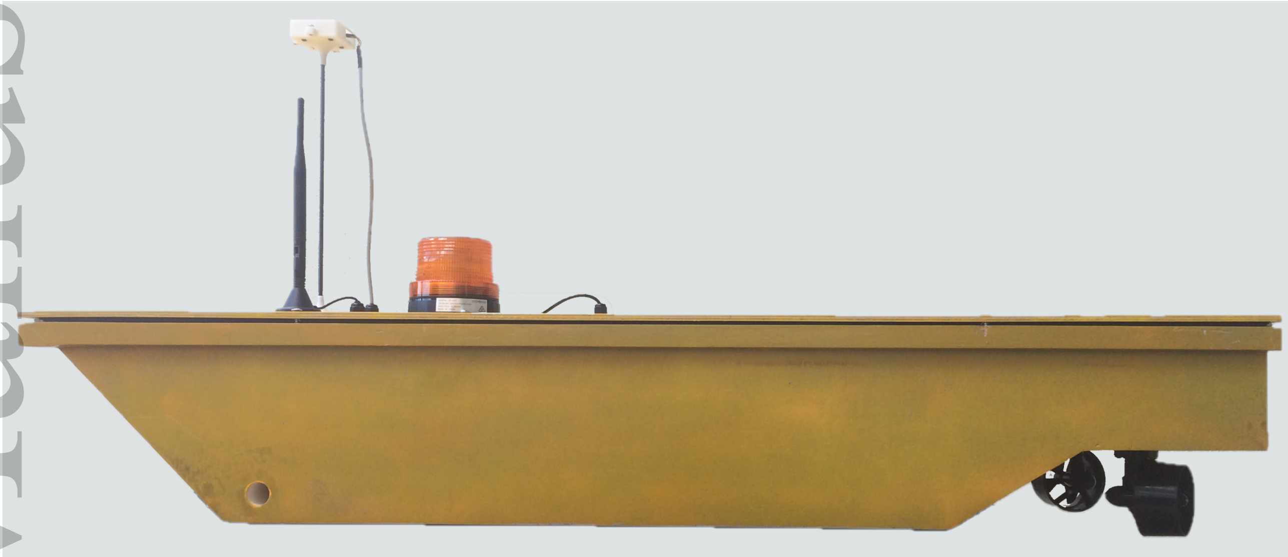


Figure 4: Profile view of the ASV. Two fully rotatable 70 mm thrusters are positioned on the aft transom. A tunnel thruster is positioned near the bow. The GPS and Wi-Fi communication masts are positioned on the deck outside of the water resistant enclosure. During trials, the IMU was also moved onto the deck to reduce magnetic interference from the DC motors.

Fig. 5 depicts the general subsystems of the ASV. At the top of the process hierarchy is the supervisor agent, which has several roles. Firstly it is designed to receive and execute missions plans from the shoreside mission planner. During mission execution it configures the ASV for the current task by passing waypoint information to the guidance system and sending configuration commands to the control system. It uses the energy consumption information provided by the energy monitor to perform the recourse action decision-making described in Sections 2.2 to 2.3. Lastly, it accepts operator commands that override and subsume all other activities.

The lower levels of the ASV system are centered around the Guidance, Navigation and Control paradigm described in Fossen (2011a). The navigation module consists of drivers for reading the GPS and IMU, and an Extended Kalman Filter (EKF) state estimator. The EKF is based on a 2D constrained general point kinematic model described in Moore and Stouch (2014). It fuses the GPS and IMU measurements into an estimate of the vehicle's position (in Universal Transverse Mercator (UTM) coordinates), orientation (following an East-North-Up convention), and angular and linear velocities in the body-frame of the vehicle (following a Forward-Port-Up convention). The chosen framing conventions differ from the marine robotics standard of using North-East-Down/Forward-Starboard-Down because the core ROS coordinate system operates by default in East-North-Up/Forward-Port-Up.

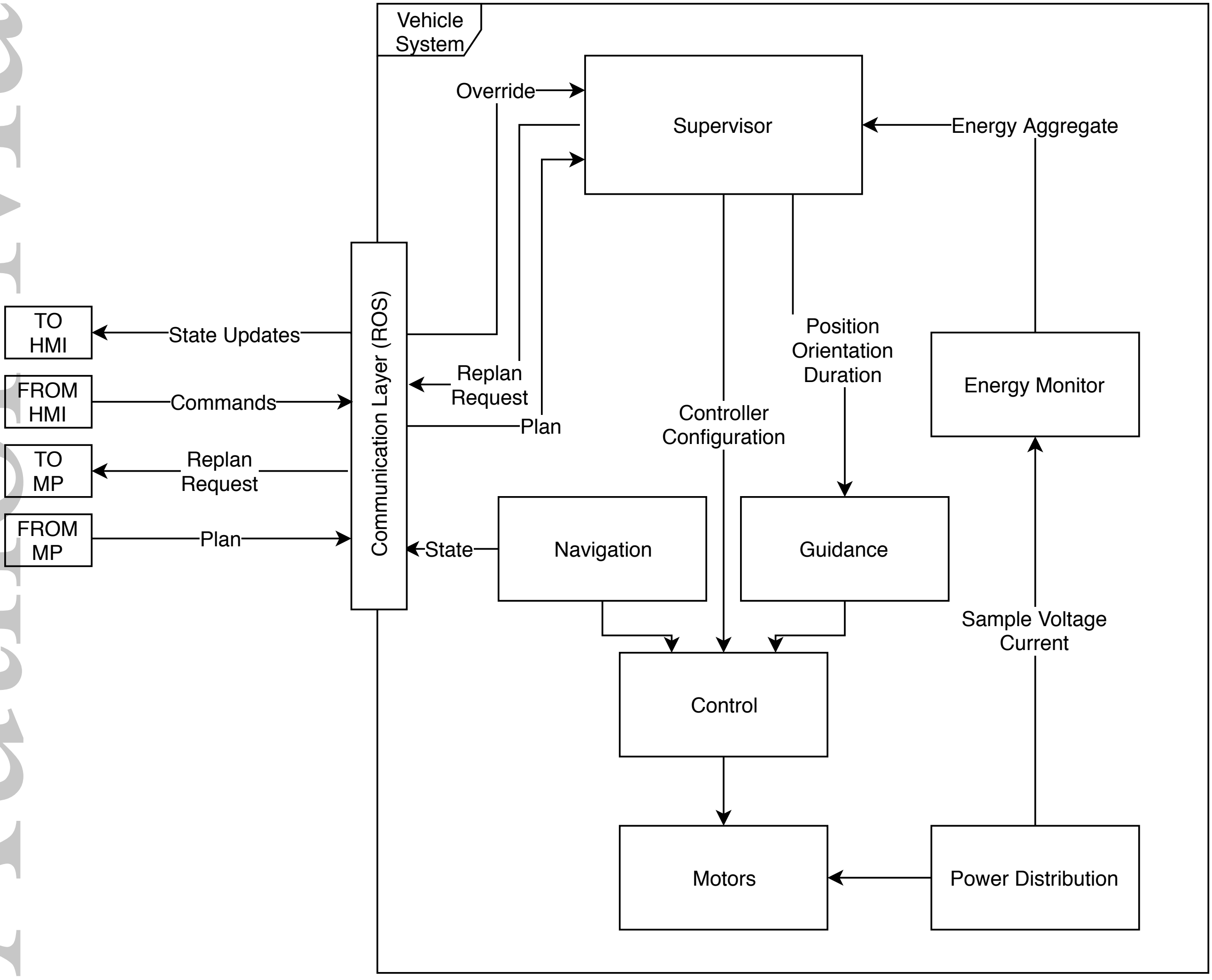


Figure 5: Subsystem layout of the ASV.

The guidance module uses the Line of Sight (LOS) guidance controller from Fossen (2011b) with a custom piecewise function (Eq. (20)) to provide speed and heading errors for the control module to regulate to zero. The guidance distance and heading error, D and ψ_e are:

$$D = \sqrt{(x_d - x)^2 + (y_d - y)^2} \tag{18}$$

$$\psi_e = \mathrm{atan2}(y_d - y, x_d - x) - \psi \tag{19}$$

where the terms are depicted in Fig. 6.

The autopilot component within the control module requires a forward speed error, which is calculated in the guidance module as follows:

$$u_d(D, r_a, \psi_e) = \begin{cases} 0 & D \leq r_a \\ 0 & |\psi_e| > \frac{\pi}{5} \\ U & D > r_a, |\psi_e| \leq \frac{\pi}{5} \end{cases} \tag{20}$$

$$u_e = u_d - u \tag{21}$$

where r_a is the radius of acceptance (i.e. the maximum distance the vehicle can be from the waypoint). U is a forward speed setpoint that the operator specifies in I_v. Eq. (20) contains conditional setpoint changes to slow the vehicle down and reduce its turning circle for large guidance heading errors (arbitrarily defined as anything larger than $\frac{\pi}{5}$ radians), and to stop the vehicle entirely when it has reached the radius of acceptance for the waypoint. When the waypoint is reached, the guidance module requests a new waypoint from the supervisor or idles if there is none.

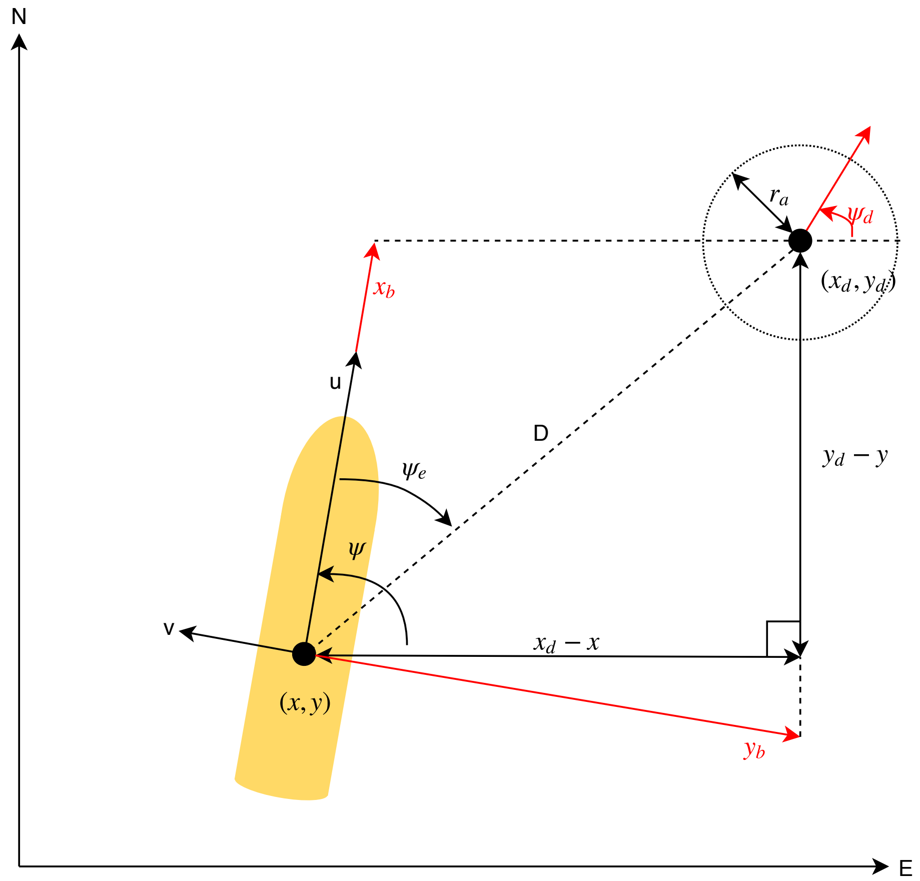


Figure 6: Geometry that the LOS guidance controller uses to calculate the autopilot and dynamic positioning controller errors. The autopilot error signals depend on ψ_e and D. The dynamic positioning error signals depend on x_b, y_b, and ψ_d. The red lines are the projection of the ASV body-frame coordinates onto the position errors, which are used to calculate x_b and y_b.

When the ASV is in dynamic positioning mode, the guidance module provides the control module with longitudinal and transverse position errors by rotating the inertia-frame components of D into the body-frame:

$$R = \begin{bmatrix} -\sin\psi & 0 & 0 \\ 0 & \sin\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{22}$$

$$\begin{bmatrix} y_b \\ x_b \\ \psi_b \end{bmatrix} = R \begin{bmatrix} x_d - x \\ y_d - y \\ \psi_d - \psi \end{bmatrix} \tag{23}$$

where x_b and y_b are the body-frame position errors as depicted in Fig. 6. ψ_b is the error between the desired vehicle heading and the current vehicle heading.

The control module consists of an autopilot controller, a dynamic positioning controller, and a control allocation algorithm. The autopilot and dynamic positioning controllers regulate the speed and heading, and position and heading respectively by outputting a vector of commanded body forces in the surge and sway directions, and a yawing moment ($\boldsymbol{\tau^c}$). The autopilot controller is a multi-input, multi-output PID controller that outputs $\boldsymbol{\tau^c}$ as:

$$\tau_X^c = K_p u_e + K_i \int u_e \, \mathrm{d}t + K_d \frac{\mathrm{d}u_e}{\mathrm{d}t} \tag{24}$$

$$\tau_Y^c = 0 \tag{25}$$

$$\tau_{N_Z}^c = K_p \psi_e + K_i \int \psi_e \, \mathrm{d}t + K_d \frac{\mathrm{d}\psi_e}{\mathrm{d}t} \tag{26}$$

The transverse velocity is not regulated, which can cause the vehicle to drift sideways when turning.

The dynamic positioning controller calculates $\boldsymbol{\tau^c}$ through the following:

$$\tau_X^c = K_p x_b + K_i \int x_b \,\mathrm{d}t + K_d \frac{\mathrm{d}x_b}{\mathrm{d}t} \tag{27}$$

$$\tau_Y^c = K_p y_b + K_i \int y_b \,\mathrm{d}t + K_d \frac{\mathrm{d}y_b}{\mathrm{d}t} \tag{28}$$

$$\tau_{N_Z}^c = K_p \psi_b + K_i \int \psi_b \,\mathrm{d}t + K_d \frac{\mathrm{d}\psi_b}{\mathrm{d}t} \tag{29}$$

where ψ_d is the desired heading for the vehicle.

The control allocation algorithm is a quadratic programming solver formulated as the constrained fixed-angle thruster allocation problem described in Fossen (2011c). Although the thrusters are fully rotatable, power consumption can be reduced by ensuring that the azimuth thrusters only rotate when the supervisor switches the control mode from autopilot to dynamic positioning (or *vice versa*). Two thruster configurations were defined for each control mode. When in autopilot mode, the thrusters are oriented parallel to the longitudinal axis of the ASV. When in dynamic positioning mode, the thrusters are swivelled ±45° to either side of the longitudinal axis. The control allocation solution yields both the commanded thrusts for each thruster, and the achieved body forces and yawing moment provided by the solution ($\boldsymbol{\tau}^a$). The commanded thrusts are converted to a vector of electronic speed controller duty cycles ($\boldsymbol{d}$), which are outputted to the motor drivers for each respective thruster.

4 Energy Forecasting

In Section 2.2.1 a naive method was presented that calculated a survival metric based on the aggregated energy measurements. Through this method the operator safeguards against the future by providing a margin of safety on the survival thresholds, making the supervisor trigger recourse actions according to the likelihood that the vehicle has used the planned energy for a task, plan, or the battery. One issue with this is that the recourse decision is made at the time of the energy measurement and runs the risk of being activated too late. A forecasting approach could be used *in situ* to predict ahead of time if a survival threshold will be crossed, allowing the supervisor to initiate recourse actions earlier and save energy in the process.

Forecasting the power consumption of the vehicle is a three-stage forecasting process that requires:

1. Prediction of the vehicle's kinematics.
2. Prediction of the vehicle's control response for a given task and kinematic state.
3. Prediction of the vehicle's power consumption given the control response of the vehicle.

The following subsections detail a hybrid state prediction and control process, using data-driven machine learning models for the kinematic state and power consumption prediction, and classic PID control processes for the control response calculations. A simplified block diagram summarising the major stages of the process is provided in Fig. 7.

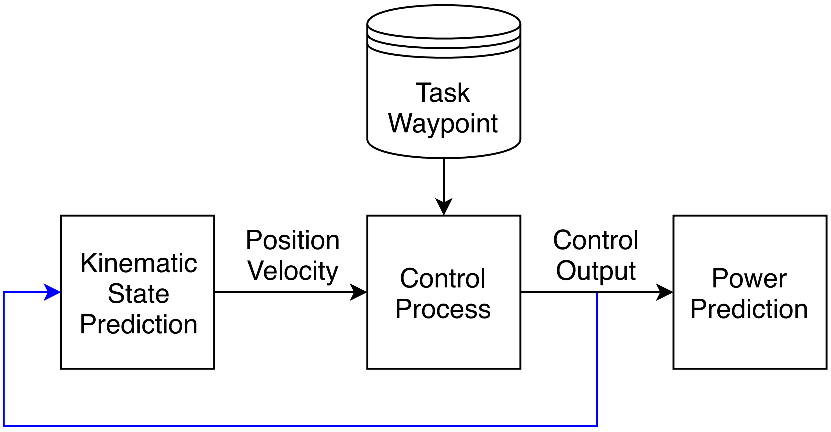


Figure 7: Simple view of the hybrid power prediction process. The first block predicts the new kinematic state of the vehicle given the control output as external feedback. The second block receives the new kinematic state and a waypoint reference from the task data and computes a control output. The third block receives the new control output and predicts the power consumption of the vehicle. The main feedback loop (i.e. setting $k = k + 1$) is coloured blue.

4.1 Kinematic State Prediction

The ASV is expected to operate in an environment without large rolling, pitching, or heaving motions, so the kinematics simplify to 3 degrees of freedom (two in translation, and one in rotation). The kinematic state of the vehicle is defined as:

$$\boldsymbol{\eta} = [x, y, \psi]^T \tag{30}$$

$$\boldsymbol{\nu} = [u, v, r]^T \tag{31}$$

where $\boldsymbol{\eta}$ is the vehicle's position in the inertia-frame. In this case the x and y coordinates are the vehicle's latitude and longitude projected into UTM, and ψ is the vehicle's heading relative to East following a East-North-Up convention. $\boldsymbol{\nu}$ is the ASV's body-frame linear velocities in the longitudinal (u) and transverse (v) directions and the angular turning rate in yaw (r). The vectorised kinetic model for marine vehicles is:

$$\boldsymbol{\tau}^a = \mathbf{M}\dot{\boldsymbol{\nu}} + (\mathbf{C}(\boldsymbol{\nu}) + \mathbf{D}(\boldsymbol{\nu}))\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) - \boldsymbol{\tau}^e \tag{32}$$

where $\boldsymbol{\tau}^a$ is the vector of achieved body-fixed forces outputted by the controller in longitudinal and transverse directions, and the commanded yawing moment. $\mathbf{M}$ is a matrix containing the summation of the added mass and inertia terms of the vehicle. $\mathbf{C}(\boldsymbol{\nu})$ is an array containing the summation of the rigid body and added mass Coriolis and centripetal terms as a function of $\boldsymbol{\nu}$. $\mathbf{D}(\boldsymbol{\nu})$ is a matrix of hydrodynamic damping terms (i.e. the drag coefficients) as a function of $\boldsymbol{\nu}$. $\mathbf{g}$ is the vector of hydrostatic forces according to the vehicle's position and orientation within the water column. $\boldsymbol{\tau}^e$ is the sum total of body-frame referenced environmental loads acting on the vehicle.

The vehicle's kinematic states are predicted according to evaluation of the kinetic model in Eq. (32).

This is a complex process as both the manoeuvring loads and environmental loads are non-linear and may not be able to be accurately observed *in situ*. The constant parameters in $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{D}$ can be identified through captive model testing of the vehicle, but must be updated if modifications to the vehicle (such as payloads) are introduced. Calculation of $\boldsymbol{\tau}^{\boldsymbol{e}}$ also requires sensing and observation of the environment (wind, current, and waves) which this ASV is not equipped to perform.

An alternative to model-based kinematic state prediction is to perform supervised learning on collected data to obtain an approximation of the Eq. (32). LSTM networks have seen success in learning predictions from time series data in other domains, making it a good candidate for predicting the trajectory and control time series data. The LSTM design in Fig. 8 performs the following operation:

$$\boldsymbol{\nu}_{k+1} = f(\boldsymbol{\nu}_k, \psi_k, \boldsymbol{\tau^a}_k) \tag{33}$$

where ψ_k and $\boldsymbol{\tau^a}_k$ are chosen as additional explanatory variables. Due to the internal feedback nature of the LSTM architecture, the network may also learn relationships between the change in and aggregate of $\boldsymbol{\nu}$ over time ($\dot{\boldsymbol{\nu}}$ and $\boldsymbol{\eta}$) but this is not guaranteed. $\dot{\boldsymbol{\nu}}$ is not included as the measured accelerations provided by the IMU are too noisy. The x and y components of $\boldsymbol{\eta}$ are also not included as they are not bounded by upper and lower limits (unlike ψ which is bound between $\pm\pi$ radians), which would make it harder for the LSTM to generalise.

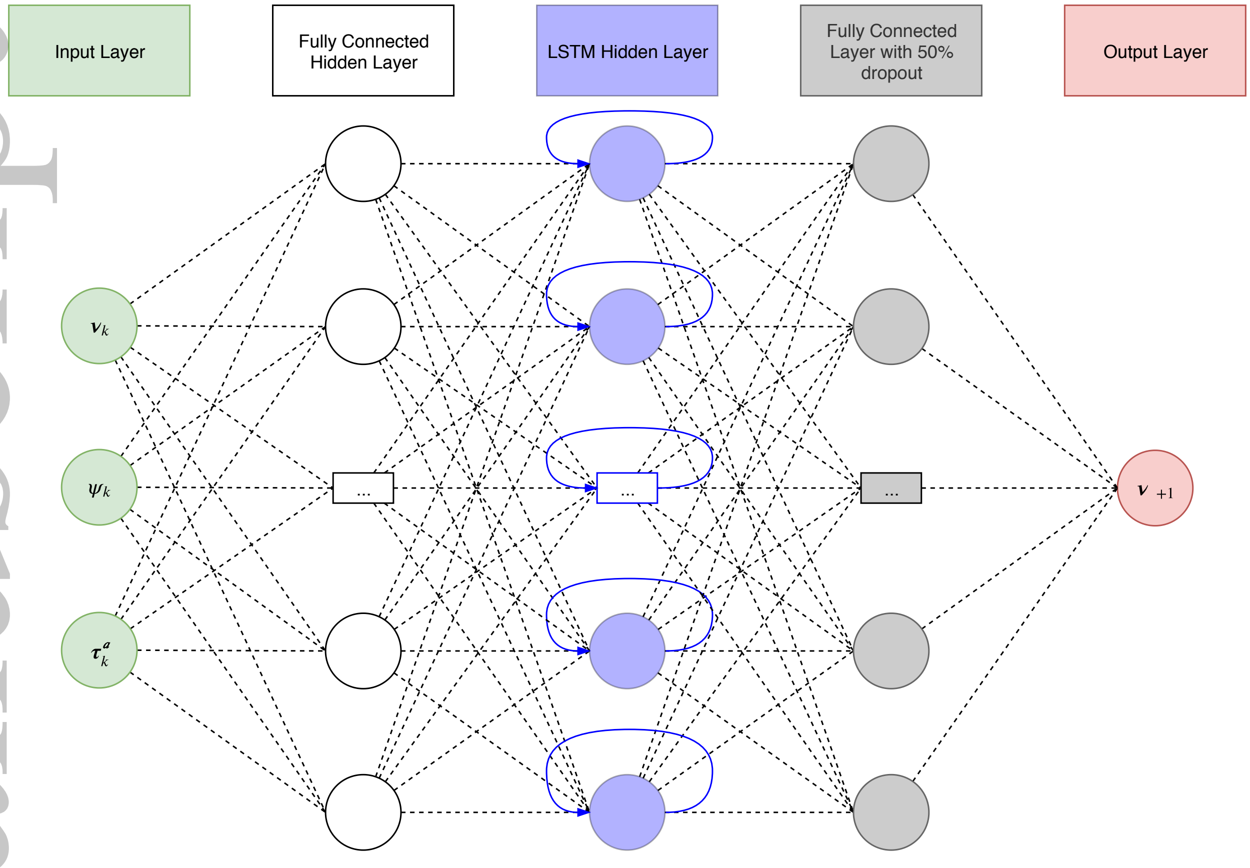


Figure 8: Network layout of the $\boldsymbol{\nu}$ state prediction LSTM. The network accepts the current body velocity vector $\boldsymbol{\nu}_k$, the current yaw heading ψ_k, and the achieved vector of body forces $\boldsymbol{\tau^a}_k$. The network consists of a latent space hidden layer, the LSTM layer, another fully connected layer with a 50% dropout to reduce the chance of overfitting, and then the output layer (which is $\boldsymbol{\nu}_{k+1}$).

4.2 Power Consumption Prediction

As discussed in Section 2.1, the power consumption of the ASV depends upon the commanded thruster outputs, the hotel load, and the efficiency of the electrical distribution system. Identification of each of these contributors can provide a good estimate of the total power consumption of the vehicle. However, complex transient effects that occur on the distribution system during changes in thruster output, and the effect of hydrodynamic loading on the thrusters during changes in output are not typically modelled. Once again, a data-driven approach using an LSTM network (see Fig. 9) may be able to capture these effects.

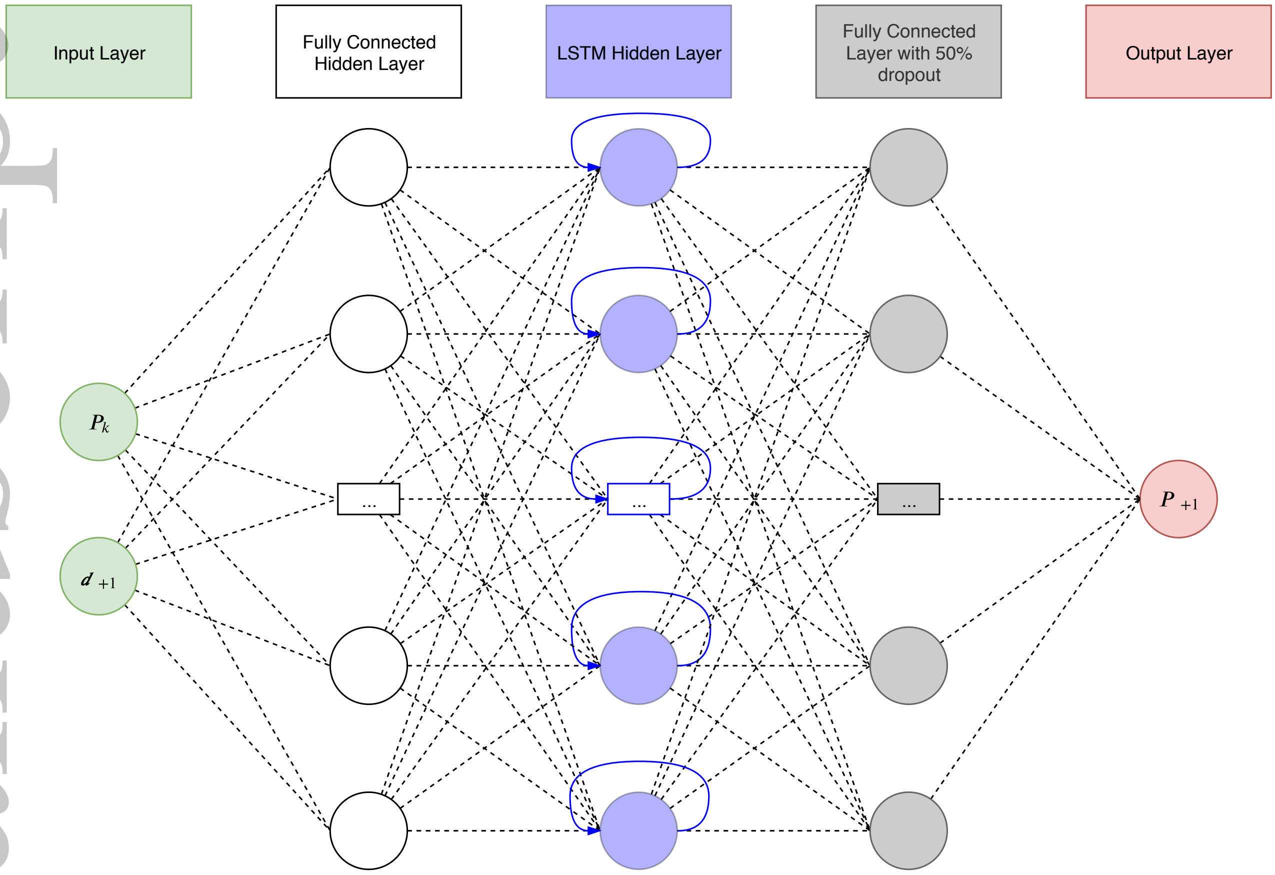

Input Layer
Fully Connected Hidden Layer
LSTM Hidden Layer
Fully Connected Layer with 50% dropout
Output Layer
P_k
d +1
...
...
...
P +1

Figure 9: Network layout of the power prediction LSTM. The network accepts the current power P_k and the predicted commanded duty cycles of the thrusters from the control allocation block ($\boldsymbol{d}_{k+1}$). The network consists of a latent space hidden layer, the LSTM layer, another fully connected layer with a 50% dropout to reduce the chance of overfitting, and then the output layer (which is P_{k+1}).

The power prediction network performs the following operation:

$$P_{k+1} = f(P_k, \boldsymbol{d}_{k+1}) \tag{34}$$

where $\boldsymbol{d}_{k+1}$ is the vector of commanded duty cycles for all thrusters as outputted by the control process prediction. Duty cycle was selected over force because duty cycle is bounded between 0 and 1, making it easier for the LSTM to generalise.

4.3 Hybrid Energy Forecaster Model

The kinematic state of the vehicle is used by the control process outlined in Section 3 to obtain the achieved body forces to regulate the kinematic state to specific set points that reach the task waypoint following the LOS guidance controller. Fig. 10 shows the full process of the hybrid energy forecast model. The $\boldsymbol{\nu}$ prediction LSTM first predicts the body-frame velocities of the vehicle at $k+1$. Then $\boldsymbol{\nu}_{k+1}$ is integrated and transformed into the inertia frame to obtain $\boldsymbol{\eta}_{k+1}$. The task waypoint, $\boldsymbol{\nu}_{k+1}$, and $\boldsymbol{\eta}_{k+1}$ are inputted into the LOS guidance controller to obtain the forward speed error, u^e_{k+1}, and the LOS heading error, ψ^e_{k+1}.

The autopilot controller functions as described in Eqs. (24) to (26). The gains for the controllers are copied from the gains used in the ASV control system. The autopilot controller outputs the vector of desired body forces and moments (${\boldsymbol{\tau}^{\boldsymbol{c}}}_{k+1}$), which is then inputted into a copy of the ASV's control allocation process. The control allocation process outputs the achieved body forces and moments, ${\boldsymbol{\tau}^{\boldsymbol{a}}}_{k+1}$, which is fed back into the $\boldsymbol{\nu}$ prediction LSTM. The duty cycles for the thrusters, $\boldsymbol{d}_{k_1}$, is also calculated by the control allocation process and is used by the power prediction LSTM to obtain P_{k+1}. ${\boldsymbol{\tau}^{\boldsymbol{a}}}_{k+1}$ is fed back into the $\boldsymbol{\nu}$ prediction LSTM for the next prediction cycle. The energy forecast can then be computed by aggregating the forecast power from Eq. (34) substituted into Eq. (11).

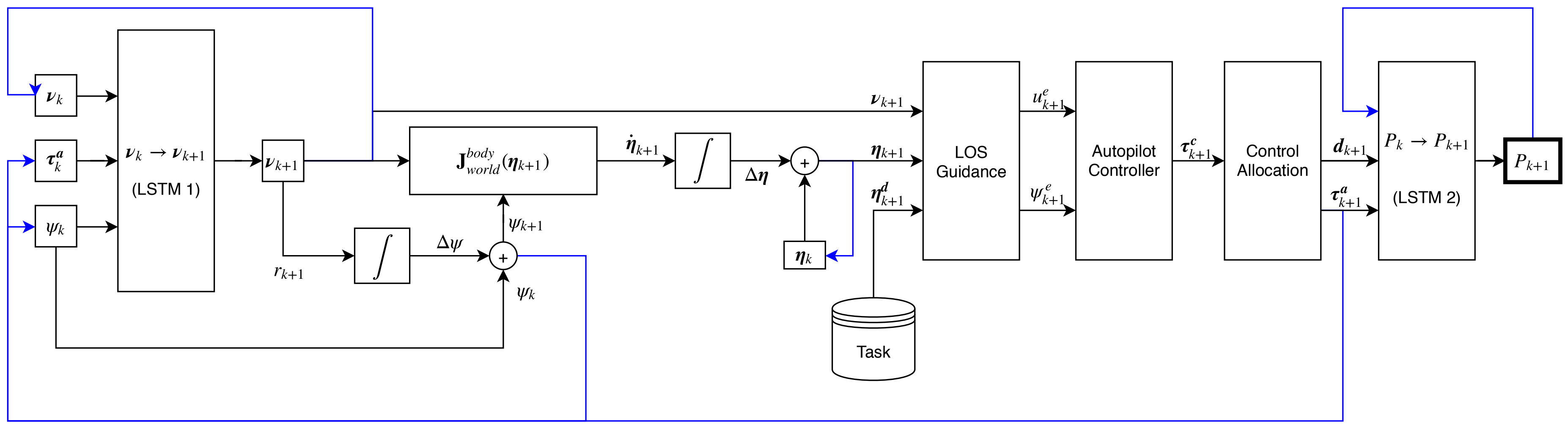
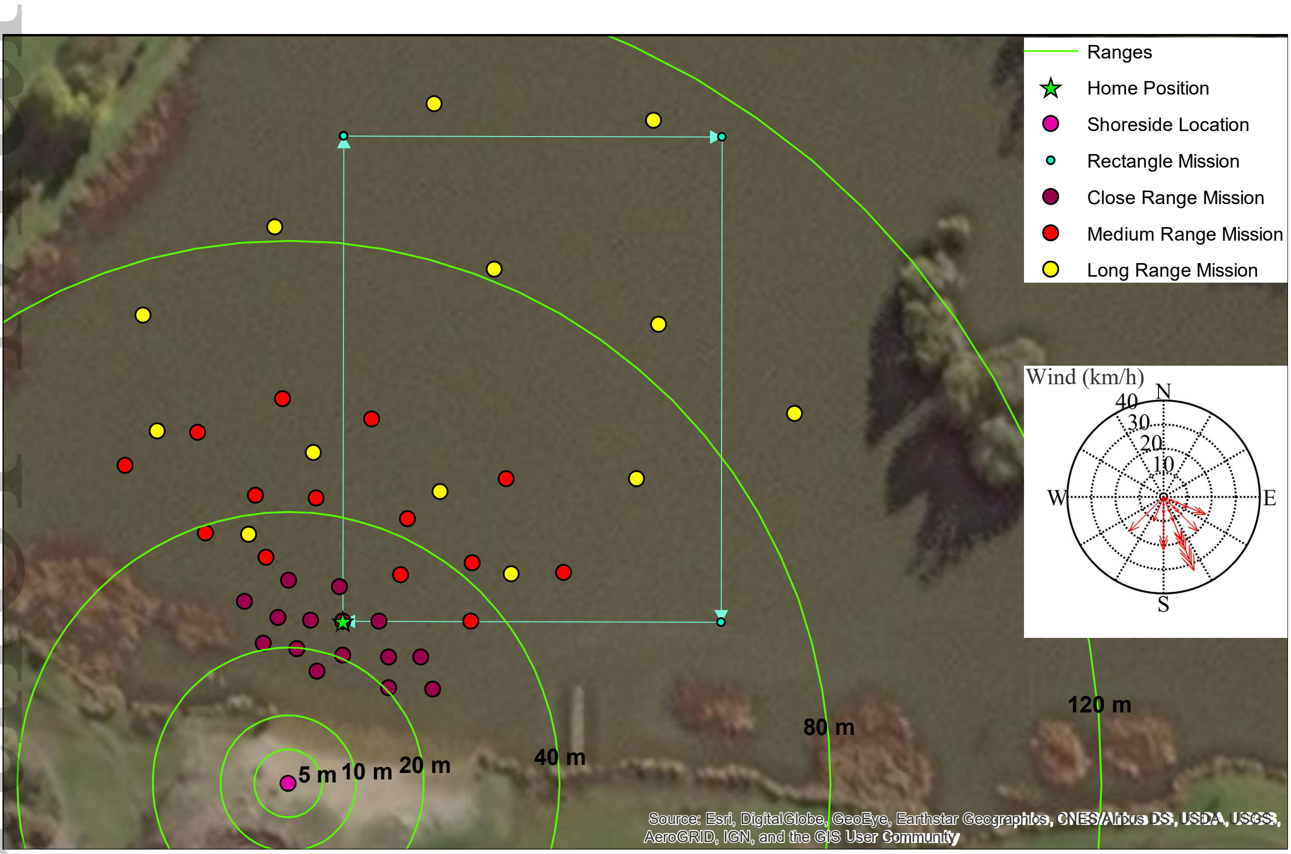


Figure 10: Full block diagram of the hybrid LSTM/control process power forecaster. The $\boldsymbol{\nu}$ state prediction LSTM produces $\boldsymbol{\nu}_{k+1}$ predictions which are then transformed to the inertia frame and integrated to obtain the vehicle's current position, a LOS guidance function calculates the setpoint for the forward speed controller and the relative bearing error, which feeds into the autopilot controller. The autopilot outputs a commanded body force and moment vector $\boldsymbol{\tau^c}_{k+1}$ to the control allocation algorithm, which determines the duty cycle for each thruster ($\boldsymbol{d}_{k+1}$). Finally the power prediction LSTM uses the power from the current step P_k and the predicted thruster duty cycle $\boldsymbol{d}_{k+1}$ to predict P_{k+1}. The blue lines indicate the feedback paths where k is set to $k+1$ for the next time step.

5 Results of Field Trials and Energy Forecasting

Fig. 11 details the mission data sets that were generated within a Geographic Information System package. The close, medium, and long range missions were specified as collections of waypoints with no specific order ($\mathcal{T}$ from Section 2.1). The rectangle mission has a human-defined order as it is a calibration test for the ASV's autopilot controller. Each mission was run several times with varying δ_p and δ_t thresholds, which are the primary criteria used by the supervisor for making *in situ* decisions. Weather data was obtained from Bureau of Meteorology (2018) during post-processing of the data. Section 5.1 presents two example runs where the behaviour of the vehicle is drastically different due to the manipulation of these thresholds.



Source: Esri, DigitalGlobe, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AeroGRID, IGN, and the GIS User Community

Figure 11: Summary of mission datasets generated for Lake Waverley. All missions except for the rectangle mission are not given a sequence by the operator. The mission planner creates a sequence during plan generation. Wind data was obtained from a local weather station 5 km from the lake, providing twice-daily measurements.

The trials were conducted with the ASV and shoreside systems over four days at Lake Waverley (Fig. 12), and a total of 47 mission runs were recorded. During trials, the effective Wi-Fi range of the shoreside router was identified to be between 50-60 m. This interfered with the replan recourse action for the long range mission, resulting in the supervisor initiating the return home recourse action if it was out of contact with the shoreside planner when δ_p was reached.

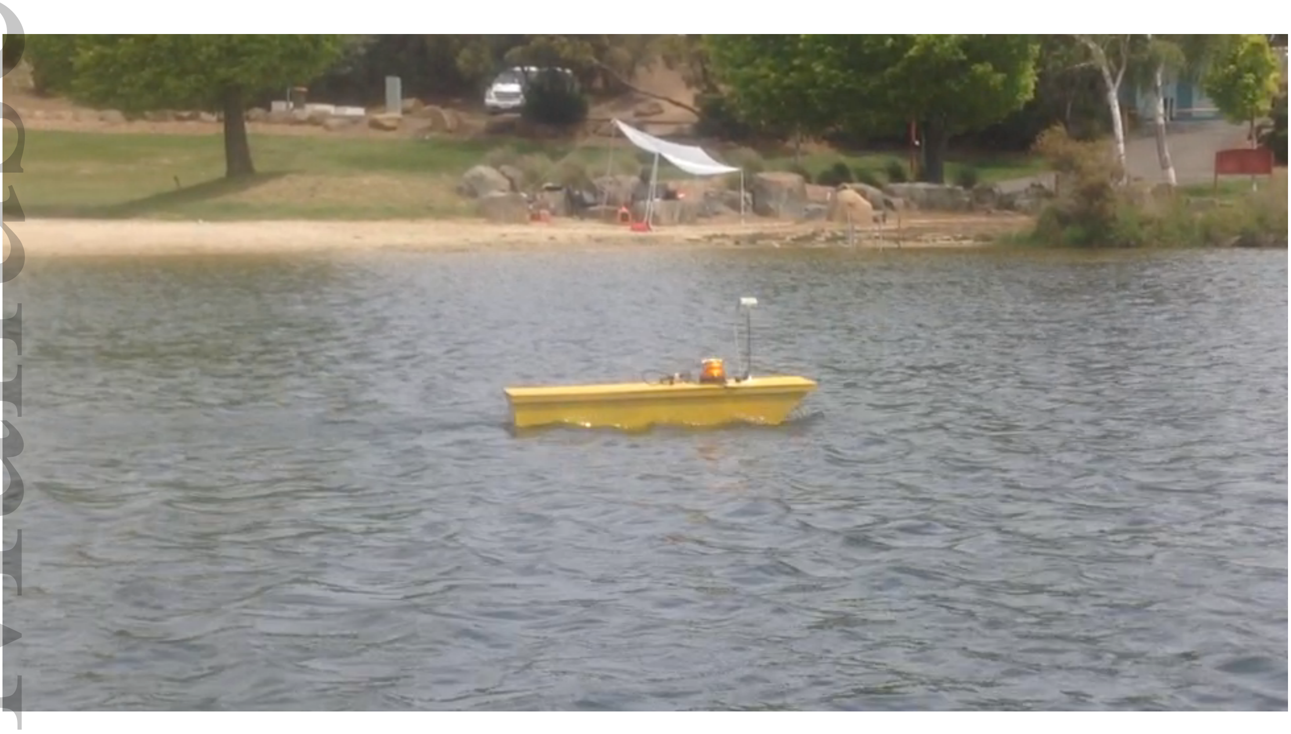


Figure 12: The ASV underway on a waypoint following task in Lake Waverley. Shoreside system setup in background.

Significant and unmeasured local gusts occurred during the course of the trials, which affected the control of the vehicle (see Fig. 13 as an example). Weather data from Bureau of Meteorology (2018) provides two reference measurements for the wind behaviour for each day of the trials. Additionally, the magnetic interference from the DC motors driving the azimuth thrusters would intermittently affect the magnetometer sensors within the IMU. This would lead to the vehicle's guidance system calculating incorrect LOS heading errors, and in the worst case causing the vehicle to circle a waypoint indefinitely. These challenges were overcome by adjusting the K_i gains of the PID controllers to be more aggressive, and by relocating the IMU towards the GPS mast. To correct the compass misalignment fault the mission had to be paused and manual override commands were given to the vehicle to align its heading with an external measurement of magnetic north so that the navigation system's compass datum could be reset. The effects of both the wind and the magnetic interference will also be important in Section 5.2.

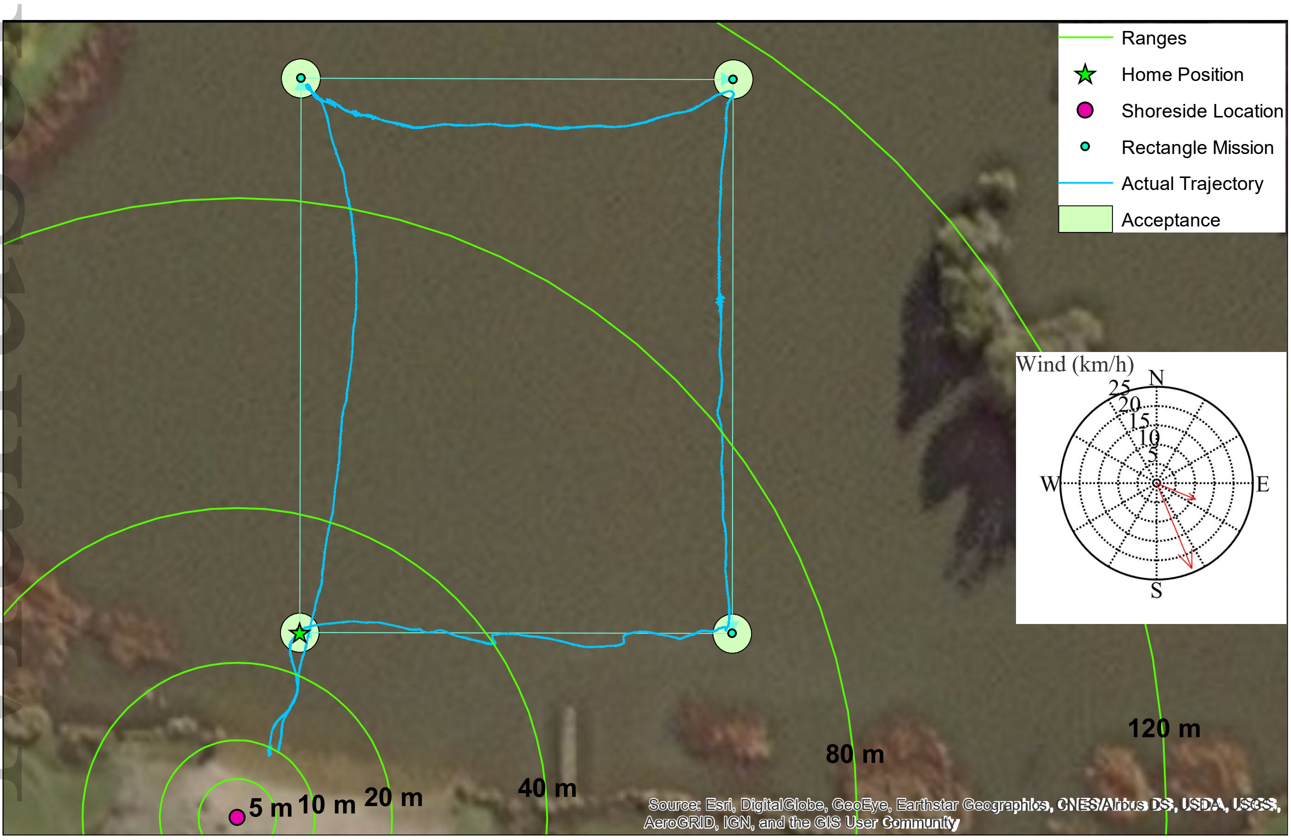
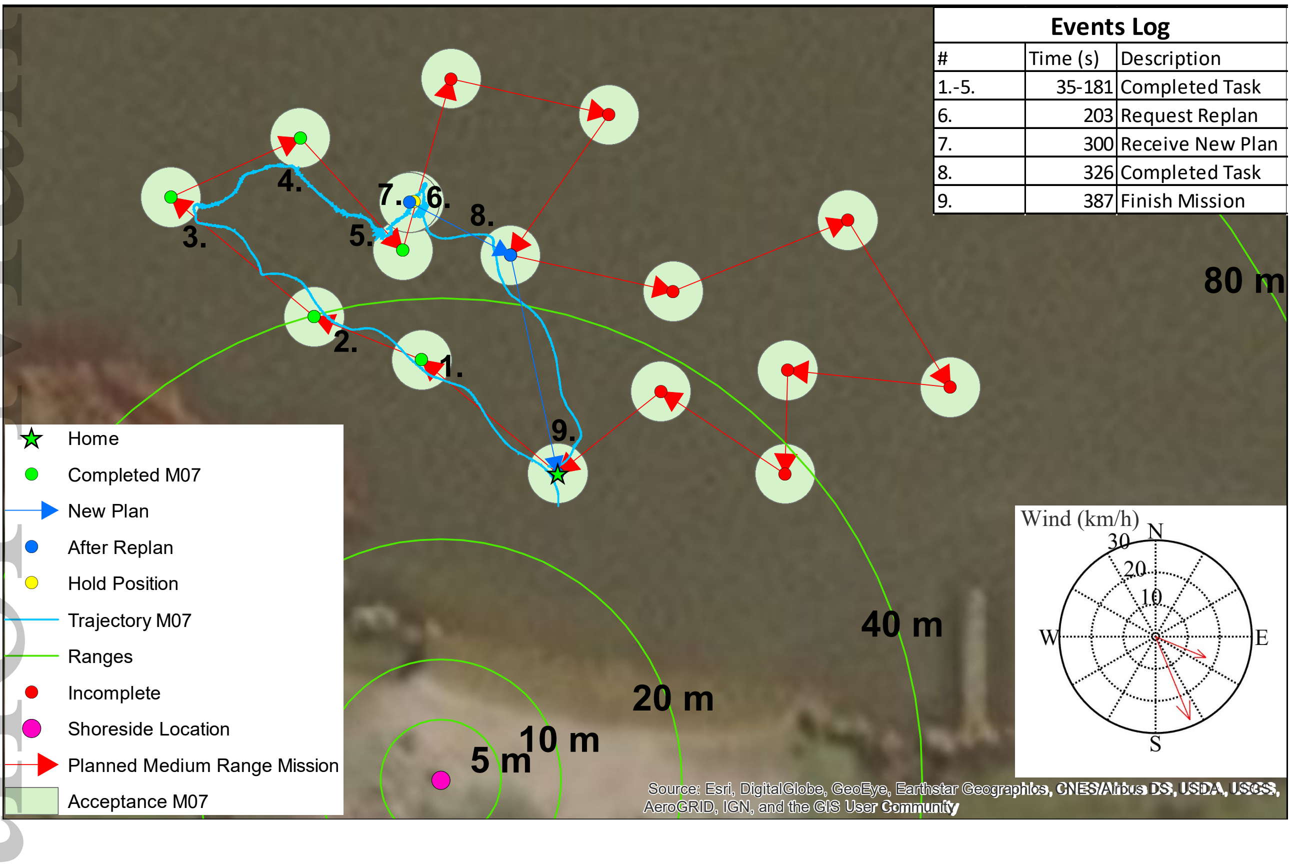


Figure 13: One of the rectangle missions performed by the ASV. Strong and intermittent North-Westerly gusts blew the vehicle sideways from the expected trajectory, leading to trajectories with cross-track errors that are sometimes large and sometimes small. The autopilot controller does not control motion in the sway direction, leading to arcs in the achieved trajectory.

5.1 Recourse Action Effects

Adjusting the δ thresholds for each of the survival functions produces significantly different behaviours from the supervisor. Combinations of different thresholds result in behaviours where it is difficult to identify which threshold contributes what to the result. To see the direct effects of δ_p and δ_t as they operate independently, this section looks at two runs of the medium size mission: one where δ_p is high and δ_t is off (i.e. < 0), and one where δ_t is high and δ_p is off.

Fig. 14 presents a run of the medium size mission with $\delta_t = -1\text{e}10^{-9}$ and $\delta_p = 0.85$. From the trajectory of the ASV, it appears that the ASV attempts to finish the waypoint tasks in the order specified by the planner. When the threshold is crossed, the ASV performs a hold position manoeuvre and requests a new plan from the shoreside planner. The new plan budgets in one waypoint task before sending the vehicle to the home position. This behaviour is similar to the return home recourse actions described in Evers et al. (2014) and Shang et al. (2016), but is capable of planning in tasks that are along the route home.



Events Log

#	Time (s)	Description
1.-5.	35-181	Completed Task
6.	203	Request Replan
7.	300	Receive New Plan
8.	326	Completed Task
9.	387	Finish Mission

Figure 14: Progression of medium range mission 7. $\delta_p = 0.85$ and $\delta_t = -1\text{e}10^{-9}$.

Fig. 15 presents the corresponding time series of battery, plan, and task survival function data for medium mission 7. The vehicle fell out of contact with the mission planner after it had successfully made a replan request, leading to a long hold position action. Once contact was reestablished, the vehicle proceeded on the new mission and completed it without triggering a new replan recourse action. The task survival function has drastic falling edges after a few seconds of performing a new task. This indicates that the planner is underestimating the energy cost of performing tasks, and also that the calculated H_t distributions have a small variance.

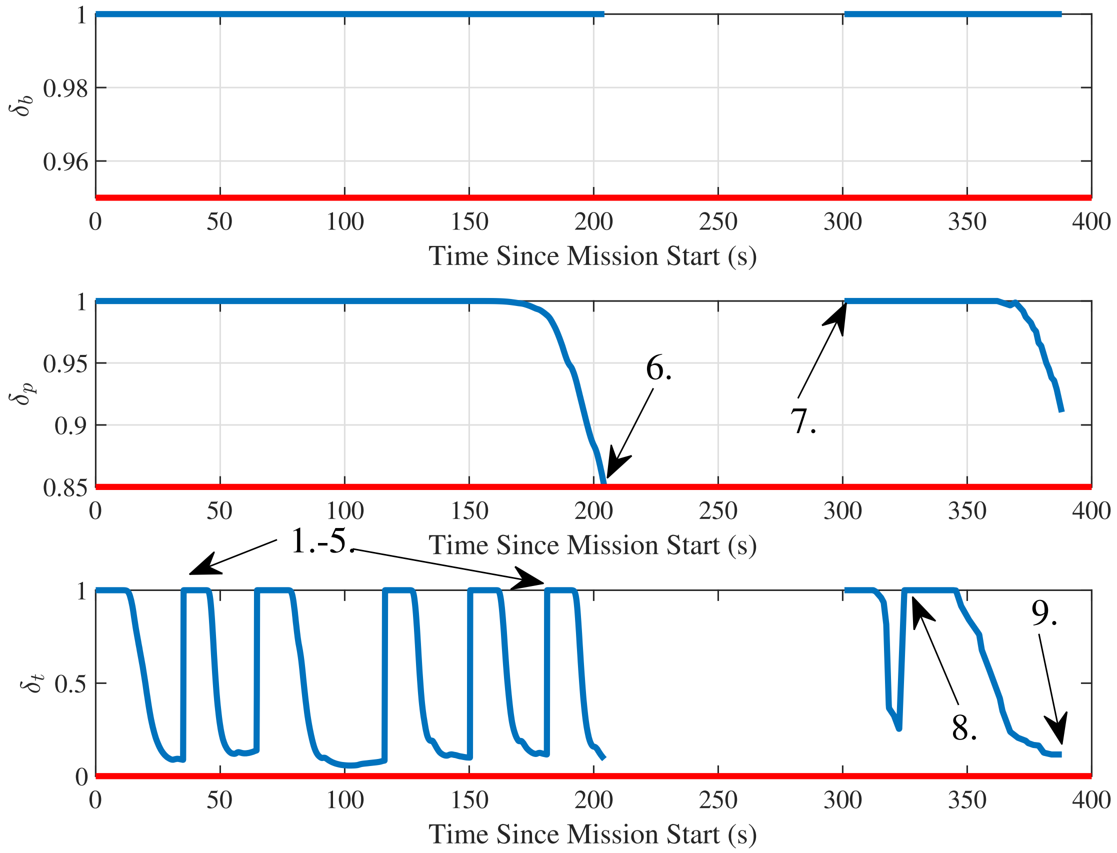
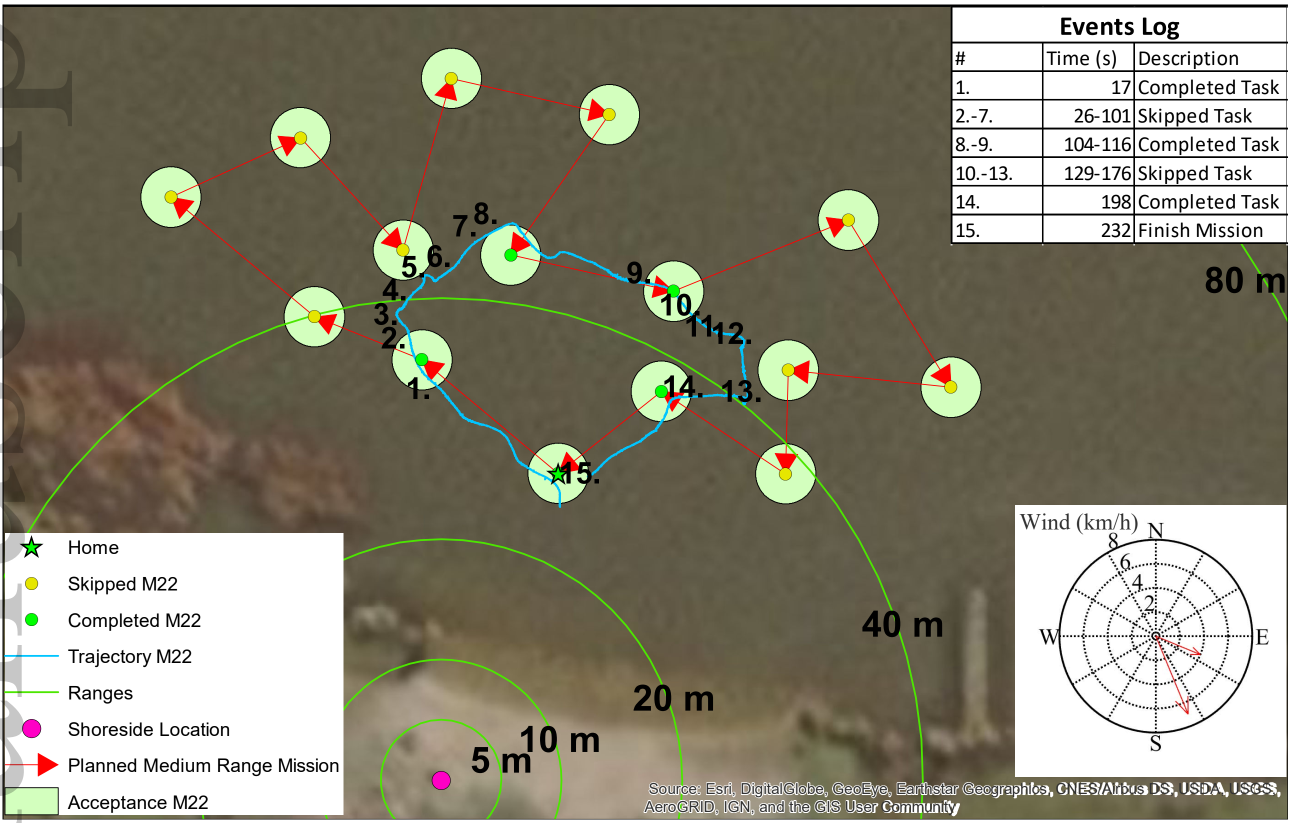


Figure 15: Survival functions of medium range mission 7. The numbered arrows correspond to the events log in Fig. 14. Task survival (δ_t) indicates that the generated plan is underestimating the power consumption of the vehicle. The gap in time (between 200-300 s) is because the survival functions are not considered by the supervisor while holding position.

As a basis of comparison, Fig. 16 presents run 22 of the medium range mission where $\delta_t = 0.5$ and $\delta_p = -1\mathrm{e}10^{-9}$. The trajectory shows that the supervisor has a very short tolerance on pursuing a task before skipping it and moving on to the next. The vehicle was only able to complete tasks where its distance ended up being significantly closer than what was originally planned for it by virtue of the task skip heuristic.



Events Log

#	Time (s)	Description
1.	17	Completed Task
2.-7.	26-101	Skipped Task
8.-9.	104-116	Completed Task
10.-13.	129-176	Skipped Task
14.	198	Completed Task
15.	232	Finish Mission

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.
80 m
40 m
20 m
10 m
5 m
Home
Skipped M22
Completed M22
Trajectory M22
Ranges
Shoreside Location
Planned Medium Range Mission
Acceptance M22
Wind (km/h)
N
E
S
W
2
4
6
8
Source: Esri, DigitalGlobe, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AeroGRID, IGN, and the GIS User Community

Figure 16: Medium range mission 22 progression. $\delta_t = 0.5$ and $\delta_p = -1\mathrm{e}10^{-9}$.

The survival functions for mission 22 are slightly different from mission 7. Fig. 17 shows that the task survival function began decaying almost immediately after the task was initiated. This is likely due to the more aggressive PID tunings and control output limits that were used on this run compared to mission 7. The controller outputted higher thrust commands to the thrusters, which results in much higher power consumption. The plan survival function also finishes quite close to 0.5, indicating that the vehicle finished the mission close to the expected energy consumption of the plan.

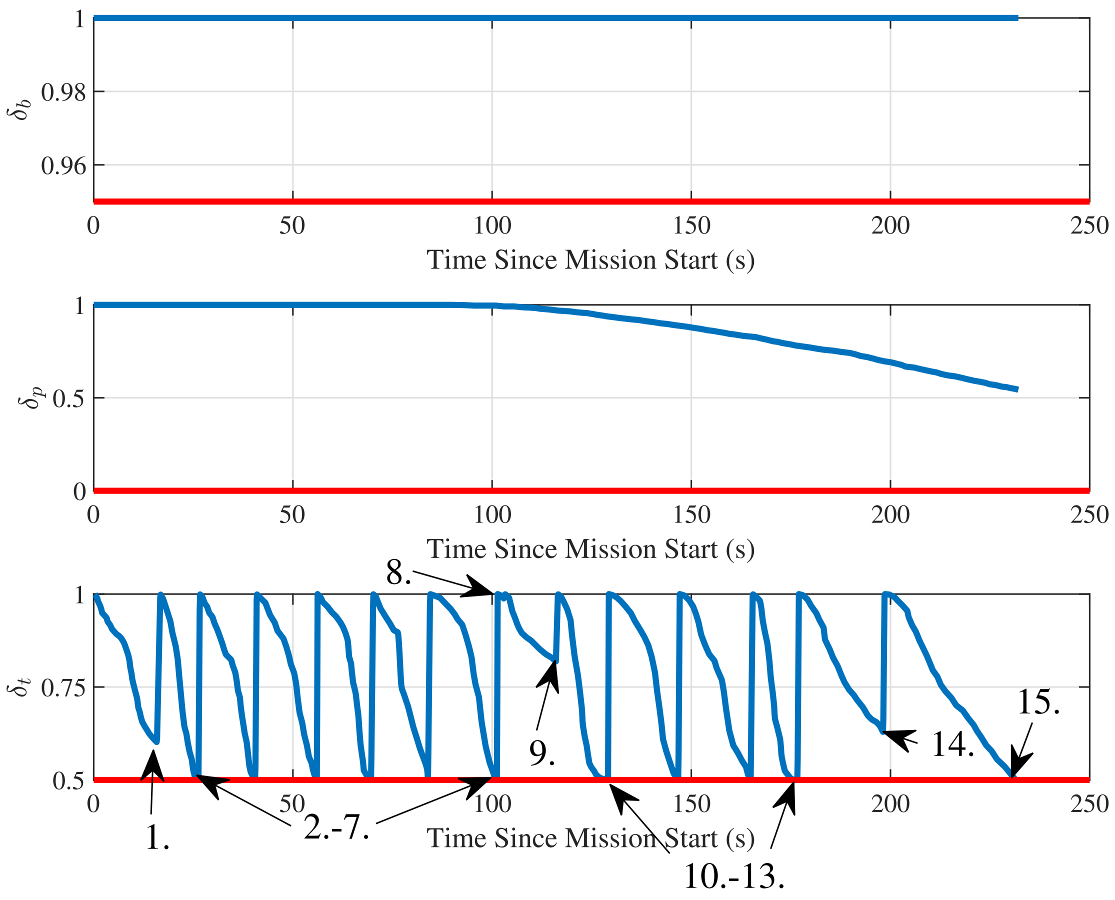


Figure 17: Survival functions of medium range mission 22. The numbered arrows correspond to the events log in Fig. 16. Task survival (δ_t) indicates that the generated plan is underestimating the power consumption of the vehicle.

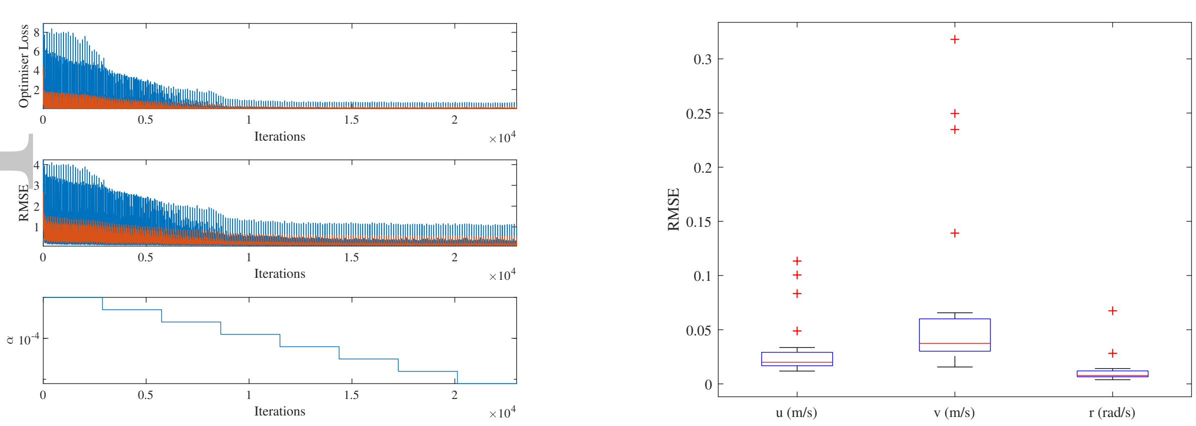
Comparing the above missions, it appears that both strategies are regulating the total energy consumption of the vehicle with respect to the plan energy consumption expectation. In mission 7, the high δ_p threshold means that the supervisor makes conservative responses (i.e. it keeps to the schedule, and returns home early, visiting tasks along the way). Mission 22 shows that a high δ_t results in very quick jumps ahead in the schedule, leading to a more rounded coverage of the mission area. In either case, the results could have been improved with energy estimations that also included environment effects.

5.2 Forecaster Evaluation

5.2.1 Training

The data collected during the Waverley trials was used to train the hybrid energy forecaster model. During each mission the ASV switches between several operating modes: manual RC, autopilot, dynamic positioning, and idle. The power prediction LSTM was trained on data from each operating mode, whereas the $\boldsymbol{\nu}$ prediction LSTM was trained on the subset of the data where the vehicle was operating in autopilot mode. The choice to only evaluate the $\boldsymbol{\nu}$ LSTM on only the autopilot data was made to simplify the controller component of the hybrid energy forecaster model. Forecasting the control input of an operator's RC command is counterintuitive, and forecasting the power consumption of a vehicle in idle mode simplifies to forecasting the hotel load. The controller component could be expanded to include the dynamic positioning controller in the future.

The data was split 9:1 between training and test data sets. Fig. 18a presents the training curve for the $\boldsymbol{\nu}$ prediction LSTM. Fig. 18b presents the Root-Mean-Squared Error (RMSE) of the trained $\boldsymbol{\nu}$ prediction LSTM in performing one-step predictions on the test data sets. The performance on the test set indicates that the error in position between the predicted and actual will grow due to the integration of the velocity error.



(a) $\boldsymbol{\nu}$ Training Curve.

(b) $\boldsymbol{\nu}$ Test RMSE Results.

Figure 18: Training performance of the $\boldsymbol{\nu}$ prediction LSTM (Fig. 18a) and the one-step prediction performance of the trained $\boldsymbol{\nu}$ prediction LSTM on the test data set (Fig. 18b). The top and middle plots of Fig. 18a are the optimisation loss and standardised RMSE (including a red moving mean trend). The bottom plot is the learning rate, which is progressively halved during training.

5.3 Forecasting Kinematic State

To evaluate the forecast effectiveness, the trained predictor is first 'charged' with feed-in data from the test set. The length of the feed-in data does effect the quality of the forecast as the LSTM will have more valid historical data available, but a trade-off has to be made with computational resources (i.e. more memory is needed to store longer feed-in). The first 5 seconds of each of the test input sets were reserved for feed-in (we found 5 seconds to be sufficiently long enough for the forecaster to produce accurate forecasts).

Fig. 19 presents the error in distance-to-target between the actual and forecast for each autopilot task (each coloured line representing a task). For positive $D(t) - \hat{D}(t)$, the forecasted vehicle position is closer to the target than actual (optimistic forecast) and *vice versa* for negative $D(t) - \hat{D}(t)$ (pessimistic forecast). The error in $\boldsymbol{\nu}$ between actual and forecast is integrated, which increases the rate of error growth in forecasted $\boldsymbol{\eta}$. This in turn affects the guidance module, which will use a different distance-to-target to calculate the LOS heading error, $\psi_e(t)$ (displayed in Fig. 20).

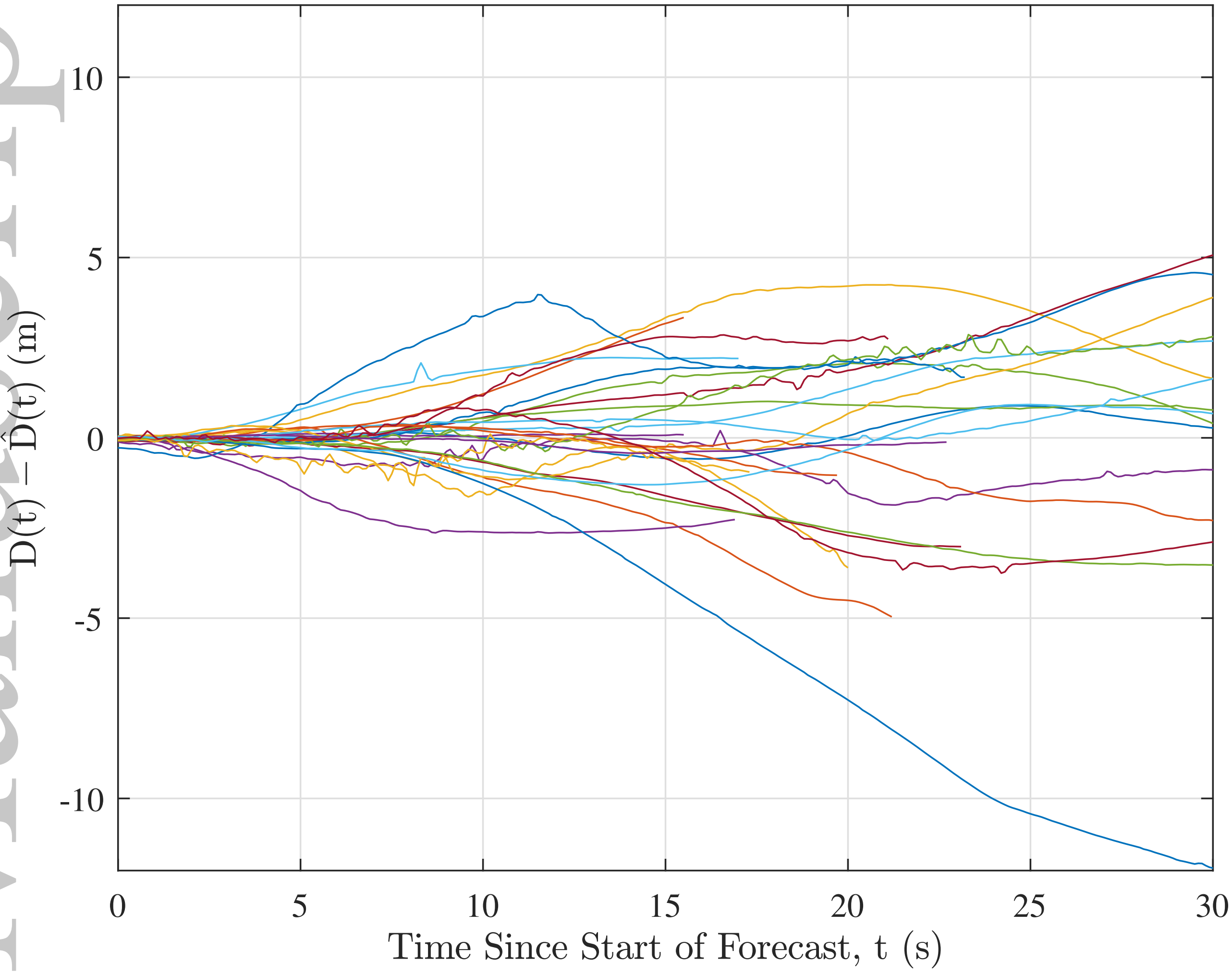


Figure 19: Error between actual and forecast distance to target over a 30 s window. The $\boldsymbol{\nu}$ LSTM and following autopilot controller were fed with 5 seconds of ground truth data before forecasting began.

In general, both the distance-to-target and the LOS heading error between forecasted and actual are reasonably small for the first 5 seconds of forecasting. Some of the significant errors in Fig. 20 are because of strong wind gusts that occurred after forecasting began (i.e. the feed-in data did not indicate enough information about the wind effects to the LSTM). In this respect we suggest extending the kinematic LSTM by including wind speed and direction measurements as additional input variables. This will allow the trained LSTM to learn a model for the estimation of τ_{wind}, improving the accuracy of the prediction. Including the next-step wind measurements as output variable would allow the LSTM to make forecasts on future wind speed and direction. Reliable and frequent wind speed and direction measurements must be obtained through an on-board anemometer for this approach to be possible (data from weather stations may be low-frequency or not entirely representative of the local conditions). Although the training and test data were filtered for any run segments that contained identifiable magnetic interference to the IMU (described in the introduction of Section 5), imperceptibly small compass errors cannot be discounted from the data and may not be consistent enough for the LSTM to incorporate.

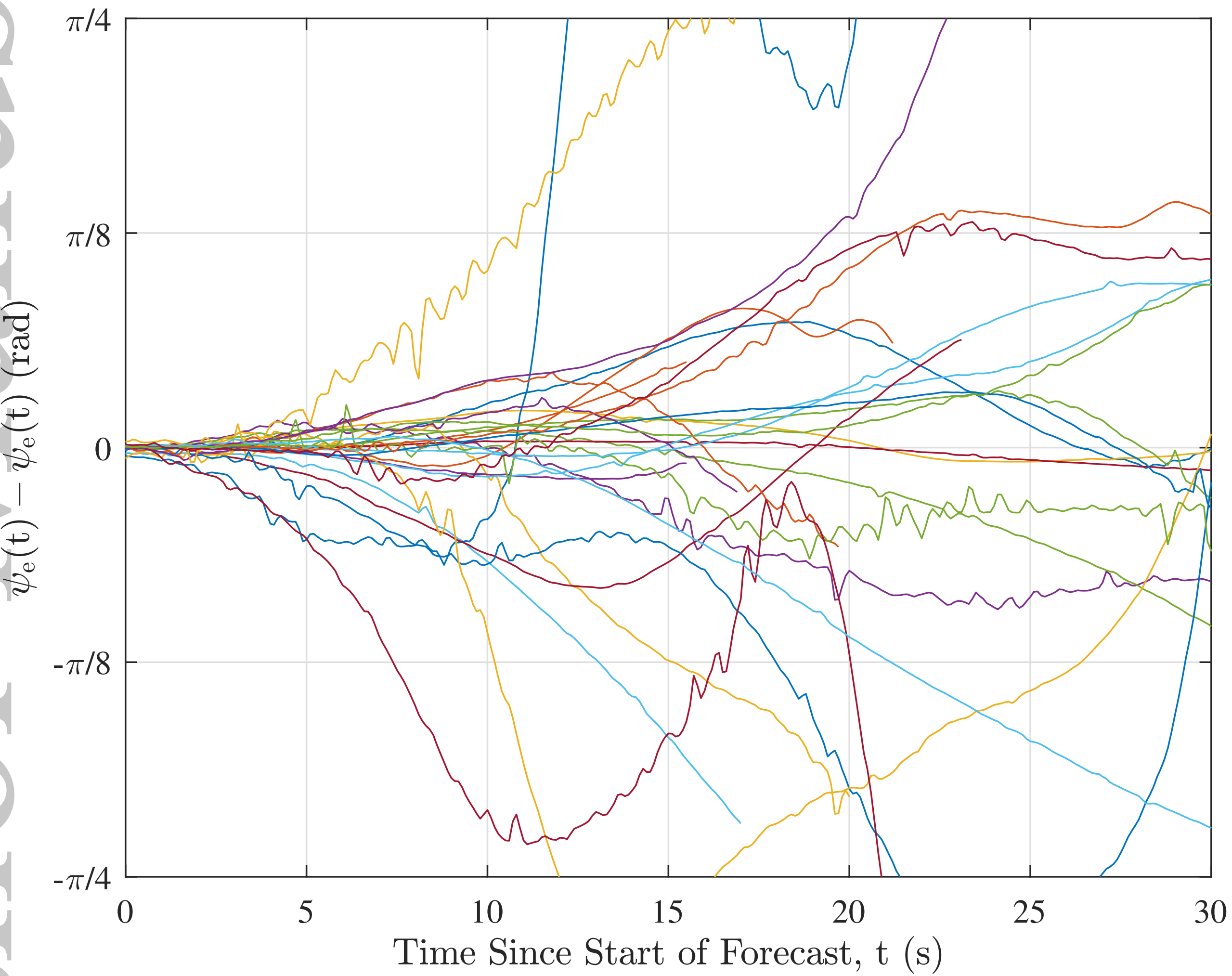
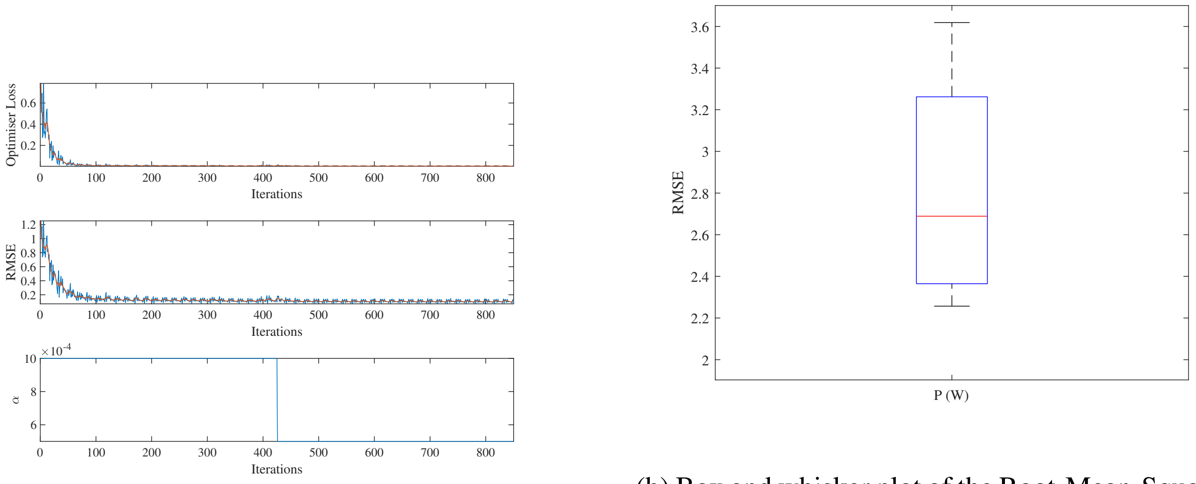


Figure 20: Error between actual and forecast LOS heading error. The $\boldsymbol{\nu}$ LSTM and following autopilot controller were fed with 5 seconds of ground truth data before forecasting began.

5.4 Forecasting Energy

Fig. 21a presents the training curve for the power prediction LSTM. Fig. 21b presents the Root-Mean-Squared-Error of the trained power prediction LSTM in performing one-step predictions on the test data sets. The power prediction timeseries for the actual and forecast are integrated to obtain the energy aggregate for each task, which is used for the rest of the forecast performance investigation.



(a) Power training curve with optimiser loss (top), RMSE (middle) and learning rate α (bottom).

(b) Box and whisker plot of the Root-Mean-Squared-Error between the test data and the power one-step prediction of the LSTM.

Figure 21: Training and one-step prediction test results for the power prediction LSTM.

As soon as the forecast begins there is an error between the actual and forecast. To properly evaluate the forecaster, some measurement for determining when the error is too large to be reliable (i.e. the reliable forecast horizon) is required. A suitable candidate is the confidence interval of the planner-generated energy distribution as this is based on what the Monte Carlo simulation and Gaussian approximation produces for the given task. The 95% and 50% confidence offsets (i.e. $1.96 \pm \sigma(k)$ and $0.67 \pm \sigma(k)$) were chosen as standard reliability thresholds for the forecaster. The confidence offsets were calculated by obtaining E_σ for a given M_o and plotting it with the total expected duration of the task transition. Fig. 22 shows the 95% ($1.96E_\sigma$) and 50% ($0.67E_\sigma$) confidence offsets of the distributions for the Monte Carlo simulated energy calculation of the all the potential tasks in the midsize mission from Section 5.1.

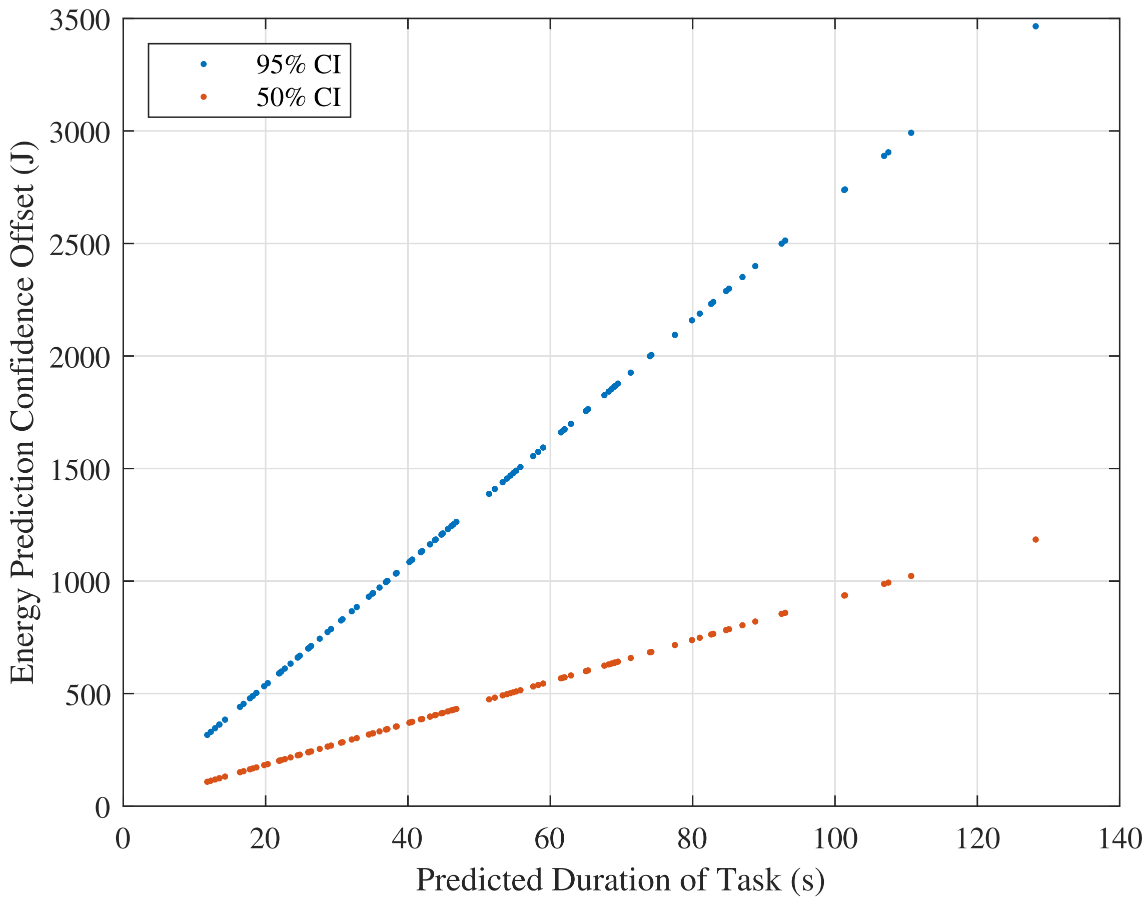


Figure 22: 95% and 50% confidence offsets of the predicted energy distribution for all possible tasks that are available in the midsize mission. Distributions were calculated from 10,000 simulations of the vehicle dynamics model.

As can be seen from the growth of the confidence intervals, the standard deviation of the approximated Gaussian distribution for each autopilot manoeuvring task increases linearly with the duration. This is because each trajectory that the planner is considering for each task is a straight line and assumes constant forward velocity of the vehicle. For missions where there are obstacles that cause the trajectory to be curved, or that have changes in velocity the linear trend will not be as clear.

The lines that fit the 95% and 50% confidence offsets show how the confidence of the Monte Carlo simulation is expected to decrease (spread) as the task length becomes longer. By comparing the spread of these confidence lines with the prediction interval spread of the forecast error, the time at which the forecaster prediction confidence exceeds the expected confidence can be identified. This time is indicative of when the forecast becomes less reliable than the model prediction made by the mission planner and is our definition of the reliable forecast horizon.

The full hybrid energy forecaster (see Fig. 10) was evaluated on test data using autopilot ”move to position” task data only (28 different tasks in total). The ν and power prediction LSTMs were ’charged’ with a feed-in of the ground truth data. The forecaster was evaluated over several trials with feed-in lengths increasing in 1 s increments from 1 s to 15 s. Fig. 23 presents the error between the actual energy consumption and the forecast energy consumption with a feed-in length of 5 s.

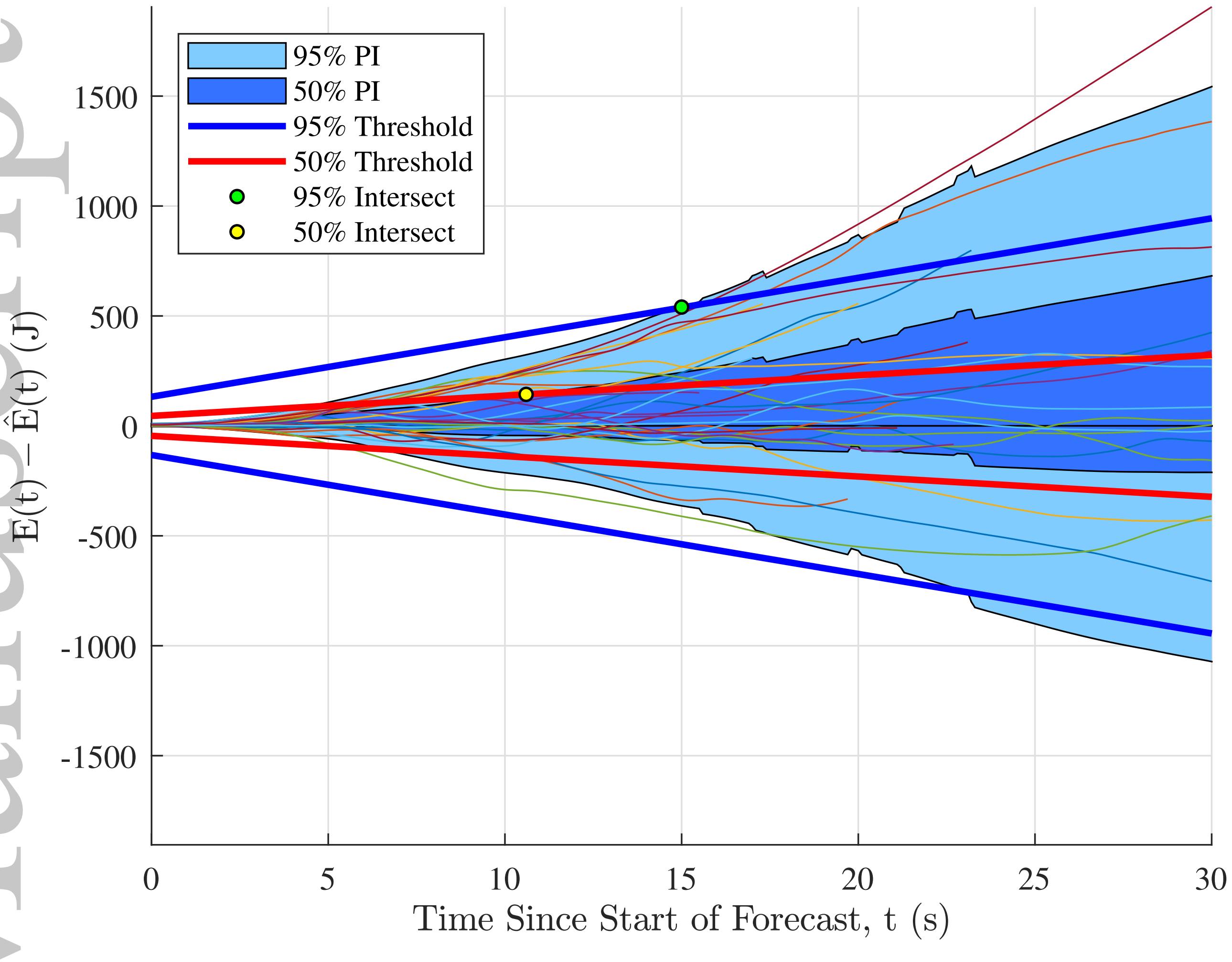


Figure 23: Error between actual and forecast energy consumption. Power prediction LSTM was provided with 5 s of ground truth power/duty cycle data before forecasting began.

Each run of the test data is represented by the thin coloured lines. The lighter shaded area is the 95% confidence interval of the forecast, and the darker shaded area is the 50% confidence interval of the forecast. What can be seen immediately from Fig. 23 is that the error is biased towards the positive and increases over time. This means that the forecaster has a higher chance of forecasting an underestimate of the required energy for any horizon. The green marker indicates when 95% of the energy error has reached the 95% confidence interval of the modelled energy prediction from the planner. The yellow marker indicates the same for a 50% confidence interval intersection. A reliable forecast horizon can be determined through these intersections. For example, if the criteria for a reliable forecast horizon was to be 95% sure that the forecaster error is within 95% of the predicted energy distribution, then the time at the green marker (approximately 15 s) would be an appropriate horizon.

A slightly more conservative approach would be to consider the absolute energy error as in Fig. 24. This brings the point at which it is 50% certain that the forecaster error is within 50% of the predicted energy distribution forward to be earlier in time than what is calculated in Fig. 23. However, information about whether the forecaster is optimistic or pessimistic is lost as the absolute energy error always assumes that the forecaster is optimistic. Additionally, the point at which it is 95% certain that the forecaster error is within 95% of the predicted energy distribution is pushed back in time compared to Fig. 23. For these reasons we recommend assessing the reliability of the forecaster according to the positive/negative energy error data as in Fig. 23.

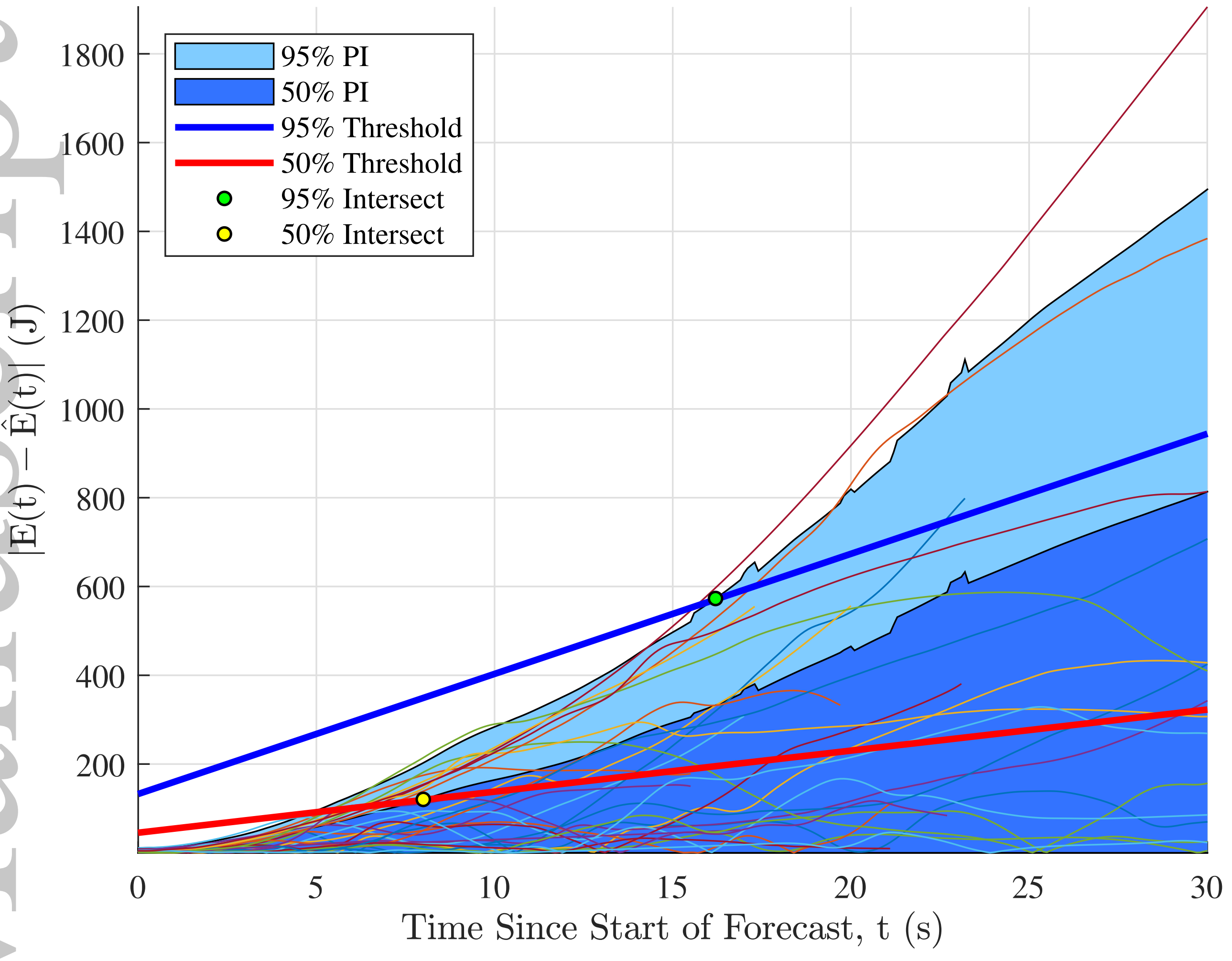


Figure 24: Absolute error between actual and forecast energy consumption. Power prediction LSTM was provided with 5 s of ground truth power/duty cycle data before forecasting began.

6 Discussion and Recommendations

Three components of energy-based planning have been described: the development of energy planning for AMVs as an extension of researched solutions to the OP-SW; the development of an ASV system capable of realising and testing the two-stage energy planning approach; and *post hoc* analysis of the vehicle's kinematic response and energy consumption with the goal of developing an energy forecasting solution. These three components contribute to the goal of producing a mission planner that can safely plan under uncertainty, can deal with communication dropouts between the vehicle and the operator, and can reliably forecast future demand for the purposes of online mission supervision and decision-making for the AMV.

In Section 5.1, we presented two different runs of the same mission plan where the supervisor was tuned with different survival thresholds. In the first run, the supervisor ignored the realisations of each task leg and focussed on the total aggregated energy. The ASV stayed on its predetermined mission plan until the supervisor was 85% sure that its total consumed energy for the mission had not exceeded the predicted energy consumption. At this point the ASV deviated from its expected plan and requested a new plan from the shoreside system (notifying the operator in the process). The supervisor demonstrated a conservative policy, making it easier for the operator to be sure of what the vehicle was doing (as it was likely to be doing what was initially determined by the first stage planner). From a geographical standpoint, (i.e. if this test mission's goal was to survey an area), the operator can only be confident that the Western half of the area was adequately surveyed.

In the second run, the supervisor was tuned with survival thresholds that focussed on the realisations of the task leg and ignored the total aggregated energy. This resulted in the ASV only visiting targets that were achieved **before** the supervisor was more than 50% sure that the energy consumed for the task was greater than the predicted. Tasks that took longer than expected were skipped, making this version of the supervisor exhibit opportunistic behaviour. From the operator's perspective, it is a lot harder to predict if the supervisor will skip a task or complete it. However, the surveyed area is much more evenly distributed around the vehicle's rendezvous point.

There are many other potential behaviours that could be exhibited by the supervisor by simply using different combinations of survival thresholds. Given the permutations, observing how each combination will affect the supervisor's reaction to the realised environment in an exhaustive fashion isn't feasible within the practical limitations of physical field experiments. The next step for evaluating this planner will be through simulation, using OP-SW testing data to properly quantify the mission outcomes (i.e. how many tasks were completed, did the vehicle return home within energy constraints, etc.) for many combinations of battery, plan, and task survival thresholds.

The purpose of the field trials was to evaluate how the planner behaves when exposed to unaccounted-for uncertainty (e.g. the planner was unaware of the effects of wind on the power consumption of the vehicle). Future field testing of the mission planner and ASV should take into account measurements of the environmental factors and the vehicle model uncertainty. Properly identifying the vehicle model means system identification of the vehicle's hydrodynamic and aerodynamic properties through captive model testing, free running experiments (preferably with controlled wind effects), or computational fluid dynamics simulation of the loads on the ASV's submerged hull (hydrodynamics) and superstructure (aerodynamics). This would allow for the design of an appropriate marine vehicle model for use in an EKF, resulting in better kinematic state estimations. It would also allow for the design of a better control process through simulation of the fully identified vehicle kinetics model. Redesign of the vehicle's IMU location will also improve the reliability of heading estimation by reducing magnetic interference.

The ultimate goal is for the supervisor to predict the survival probability and make active changes to the plan that will positively affect its progress towards completing missions. The data-driven approach of the LSTM forecaster seems promising, even with the noisy kinematic state and energy consumption training data used. The forecaster was shown to have a reliable horizon between 5 and 15 seconds, depending on the operator's criteria. This horizon is too small for it to be of immediate practical use for this prototype model, as the data that it was trained on originated from noisy, low-cost components. However, being able to accurately forecast energy consumption ahead of time *in situ* presents the vehicle planner with the opportunity to make energy-informed decisions ahead of time, which could be critical in ensuring the vehicle returns to a recoverable position.

Noise and inteference effects on the navigation sensors (such as magnetic interference on the IMU) could not be fully accounted for and comes with the territory of using low-cost sensors. We suggest that the reliable energy forecast horizon could be extended if the LSTM was trained on data from commercially available state-of-the-art navigation equipment. As a separate entity the energy forecaster could be implemented on many different vehicle types, requiring reliable previous mission history data. Transfer learning shows promise for enabling LSTMs trained on one vehicle data set to be adapted to another vehicle without significant loss in accuracy (Laptev et al., 2018).

The mission planner energy model could be improved by including more information from the environment or environment forecasts (such as wind, waves, currents and traffic conditions), such as what has been achieved in Hollinger et al. (2016). Additionally, the current trajectory generator assumes the vehicle instantly changes orientation to face the next task. The trajectory generator could be developed further to more realistically represent the vehicle's true path over ground.

7 Conclusions

This paper addresses two challenges for marine vehicle mission planning: adapting a plan based on the realisation of a partially unknown environment, and forecasting the progress of a given plan based on uncertain environment observations. In both cases, quantifying the environment as much as possible reduces uncertainty and allows for more reliable plan adaptation and plan forecasting. The power supply of the vehicle is a critical point of mission failure (Brito et al., 2014), thus energy consumption was chosen as the mechanism for evaluating the environment and the vehicle's progression along the plan.

Making reliable forecasts of the vehicle's energy consumption improves the decision-making agent's skill in generating mission critical decisions ahead of time. Most importantly this reduces the risk of the vehicle getting into a situation where there are no available remedies. It also means that the vehicle can use the time saved to do more tasks.

The results presented in this paper demonstrate the potential that energy planning has for quantifying the uncertainty in mission planning for marine platforms. An existing energy-aware mission planner was expanded into a two-stage planner and was tested on a custom AMV system operating in an uncertain environment. The second-stage supervisor agent uses survival functions as decision-making criteria for recourse actions, which the operator can threshold to control the supervisor's recourse policy. The energy forecaster demonstrated that predicting the energy consumption of a marine vehicle is possible even with an incomplete representation of the marine environment. Combining these three components produces a mission planner capable of creating an energy-aware plan; that is able to forecast the feasibility of the vehicle's current plan into the future; and take preemptive action to minimise the risk of power supply failures.

Acknowledgements

The authors acknowledge the support of the Australian Government's Research Training Programme and the Institute of Marine Engineering, Science and Technology's Laurie Prandolini Research Fellowship. This research was supported under Australian Research Council's Special Research Initiative for Antarctic Gateway Partnership (Project ID SR140300001). Special thanks to the technical staff of the Australian Maritime College and to Cameron Thompson for assistance during field trials.

References

Ai-Chang, M., Bresina, J., Charest, L., Chase, A., Hsu, J. C. J., Jonsson, A., Kanefsky, B., Morris, P., Rajan, K., Yglesias, J., Chafin, B. G., Dias, W. C., and Maldague, P. F. (2004). MAPGEN: mixed-initiative planning and scheduling for the Mars Exploration Rover mission. *IEEE Intelligent Systems*, 19(1):8–12.

Brito, M., Smeed, D., and Griffiths, G. (2014). Underwater glider reliability and implications for survey design. *Journal of Atmospheric and Oceanic Technology*, 31(12):2858–2870.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1):14–23.

Bureau of Meteorology (2018). Launceston, Tasmania November 2018 Daily Weather Observations. BoM, http://www.bom.gov.au/climate/dwo/201811/html/IDCJDW7025.201811.shtml. Accessed: December 2018.

Caley, J. A., Lawrance, N. R. J., and Hollinger, G. A. (2017). Deep networks with confidence bounds for robotic information gathering. In *Proceedings of Robotics: Science and Systems Conference Workshop on New Frontiers for Deep Learning in Robotics (RSS)*, Boston, MA, USA.

Chao, I., Golden, B. L., and Wasil, E. A. (1996). The team orienteering problem. *European Journal of Operational Research*, 88(3):464–474.

Evers, L., Glorie, K., van der Ster, S., Barros, A. I., and Monsuur, H. (2014). A two-stage approach to the orienteering problem with stochastic weights. *Computers & Operations Research*, 43:248–260.

Ferreira, A. S., Costa, M., Py, F., Pinto, J., Silva, M. A., Nimmo-Smith, A., Johansen, T. A., de Sousa, J. B., and Rajan, K. (2018). Advancing multi-vehicle deployments in oceanographic field experiments. *Autonomous Robots*, 43:1555–1574.

Fossen, T. (2011a). *Handbook of Marine Craft Hydrodynamics and Motion Control*, chapter 9, pages 232–233. John Wiley & Sons, Ltd.

Fossen, T. (2011b). *Handbook of Marine Craft Hydrodynamics and Motion Control*, chapter 10, pages 257–266. John Wiley & Sons, Ltd.

Fossen, T. (2011c). *Handbook of Marine Craft Hydrodynamics and Motion Control*, chapter 12, pages 398–415. John Wiley & Sons, Ltd.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232.

Hamza, A. and Ayanian, N. (2017). Forecasting battery state of charge for robot missions. In *Proceedings of the Symposium on Applied Computing*, SAC '17, pages 249–255, New York, NY, USA. ACM.

Hollinger, G. A., Pereira, A. A., Binney, J., Somers, T., and Sukhatme, G. S. (2016). Learning uncertainty in ocean current predictions for safe and reliable navigation of underwater vehicles. *Journal of Field Robotics*, 33(1):47–66.

Laptev, N., Yu, J., and Rajagopal, R. (2018). Reconstruction and Regression Loss for Time-Series Transfer Learning. In *Proceedings of ACM Conference (SIGKDD MiLeTS'18)*, London, UK. ACM.

McGann, C., Py, F., Rajan, K., Thomas, H., Henthorn, R., and McEwen, R. S. (2008). A deliberative architecture for AUV control. In *2008 IEEE International Conference on Robotics and Automation*, pages 1049–1054, Pasadena, CA, USA.

Moore, T. and Stouch, D. (2014). A generalized Extended Kalman Filter implementation for the robot operating system. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*, Padua, Italy. Springer.

Py, F., Pinto, J., Silva, M. A., Johansen, T. A., Sousa, J., and Rajan, K. (2016). EUROPtus: A mixed-initiative controller for multi-vehicle oceanographic field experiments. In *International Symposium on Experimental Robotics*, pages 323–340, Tokyo, Japan.

Sadrpour, A., Jin, J. J., and Ulsoy, A. G. (2013). Mission energy prediction for unmanned ground vehicles using real-time measurements and prior knowledge. *Journal of Field Robotics*, 30(3):399–414.

Shang, K., Chan, F. T. S., Karungaru, S., Terada, K., Feng, Z., and Ke, L. (2016). Two-stage robust optimization for orienteering problem with stochastic weights. *arXiv e-prints*, page arXiv:1701.00090.

Sotzing, C. C., Evans, J., and Lane, D. M. (2007). A multi-agent architecture to increase coordination efficiency in multi-AUV operations. In *OCEANS 2007 - Europe*, pages 1–6, Aberdeen, Scotland, UK.

Thompson, F. and Galeazzi, R. (2020). Robust mission planning for autonomous marine vehicle fleets. *Robotics and Autonomous Systems*, 124.

Tsiligirides, T. (1984). Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, 35(9):797–809.

Tsiogkas, N. and Lane, D. M. (2018). An evolutionary algorithm for online, resource-constrained, multivehicle sensing mission planning. *IEEE Robotics and Automation Letters*, 3(2):1199–1206.

Zaytar, M. A. and Amrani, C. E. (2016). Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. *International Journal of Computer Applications*, 143(11):7–11.

Zlot, R. M. (2006). *An Auction-Based Approach to Complex Task Allocation for Multirobot Teams*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.