AN ENERGY-AWARE ARCHITECTURE:

A PRACTICAL IMPLEMENTATION FOR AUTONOMOUS UNDERWATER VEHICLES

by
Valerio De Carolis

Submitted for the degree of
Doctor of Philosophy

Ocean Systems Laboratory
School of Engineering and Physical Sciences
Heriot-Watt University

2016

Abstract

Energy awareness, fault tolerance and performance estimation are important aspects for extending the autonomy levels of today's autonomous vehicles. Those are related to the concepts of *survivability* and *reliability*, two important factors that often limit the trust of end users in conducting large-scale deployments of such vehicles. With the aim of preparing the way for *persistent* autonomous operations this work focuses its efforts on investigating those effects on underwater vehicles capable of long-term missions.

A novel energy-aware architecture for autonomous underwater vehicles (AUVs) is presented. This, by monitoring at runtime the vehicle's energy usage, is capable of detecting and mitigating failures in the propulsion subsystem, one of the most common sources of mission-time problems. Furthermore it estimates the vehicle's performance when operating in unknown environments and in the presence of external disturbances. These capabilities are a great contribution for reducing the operational uncertainty that most underwater platforms face during their deployment. Using knowledge collected while conducting real missions the proposed architecture allows the optimisation of on-board resource usage. This improves the vehicle's effectiveness when operating in unknown stochastic scenarios or when facing the problem of resource scarcity.

The architecture has been implemented on a real vehicle, Nessie AUV, used for real sea experiments as part of multiple research projects. These gave the opportunity of evaluating the improvements of the proposed system when considering more complex autonomous tasks. Together with Nessie AUV, the commercial platform IVER3 AUV has been involved in the evaluating the feasibility of this approach. Results and operational experience, gathered both in real sea scenarios and in controlled environment experiments, are discussed in detail showing the benefits and the operational constraints of the introduced architecture, alongside suggestions for future research directions.

Acknowledgements

It has been three years since I joined the Ocean Systems Laboratory (OSL) under the supervision of Prof. David Lane and Dr. Keith Brown. The OSL team welcomed me with enthusiasm since my arrival, it gave me time and opportunity to grow, it supported me in facing the great challenges encountered during the PhD process and it has been my new home ever since. This work is a small part of what has been my contribution to the daily activities of a vibrant research laboratory, to all the efforts put in further developing our systems and to the wonderful time spent "getting our hands dirty" together with Arnau Puig Mensa, Marian Andrecki, Gwenole Henry and their robotics projects.

I would like to thank Prof. David Lane, who showed me how people skills are a valuable characteristic when facing complex task, and Dr. Keith Brown, which guided me throughout the whole PhD process. A special notice goes to our chief technician Len McLean, for supporting the development of this long work and for teaching me a lot of interesting things. Finally, I would like to thank my supporters, my family, my friends here and at home, for always being there and for having often long conversations despite running out of time. The new relationships I made and the experience gained will always hold a special place in my memories. Thank you all.

ACADEMIC REGISTRY
Research Thesis Submission

Name*:*	*Valerio De Carolis*		
School/PGI:	School of Engineering and Physical Sciences		
Version: *(i.e. First, Resubmission, Final)*	Final	Degree Sought (Award **and** Subject area)	*PhD*

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1) the thesis embodies the results of my own work and has been composed by myself
2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate*:*	Valerio De Carolis	Date:	

Submission

Submitted By *(name in capitals):*	
Signature of Individual Submitting:	
Date Submitted:	

For Completion in the Student Service Centre (SSC)

Received in the SSC by *(name in capitals):*			
Method of Submission *(Handed in to SSC; posted through internal/external mail):*			
*E-thesis Submitted (**mandatory for final theses)***			
Signature:		Date:	

Please note this form should bound into the submitted thesis.

Updated February 2008, November 2008, February 2009, January 2011

Contents

List of Figures

List of Tables

List of Abbreviations

AUV Autonomous Underwater Vehicle.

BMS Battery Management System.

CC Coulomb-counting.

COP Correlated Orienteering Problem.

COTS commercial off-the-self.

DOD Depth of Discharge.

DOF degree of freedom.

DVL Doppler Velocity Log.

EA-COP Energy-aware Correlated Orienteering Problem.

EA-OP Energy-aware Orienteering Problem.

ESC Electronic Speed Control.

EV Electric Vehicle.

FDI Fault Detection and Isolation.

FDIR Fault Detection, Isolation and Recovery.

FOG Fiber Optic Gyroscope.

GNSS Global Navigation Satellite System.

HAL Hardware Abstraction Layer.

HIL hardware-in-the-loop.

KP Knapsack Problem.

Li-ion Lithium Ion.

LiPo Lithium Ion Polymer.

LWPR Locally Weighted Projection Regression.

MILP Mixed Integer Linear Programming.

MIQP Mixed Integer Quadratic Programming.

MSE Mean Squared Error.

NiCd Nickel Cadmium.

NiMH Nickel Metal Hydride.

NRMSE Normalised Mean Root-mean-squared Error.

OP Orienteering Problem.

OVRP Open Vehicle Routing Problem.

PCA Principal Component Analysis.

PoMC Probability of Mission Completion.

RMSE Root-mean-squared Error.

ROV Remotely Operated Vehicle.

RUL Remaining Useful Life.

SOC State of Charge.

SOH State of Health.

SVM Support Vector Machine.

TSP Travelling Salesman Problem.

UAV Unmanned Aerial Vehicle.

UGV Unmanned Ground Vehicle.

USV Unmanned Surface Vehicle.

List of Symbols

α Smoothing parameter used with exponential functions.

B, B^{-1} Direct and inverse thruster configuration matrix (TCM).

B_w Weighted thruster configuration matrix (TCM).

d Binary fault decision variable.

$\Delta v_c, \Delta \psi_c$ Sea current's speed and direction uncertainty.

Δr Residual among short-term energy features.

Δr_{max} Maximum residual among short-term energy features.

Δu Maximum allowed difference among consecutive input commands u.

$e_{bat}(t)$ Battery stored energy at time t.

$e_{bat_{full}}$ Battery design energy (measured in J).

e_m Total energy consumption for the mission process.

e_{max} Allocated energy for the current mission plan.

ε Energy usage per unit distance (measured in J/m).

$\varepsilon_{nav}(\psi)$ Estimated energy usage per unit distance for navigation heading ψ.

E_r, E_m Short-term energy features (real and modelled components).

e_T Energy threshold for the mission process.

$e(t)$ Cumulative energy measured at time t.

η_T Thrust efficiency coefficient.

f Actuators' force request vector ($f_{(1\times N)} = [f_0\ f_1\ \dots\ f_N]$).

$f_M(e_m, t_m)$ Joint probability density for the mission process M.

f_{act}, f_{max} Thruster's available and maximum output force.

$\gamma_{\eta_T}, \delta_{\eta_T}$ Thrust efficiency adjustment coefficients (increase, decrease).

$G_m(.)$ Non-linear *throttle-to-thrust* characteristic function.

$g_m(.)$ Non-linear *throttle-to-current* characteristic function.

I_{bat} Battery instantaneous current draw.

$I(t)$ Current measured at time t.

λ_d Detection threshold used in the fault detection algorithm.

$\Lambda_{high}, \Lambda_{low}$ Maximum rate of change for input command u (increase, decrease).

m Diagnostic metric (derived from residuals).

μ_m, σ_m^2 Diagnostic metric's expected mean and variance.

N Number of thrusters or actuators.

N_s Number of samples.

ν Vector of linear and angular velocities in vehicle's body-fixed frame ($\nu_{(1\times6)} = [u\ v\ w\ p\ q\ r]$).

$\nu_{cruise}(\psi)$ Estimated cruise speed for navigation heading ψ.

$\nu_{allowed}, \nu_{cruise}$ Vehicle's adjusted and standard cruise speed vectors.

$\nu_c(t), \psi_c(t)$ Sea current's speed and direction at time t.

P_F Probability of failure.

P_{fa} Probability of false alarm.

τ Generalized thrust vector in vehicle's body-fixed frame ($\tau_{(1\times6)} = [x\ y\ z\ k\ m\ n]$).

T_{est} Estimated thrust for a given actuator.

T_i Instantaneous thrust for the i-th actuator.

t_m Total execution time for the mission process.

t_{max} Allocated time for the current mission plan.

t_r, t_f Settling and failure time thresholds used in the fault detection algorithm.

T_s Sampling time for energy, voltage and current measurements.

t_T Time threshold for the mission process.

$u_{lim}(t)$ Control input command after rate limiting at time t.

$u(t)$ Control input command at time t (e.g. *throttle* command).

V_{bus} Electric bus voltage, approximated as constant.

V_{high} Battery maximum rated voltage.

V_{low} Battery minimum rated voltage.

$V(t)$ Voltage measured at time t.

W Weighting matrix used for thruster remapping.

W_e, W_f Integration windows used for calculating the short-term energy residuals.

List of Publications

This research has produced the following publications:

- **Energy-aware Fault-mitigation Architecture for Underwater Vehicles** – V. De Carolis, K. Brown and D. Lane in *Journal of Autonomous Robots, Springer* (2016), doi: 10.1007/s10514-016-9585-x, issn: 1573-7527.
- **Run-time Energy Estimation and Route Optimization for Autonomous Underwater Vehicles** – V. De Carolis, K. Brown and D. Lane submitted to *IEEE Journal of Oceanic Engineering* (2016).
- **Energy-Constrained Informative Routing for AUVs** – N. Tsiogkas, V. De Carolis and D. Lane in *Proceedings of IEEE-MTS Oceans'16, Shanghai, China* (2016).
- **Online Fault Detection and Model Adaptation for Underwater Vehicles in the Case of Thruster Failures** – G. Fagogenis, V. De Carolis, and D. Lane in *Robotics and Automation (ICRA), 2016 IEEE International Conference, Stockholm, Sweden* (2016).
- **An Adaptive Controller for Autonomous Underwater Vehicles** – C. Barbalata, V. De Carolis, M. Dunnigan, Y. Petillot and D. Lane in *Proceedings of Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference* (2015).
- **Low-cost Energy Measurement and Estimation for Autonomous Underwater Vehicle** – V. De Carolis, D. Lane, and K. Brown in *Proceedings of IEEE-MTS Oceans'14, Taipei, Taiwan* (2014).
- **Probabilistic Approaches in Ontologies: semantics and uncertainty for AUV autonomy** – F. Maurelli, Z. Saigol, G. Papadimitriou, T. Larkworthy, V. De Carolis, and D. Lane in *Proceedings of IEEE-MTS Oceans'13, San Diego, USA* (2013).

Chapter 1

Introduction

Our freedom to doubt was born out of a struggle against authority in the early days of science. It was a very deep and strong struggle: permit us to question, to doubt, to not be sure. I think that it is important that we do not forget this struggle and thus perhaps lose what we have gained.

Richard P. Feynman

In the field of underwater vehicles, and autonomous robotics in general, long-term autonomy, robust operations and self-awareness are active areas of research. In the recent years different joint efforts [1]–[4], in the form of European projects, have been exploring new operating scenarios for the underwater domain, where multiple vehicles are often envisioned conducting cooperative long missions with high degrees of autonomy and without human intervention. These scenarios require improvements to be made in different areas [5], [6] of today's autonomous architectures, for instance, advanced mission planning, on-board knowledge representation, real-time status assessment and self-aware management, to name a few. These requirements are needed to deliver reliable and trustworthy platforms that can found applications in real life scenarios, such as oil and gas, maritime security, underwater archaeology or sea construction. One key aspect of this new wave of research is the *persistent autonomy* [1] concept: an improved level of autonomy [7] where underwater platforms are tasked with the execution of longer and more complex missions while being capable of coping with dynamic environments, uncertainty and unexpected events. This also has the goal of reducing the need for operator assistance to a minimum and, at the same time, maximizing the platform's efficiency when operating in the field.

Several strategies are often employed to characterise the operation of a complex system, such as Autonomous Underwater Vehicles (AUVs) deployed in marine environments. One of those is the introduction of non-functional requirements. These are used in this domain

to help end users or operators to qualify the system's properties in terms of expected behaviours, end goals and service availability. Non-functional requirements are opposed to functional ones, focused on characterising, instead, behaviours of specific system's components such as sensors or scientific payload. In the underwater context, service availability is usually constrained by the physical limitations of marine environments, such as low bandwidth communications, absence of Global Navigation Satellite System (GNSS) signals, resource shortage, limited *a priori* knowledge of mission environments or, more generally, the presence of unknown conditions and dynamic scenarios. Given these limitations non-functional requirements are often broadly defined.

A relevant requirement for long-term underwater operations is the *operability* one. This defines the ability to keep a system reliable and in a safe functioning condition under a wide range of possible operating scenarios, such as the ones encountered during long-term operations. The operability requirement is related to two more specific ones: *reliability* and *survivability*. The former, classically defined as the ability of a system to operate under given conditions for a specific amount of time, is used for evaluating the system with respect to failures of its internal components. The latter, on the other hand, defines a similar ability when considering, instead, external factors and potential failures on the system's operations not originated from the system itself.

The challenges of long-term autonomy and the need of improving the *operability* of unmanned platforms have pushed researchers to address these issues with the introduction of different strategies. These are, for instance, *behaviour-based* architectures [8], *expert* systems [9] and *semantic* representations [10], [11], that allow a vehicle to react, respond and, potentially, overcome an unexpected event or failure encountered during long-term operations. Each approach leverages a specific aspect of the artificial intelligence domain and it offers a general solution that often abstracts away from the physical properties of the underlying underwater vehicle. This feature, while reasonable for vehicles used only in research laboratories, becomes a limitation when those are tasked with more challenging field operations [12], such as deep-sea inspections, cooperative tasks or under-ice missions, that require a guaranteed level of reliability from the physical hardware and software architectures in order to survive the harsh environmental conditions to which underwater vehicles are constantly exposed.

An aspect, often underestimated by previous proposals, is that most underwater vehicles share a common design where a limited energy source provides the electrical power required by all internal subsystems and additional payloads. This characteristic highlights the fact that the energy usage still represents a critical aspect of their design where even a small *undetected fault* or *variation from the expected behaviour* could result in a non-predictable or dangerous outcome. This feature suggests that increased awareness, specifically in the energy domain, represents an enabling factor for long-term autonomous operations conducted with a minimum amount of external intervention. Despite its importance, limited research has been done on the energy-awareness topic for unmanned

vehicles. Main contributions are in the field of ground [13]–[15] and aerial vehicles [16], [17], where recent studies show that energy-aware mission characterization and energy-reliable planning are beneficial techniques to improve the effectiveness of vehicles during field operations. In the underwater domain, on the other hand, energy related aspects have been only marginally addressed.

1.1 Research Objectives

The aim of this work is to extend previous literature in the underwater field, specifically by introducing the concept of *energy-awareness* as essential in the architectural design (e.g. hardware and software) of next-generation vehicles. The work investigates the use of knowledge derived from the runtime analysis of on-board energy consumption. This, together with other techniques, aims at developing an architecture that can improve the *reliability* and *survivability* aspects of long-term autonomy AUVs. Two main questions are explicitly addressed by this research:

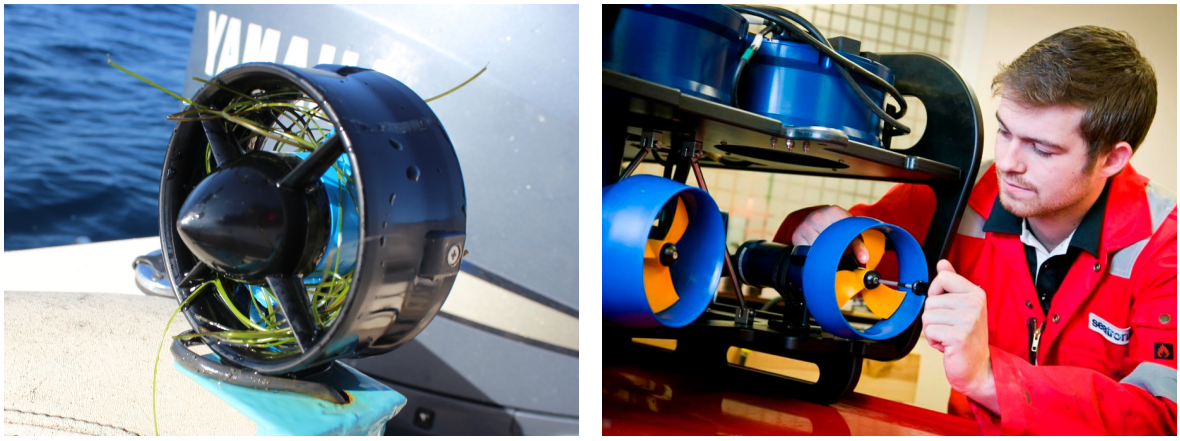
- *Is it possible to assess the internal qualitative state of an underwater vehicle by monitoring the energy usage of its internal components?*
- *Is it possible to estimate the effect of external disturbances on future mission's performance by monitoring the energy consumption of an underwater vehicle?*

These are related to some relevant limitations, identified in previous research, to the long-term deployment of autonomous vehicles: the presence of efficient *failure mitigation capabilities* [18], [19] and the availability of *on-line reliability estimation* modules [13], [15] that can function both in presence of operational uncertainty and in unknown environments. These aspects, analysed individually in the past, are evaluated together in this work with respect to their implication for the vehicle's autonomy architecture.

1.2 Methodology Overview

A practical implementation of an energy-aware architecture is presented for an existing underwater vehicle, the Nessie AUV, used in real sea trials. This required, initially, the development of a hardware solution (known as low-cost energy monitor) that provides the existing platform with energy monitoring capabilities for its internal subsystems. Using measurements collected in controlled environments this solution allowed the characterisation of the behaviour of internal components and the derivation of analytical models later used in the proposed architecture for fault diagnostic purposes.

Attention is then focused on introducing an energy-based fault mitigation framework. This monitors at runtime the behaviour of vehicle's propulsion: a relevant subsystem that is often affected by failures during sea operations, is shown for instance in Figure 1.1. The framework, after recognising unexpected conditions such as soft-failures (or degradations),



(a) (b)

Figure 1.1: Example of marine actuators. Field operations in shallow water are often affected by seaweed ingestion in the thruster's duct. Maintenance operations require long intervention time in particular if the thruster is damaged during sea operations.

is able to introduce an adequate reaction, for instance reconfiguring the use of actuators, that allows the vehicle to resume its normal operations. Field experiments are then used to validate the usefulness of such an approach. Results show good mitigation capabilities, when in presence of moderate degradations and complete support for detection of faults even when in presence of environmental disturbances.

After handling the mitigation aspects, attention is shifted toward the introduction of a runtime performance estimation framework. This is used for evaluating the vehicle's navigation performance when in presence of external disturbances, such as sea currents. The framework employs measurements collected at runtime with the low-cost energy monitor. After an initial collection of samples it provides higher level modules with constantly updated energy metrics that characterise the vehicle's behaviour with respect to the external environment. Another relevant feature is the feasibility assessment for the current vehicle's mission. This is done using an iterative process that employs metrics calculated at runtime together with knowledge of the mission's plan to evaluate if constraints on the resource usage can be respected when taking into account the measured locomotion performance. Together with assessment aspects, the performance estimations are employed in route optimisation procedures aimed at adjusting the vehicle's behaviour when conducting inspection tasks in unknown scenarios. These, shown for example in Figure 1.2, are objectives of interest of the recent research efforts [1]–[4] focused on improving the intervention capabilities of next-generation AUVs. Even with the runtime performance estimation framework, field experiments are used to validate the derivation of environmental knowledge from energy measurements and the estimation of mission feasibility in presence of external disturbances. Overall results show that the analysis of energy provides enough information to assess with good confidence the vehicle's performance, to evaluate the probability of completing a mission successfully and to provide useful metrics for optimisation procedures that allow improving the vehicle's effectiveness in the field.

(a) (b)

Figure 1.2: Field operations with Nessie AUV conducting inspection tasks around human-made structures. Operators supervise an initial field testing using a remote console where telemetry data is shown. The vehicle inspects the supporting structures of a marine pier and collects geometric information using the on-board sensors.

The results collected in this work suggest that an improved energy-awareness represents a valid proposition for extending the underwater platform capabilities while, at the same time, reducing their dependability on human intervention. Remarkable features of this work are: the introduction of a low-cost energy monitoring solution suitable to be used in a broad class of vehicle's designs, the use of energy-derived metrics in the implementation of a fault mitigation system, capable of detecting unexpected behaviours such as soft-failures or degradations in the propulsion system, and in the estimation of the navigation performance when in presence of external disturbances, evaluating their consequences on the mission given energy usage constraints. The development of the work is also supported by the use of a state-of-the-art non-linear regression algorithm, known as Locally Weighted Projection Regression (LWPR), that is employed in the data-driven approach used for representing vehicle's components and environmental effects.

1.3 Outline

The rest of this work is organised as follows. In Chapter 2 related studies found in literature are presented. Work done focusing on the main subsystems of autonomous underwater vehicle is discussed, with attention to the battery and propulsion subsystems as object of interest for the proposed architecture. After the initial discussion relevant work done in the context of fault detection, diagnosis and mitigation is introduced. This describes other strategies, used in the past, to address the issue of survivability for underwater vehicles. Together with this aspect, other studies focused around the concept of reliability in the context of autonomous missions are presented.

In Chapter 3 the new energy-aware architecture is proposed while describing its use in the context of a wider autonomy software solution, previously employed in field experiments conducted with the experimental platform involved in this work. The main

architectural elements, a fault mitigation and a runtime performance estimation framework, are also presented in this chapter. These, while relying on the runtime analysis of the vehicle's energy consumption, address the two principal aspects studied in this research: self-assessment of the vehicle's state and analysis of external factors. Beside discussing these features, the information flow within developed modules is also described with respect to the execution of a general autonomous mission.

In Chapter 4 a detailed discussion of the fault mitigation framework is presented. This targets the propulsion subsystem, a component often affected by operational uncertainty, and aims at detecting and mitigate the presence of degradation faults. These affects marine actuators by changing their efficiency, disrupting the vehicle's navigation in the more severe cases or lowering the overall performance in the less severe ones. These failures are targeted explicitly as they represent a generalisation of the simpler scenario, often found in literature, when a sudden loss of the actuators is considered.

In Chapter 5 the runtime performance estimation framework is introduced. This makes use of energy measurements similar to the ones collected by the mitigation framework and it evaluates the navigation performance of the vehicle when operating in presence of external disturbances. Together with estimation aspects two optimisation procedures, still built around energy-derived metrics, are also introduced. These employ the collected operational knowledge to adapt the vehicle's tasks, calculating a suitable navigation route that is efficient both in terms of energy consumption and maximising the gathering of information in the case of inspection missions.

In Chapter 7 the experimental validation of the proposed architecture is discussed. First the operational experience acquired during sea trial campaigns involving the Nessie AUV is reported. Second the results of experiments conducted in real sea scenarios and in controlled environments are presented. A relevant aspect of this validation, targeting the fault mitigation framework, is the reported experience when a real failure in the propulsion subsystem of the Nessie AUV has been discovered at the end of sea operations. This event allowed the collection of relevant data and the evaluation of the proposed diagnostic system when in presence of both synthetic and real failures. Together with such analyses, other results are presented for experiments focusing the runtime estimation framework. In this case two different platforms, the Nessie AUV together with a commercial solution, the IVER3 AUV, have been deployed for field missions while their navigation performance is analysed by the proposed architecture. In this context further experiments conducted in simulated scenarios are also discussed. These show how the knowledge acquired using the proposed architecture is employed by route optimisation procedures to improve the vehicle's effectiveness when operating in unknown environments. In Chapter 8, finally, the contributions of this work are summarised and the major findings are presented with respect to the relevant elements of the energy-aware architecture. After discussing those, future research directions are also presented, highlighting additional techniques that could further extend the capabilities of the proposed architecture.

Chapter 2

Related Work

We have found it of paramount importance that in order to progress we must recognize our ignorance and leave room for doubt. Scientific knowledge is a body of statements of varying degrees of certainty – some most unsure, some nearly sure, but none absolutely certain.

Richard P. Feynman

AUVs, like other unmanned systems, may share a common architecture where a limited energy source provides the electrical power needed by internal subsystems connected to the primary electric bus. This characteristic makes these platforms depend on the constant availability of on-board resources and it highlights the fact that energy usage still represents a key aspect of their design. In fact, when operating away from human supervision even small *undetected* failures could result in a non-predictable or dangerous outcome for the system itself. Such events often materialise as *degradations*, small deviations from expected behaviours rather than abrupt or sudden changes in the operating conditions, of an internal subsystem. This subtle characteristic makes these faults difficult to detect if dedicated system management modules are not introduced in the original vehicle's architecture. For those reasons *reliability* and *survivability* concepts [10], [20] or, more in general, *operability* ones still represent an open problem in the autonomous vehicles community. In the following sections previous relevant work is presented together with its implications for marine vehicles. First, underwater vehicles are briefly discussed and their relevant subsystems are analysed. Second, diagnostic techniques applied in the context of fault detection and mitigation are introduced. Finally, high abstraction level decisions in the mission management's context are also discussed when considering reliability aspects and environmental uncertainty.

2.1 Autonomous Underwater Vehicles

Researchers analysed *availability* issues of autonomous systems in different contexts: from industrial automation domains to more common Electric Vehicles (EVs) and unmanned platforms, such as Unmanned Aerial Vehicles (UAVs), Unmanned Ground Vehicles (UGVs) or Unmanned Surface Vehicles (USVs). In the underwater domain, however, several challenges often limit a detailed analysis of those aspects. In fact, reduced communication bandwidth, short operating ranges, higher component costs, small payload sizes and low computational capabilities have, historically, required vehicle's designers to concentrate their focus on a specific set of tasks and to limit on-board systems to what is strictly necessary to serve the vehicle's purpose. In the recent years, with the development of sea-gliders [21] operating in ocean-scale deployments and with the experience gathered during long-term under-ice missions [22], [23], reliability and survivability aspects have become again a priority for the next generation of underwater designs.

Most of the current AUVs, for instance, used in research [24]–[26], competitions [27]–[29] or commercial applications [30]–[32], include monitoring capabilities of on-board resources, such as battery, voltages and temperatures. These capabilities, however, are often limited to data acquisition or telemetry roles and are not fully integrated within the vehicle's software architecture for advanced health management and monitoring operations. This is because most of the time reliable operations are still achieved through a human-in-the-loop process, such as scheduled maintenance and dedicated testing, that certifies the platform's efficiency before deploying vehicles in the real conditions. Two relevant subsystems, *battery* and *propulsion*, are often the focus of maintenance operations. These components represent the key parts of any design and have been focus of previous research studies.

2.2 Battery System

Bradley *et al.* [33] conducted a survey on power systems for AUVs highlighting how the battery system becomes a critical component for long-term deployments. In this work the authors also analyse how the introduction of Lithium Ion (Li-ion) and Lithium Ion Polymer (LiPo) batteries in the underwater domain is beneficial compared to other technologies, such as Lead–acid, Nickel Cadmium (NiCd) or Nickel Metal Hydride (NiMH) chemistry, used in earlier designs. Table 2.1 highlights the main characteristics of the most common battery chemistries.

Li-ion batteries are known for their large energy density ($\geq 200\,\mathrm{W\,h/L}$) and high specific energy ($\geq 150\,\mathrm{W\,h/kg}$). This characteristic allows designers to store large quantities of energy (1 kWh to 15 kWh) in small confined volumes such as marine hulls. The choice of this chemistry is widespread among modern platforms, especially for low-cost small-size vehicles (20 kg to 80 kg) and commercial medium-size platforms (100 kg to

Chemistry	**Energy Density** (W h/L)	**Specific Energy** (W h/kg)	**Cycles**
Lead–Acid	60 – 75	30 – 40	500 – 800
Nickel Cadmium (NiCd)	50 – 150	40 – 60	≈ 1500
Nickel Metal (NiMH)	140 – 300	30 – 80	500 – 1000
Li-ion ($LiCoO_2$)	250 – 360	150 – 250	400 – 1200
Lithium Polymer	230 – 300	130 – 200	500 – 1000
Lithium Iron ($LiFePO_4$)	170 – 220	80 – 120	≥ 2000

Table 2.1: Comparison of rechargeable battery chemistries.

600 kg). Under optimal conditions these vehicles are able to operate continuously for 6 h to 48 h before depleting the on-board energy resources [34]. Table 2.2 shows an overview for current commercial AUVs, highlighting their expected duration and stored energy resources.

Name	**Depth** (m)	**Weight** (kg)	**Energy** (kWh)	**Duration** (h)
Remus 100	100	45	1.0	10
IVER3	100	38	0.7	12
Bluefin-9	200	50	1.5	12
Bluefin-12	200	204	4.5	26
Remus 600	600	326	5.2	50
MUNIN	1000	300	5.0	22
Hugin 1000	3000	850	15.0	24

Table 2.2: Comparison of small and medium class commercial AUVs.

2.2.1 Analytical Models

Despite the benefits of Li-ion and LiPo batteries their non-linear behaviour introduces extra challenges that system engineers need to take into account when integrating those components. Management of energy resources has been studied extensively [35] in the context of portable and mobile systems. Different methods [36] have been proposed to evaluate the State of Charge (SOC), State of Health (SOH) and Depth of Discharge (DOD) of integrated battery packs from discrete voltage, current and temperature measurements. Several formulations [37]–[39] have been proposed to analytically represent the state of a battery and thus infer the effective residual capacity in real usage scenarios. In these works equivalent circuit models are introduced and validated in controlled experiments. Examples of those are shown in Figure 2.1 and in Figure 2.2, where, respectively, an accurate lumped parameter equivalent circuit and a simpler, yet effective, low parameter count alternative are presented. The first model has been introduced by Chen *et al.* [39] and combines an accurate current-voltage characteristic (I-V) with an equivalent model for the battery lifetime, expressed in this case as combination of battery capacity and self-discharge effects. This model has been validated on several Li-ion and NiMH cells and it shows a 30 mV maximum error voltage together with a 0.4% battery runtime prediction

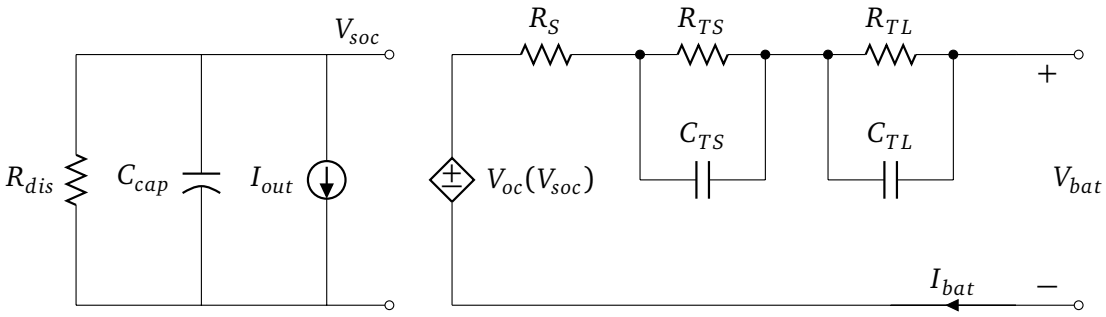


Figure 2.1: Accurate Li-ion battery model with lumped parameters. This includes lifetime (left) and current-voltage characteristic (right) components to reproduce the behaviour of real battery cells. Battery capacity is modelled using the charge stored in the capacitor C_{cap}. Recovery and second-order effects are modelled using capacitors C_{TS} and C_{TL} (adapted from [39]).

error in presence of different load profiles. Such a performance is achieved using a large number of parameters (more than 7) and knowledge of cycle count and temperature effects. The model itself is a combination of previous state-of-the-art formulations and it used as a reference in term of performance [38]. On the other hand, the second model, introduced by Johnson *et al.* [37], trades some accuracy by neglecting some non-linear dependencies often observed in Li-ion chemistries and simplifies the final lumped parameter equivalent circuit. In this case only 5 parameters are used and no extra dependencies are required to model the two principal aspects of vehicle's batteries, such as their internal resistance and the non-linear depletion effect under load. This has been derived from previous formulations and has been applied with satisfactory results to commercial batteries used in the EV community.

Additional work [40], [41] has been done recently towards a probabilistic estimation of battery state. This expands previous studies introducing relationships with ageing effects and components health state often only observed on longer time scales. These studies combine analytical models, real measurements and historical data using different approaches, such as Bayesian frameworks and Particle Filter methods, and improve the quality of state estimations together with Remaining Useful Life (RUL) [42] predictions.

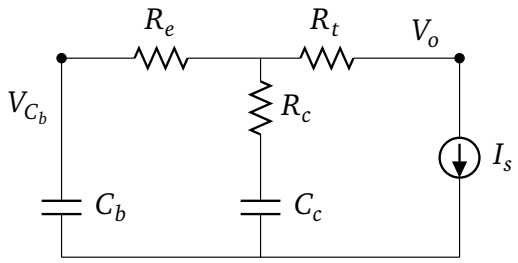


Figure 2.2: Simplified Li-ion battery model with lumped parameters. Battery capacity and recovery effects are modelled using the charge stored in the capacitors C_b and C_c (adapted from [37]).

2.2.2 SOC Estimation

Abundance of battery models, often different in terms of parameters and complexity, finds, in some cases, limited application in the underwater domain. This is because, for instance, several methods given their complexity rely on the off-line processing of a large amount of data or because the lack of on-board computational resources prevents a correct evaluation of runtime estimations needed to model the non-linear battery effects. Simpler techniques, however, are available in literature [43]. Some of these estimate the SOC by means of coulomb-counting (CC) procedures [44] and provide a good trade-off between complexity and accuracy often required for the implementation in embedded monitoring systems, such as Battery Management Systems (BMSs). Other approaches [45], instead, rely on fusion techniques, such as Kalman filter and its variants, to combine lumped parameters linear models with real measurements and, thus, estimate the SOC for a given battery system. Table 2.3 shows the principal features of different classes of SOC estimation techniques.

Method	Advantages	Disadvantages	Example
Direct	Quick, provides SOH estimation	Hard to access, chemistry dependant	Acid density, cathodic galvanostatic
Indirect Off-line	Accurate, provides SOH estimation	Temperature dependant, costly, exclusive usage	Full discharge, internal resistance
Indirect On-line	Economic, supports all battery types, dynamic	Historical data, may require large computation	Coulomb counting, OCV, PHM algorithms

Table 2.3: Comparison of SOC estimation methods. *Direct* methods observe internal quantities within a battery system and require physical access to individual cells. *Indirect* methods, on the other hand, only target quantities that are externally observable to a battery system.

From a system point of view battery models are represented using a continuous time state-state formulation:

$$\dot{\mathbf{x}} = \mathbf{A}\,\mathbf{x} + \mathbf{B}\,\mathbf{u} \tag{2.1}$$

$$\mathbf{y} = \mathbf{C}\,\mathbf{x} + \mathbf{D}\,\mathbf{u} \tag{2.2}$$

where $\mathbf{x}$ is the state vector, $\mathbf{y}$ the output vector, $\mathbf{u}$ the input vector and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ are matrices that describe the linear system. In the case of a simplified lumped parameters battery model, such as the NREL-Saft [37] also shown in Figure 2.2, these equations take the following expression:

$$\begin{bmatrix} \dot{V}_{C_b} \\ \dot{V}_{C_c} \end{bmatrix} = \begin{bmatrix} \frac{-1}{C_b(Re+Rc)} & \frac{1}{C_b(R_e+R_c)} \\ \frac{1}{C_c(R_e+R_c)} & \frac{-1}{C_c(R_e+R_c)} \end{bmatrix} \begin{bmatrix} V_{C_b} \\ V_{C_c} \end{bmatrix} + \begin{bmatrix} \frac{-R_c}{C_b(R_e+R_c)} \\ \frac{-1}{C_c} + \frac{R_cR_e}{(R_e+R_c)} \end{bmatrix} \begin{bmatrix} I_s \end{bmatrix} \tag{2.3}$$

and:

$$\begin{bmatrix} V_o \end{bmatrix} = \begin{bmatrix} \frac{R_c}{(R_e+R_c)} & \frac{1}{(R_e+R_c)} \end{bmatrix} \begin{bmatrix} V_{C_b} \\ V_{C_c} \end{bmatrix} + \begin{bmatrix} R_t + \frac{R_c R_e}{(R_e+R_c)} \end{bmatrix} \begin{bmatrix} I_s \end{bmatrix} \tag{2.4}$$

where R_t, R_c, R_e, C_c are model parameters, C_b the equivalent capacity that represents the energy stored inside the battery cell, V_{C_b}, V_{C_c}, respectively the voltages across C_b and C_c capacitors, I_s the current drawn from the battery and V_o the output voltage. The pair R_c, C_c is used to model the voltage recovery effect of Li-ion chemistries. The R_t and R_e, instead, are used to model the internal resistance effects of the battery in open-circuit and under-load conditions.

Using this model, the battery state is represented by the pair V_{C_b}, V_{C_c} and the overall battery energy is thus expressed as the usable charge stored inside in the capacitor C_b:

$$e_{bat_{full}} = \frac{1}{2} C_b V_{high}^2 - \frac{1}{2} C_b V_{low}^2 \tag{2.5}$$

where V_{high} and V_{low} represent the operating voltage range of the modelled battery. In this case the residual battery energy at a given time can be expressed as:

$$e_{bat}(t) = \frac{1}{2} C_b \left(V_{high}^2 - V(t)^2 \right) \qquad V(t) \geq V_{low} \tag{2.6}$$

where $V(t)$ is the state voltage calculated using the dynamical model while using the battery. This model has been applied successfully in the case of Flying Fish UAV [17], an solar-powered sea-plane used for research, to single cell Li-ion batteries common in the RC model domain. Tuned parameters for small cells are $R_t = 2.2\,\text{m}\Omega$, $R_c = 0.4\,\text{m}\Omega$, $R_e = 1.1\,\text{m}\Omega$, $C_c = 4.0\,\text{kF}$, $C_b = 18.45\,\text{kF}$ when considering an individual Li-ion cell with operating range of 3.1 V to 4.2 V.

2.2.3 Coulomb-Counting Method

Coulomb methods estimate the SOC by integrating current measurements from a given starting condition:

$$SOC(t) = \begin{cases} SOC_0 + \frac{1}{C_n} \int_{t_0}^{t} |I_{bat}|\, dt & \text{charge} \\ SOC_0 - \frac{1}{C_n} \int_{t_0}^{t} |I_{bat}|\, dt & \text{discharge} \end{cases} \tag{2.7}$$

their accuracy is related to the availability of precise current measurements I_{bat}, jitter-free time intervals dt, specific knowledge of the initial battery status SOC_0 and total battery capacity C_n. Coulomb-counting methods do not fully exploit the availability of analytical models and therefore they require a low implementation effort.

In many cases prognostic techniques [43] can provide accurate initial conditions that make the application of this relative simple counting process attractive in specific scenarios.

Low noise conditions are usually rare in complex systems and data fusion techniques are often required to increase the accuracy of resulting estimations.

2.2.4 Fusion Methods

Fusion methods rely on the availability of simplified linear analytical models that in combination with runtime measurements can provide a robust estimation of the SOC. When implementing embedded systems it is often common to represent the above models in discrete time. This allows a more straightforward transition from modelling phase and development ones. In the case of a linear system in presence of external noise the following formulation is used:

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \tag{2.8}$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{D}_k\mathbf{u}_k + \mathbf{v}_k \tag{2.9}$$

$$\mathbf{w}_k \sim N(\mathbf{r}_k, \mathbf{R}_k) \tag{2.10}$$

$$\mathbf{v}_k \sim N(\mathbf{q}_k, \mathbf{Q}_k) \tag{2.11}$$

where $\mathbf{w}_k$, $\mathbf{v}_k$ represent, respectively, the process and the measurement noise that characterise a real system. These quantities are assumed to follow a multivariate normal distribution $N(\mu, \Sigma)$ with known mean and covariance, respectively, $\mathbf{r}_k$, $\mathbf{q}_k$ and $\mathbf{R}_k$, $\mathbf{Q}_k$.

Using this system representation battery models are augmented [45] with the introduction of Kalman filtering procedure to combine the analytical formulations with real measurements and adjust at runtime the estimation of SOC for a given device. Under Kalman assumption the system can be seen as:

$$\mathbf{x}_{k|k-1} = \mathbf{F}_k\mathbf{x}_{k-1|k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k \tag{2.12}$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{2.13}$$

where $\mathbf{F}_k$ is the state transition model, $\mathbf{B}_k$ the control-input model, $\mathbf{z}_k$ the observation (or measurement) at time k of the true state $\mathbf{x}_k$ and $\mathbf{H}_k$ the observation model, and:

$$\mathbf{w}_k \sim N(0, \mathbf{R}_k) \tag{2.14}$$

$$\mathbf{v}_k \sim N(0, \mathbf{Q}_k) \tag{2.15}$$

where $\mathbf{w}_k$, $\mathbf{v}_k$ represent, respectively, the process and the measurement noise under the zero-mean assumption. In its simple form the Kalman filter (KF) equations are thus expressed as:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k\mathbf{u}_k \tag{2.16}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k\mathbf{P}_{k-1|k-1} + \mathbf{F}_k^T + \mathbf{Q}_k \tag{2.17}$$

for the prediction stage, where $\mathbf{Q}_k$ is the process noise covariance, $\mathbf{P}_{k-1|k-1}$ the *a priori* error covariance matrix. The update stage, instead, is expressed as:

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1} \tag{2.18}$$

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k \tag{2.19}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1} \tag{2.20}$$

where $\tilde{\mathbf{y}}_k$ is the innovation (or measurement residual), $\mathbf{R}_k$ the observation noise covariance, $\mathbf{K}_k$ the Kalman gain, derived from the $\mathbf{S}_k$ innovation covariance, and as:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k \tag{2.21}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1} \tag{2.22}$$

where $\hat{\mathbf{x}}_{k|k}$, $\mathbf{P}_k$ the *a posteriori* state estimate and covariance estimate of the system. In this case $\mathbf{F}_k \sim \mathbf{A}_k$ and $\mathbf{H}_k \sim \mathbf{C}_k$ are derived from the discrete-time linear system formulation for a given battery model.

When considering larger vehicle systems, the limiting factor for SOC estimates is usually not related to the accuracy of estimation methods but rather the precision of embedded sensors and measuring devices. Nonetheless variations of the original Kalman Filter are often found in literature. One example is the Extended Kalman Filter (EKF) [45] applied to Li-ion chemistry. The EKF soften some strong assumptions required in the standard formulation. In the EKF state transition and observation models don't need to be linear functions of the state $\mathbf{x}_k$ but may be, instead, differentiable functions:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_k \tag{2.23}$$

$$\mathbf{z}_k = g(\mathbf{x}_k) + \mathbf{v}_k \tag{2.24}$$

allowing, thus, a more complex relationship between variables than the standard filter variant. In battery research, the EKF has been successfully applied to different types of Li-ion cells [45] achieving an average SOC estimation error and a voltage error rate as low as 3.19% and 4.15% respectively.

2.2.5 Current Sensing

Most of battery models introduced so far rely on accurate current sensing capabilities. This is because the internal battery state is better derived using precise measurements of the current flowing through the device. Ziegler *et al.* [46] presented a detailed review of current sensing techniques. These cover different application scenarios, from embedded low power systems to larger industrial applications. In the context of a mobile system several options are available to vehicle's designers. Extra care is needed in selecting the right trade-off between sensing efficiency and measurement accuracy. Table 2.4 shows

the main features of the principal current sensing technologies.

Name	Bandwidth	DC cap.	Accuracy	Isolation	Range	Power Loss
Shunt Resistor (SMD)	kHz – MHz	yes	0.1% – 2%	no	mA – A	mW – W
Current Transformer	kHz – MHz	no	0.1% – 1%	yes	A – kA	mW
Rogowski Coil	kHz – MHz	no	0.2% – 5%	yes	A – MA	mW
Hall Effect	kHz	yes	0.5% – 5%	yes	mA – kA	mW

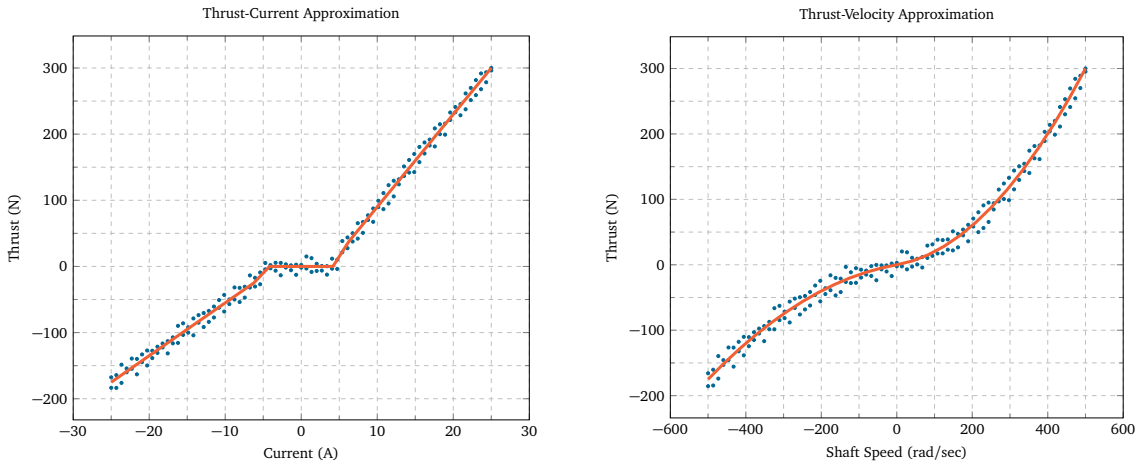
Table 2.4: Comparison of common current sensing technologies.

In the domain of battery powered vehicles important aspects are the isolation characteristics of embedded sensors, due to the dynamic nature of on-board loads, and their integration capabilities [47], [48]. For these reasons, conventional *Shunt Resistor* methods find application mostly in low power subsystems while *Hall Effect* and other magnetic technologies are commonly applied for monitoring the main electrical bus and other high power subsystems, such as propulsion and attitude control ones. The availability of current sensors is a relevant aspect in a vehicle's design as it allows not only the accurate derivation of battery states but it can further employed in monitoring [49], [50] the energy consumption and the power usage in autonomous vehicles.

2.3 Propulsion System

Beside the battery system another relevant component that affects the vehicle's reliability is the propulsion subsystem. This provides locomotion and control capabilities to marine vehicles and represents the central component for different types of operations, such as low speed manoeuvring, station keeping and general navigation [51]. For autonomous underwater systems, with exception of sea gliders or bio-inspired vehicles, electrical motors and propellers, in the form of marine thrusters, are the technology of choice for this subsystem. Even in this domain several methods have been proposed in literature to derive accurate component models by means of practical identification in controlled environments or analytical modelling. Components of interest for these analyses are underwater thrusters, also known as main actuators, used for control or general locomotion, and auxiliary actuators, used for operating control surfaces, such as fins and rudders, often found in more advanced AUV designs.

Healey *et al.* [52] and later Whitcomb *et al.* [53] introduced a relevant study of modelling techniques for small marine thrusters. In the first work a three element model (electric motor, propeller mapping and fluid model) is derived together with a set of identification experiments. These elements are responsible for the principal effects that characterise the behaviour of real devices. This approach, while it is extensive, often requires a great number of parameters and identification procedures to describe correctly the behaviour a specific type of actuator. The work of Kim *et al.* [54] improves the modelling accuracy by taking into account the effects of ambient flow velocity. This models the drop in performance that real actuators show during operation in non-static conditions. Even



(a) **(b)**

Figure 2.3: Example of thrust current and velocity approximations for marine thrusters. Collected samples are fitted using piecewise function approximations with respect to the forward and reverse operating modes of the analysed actuator (adapted from [55]).

in this case, however, the need of extensive controlled environment experiments, where water flow and component velocity can be adjusted, makes the suggested procedure impractical for hardware-in-the-loop testing and for on-board implementation of the resulting models.

Most of the early studies focus on the detailed analysis of thrusters behaviour under different environmental conditions and they do not take into account the effects of control electronics found in commercial off-the-self (COTS) components. These are, for instance, Electronic Speed Control (ESC) circuits or more advanced motor controllers usually packaged together with efficient marine actuators. An ESC acts as a proxy between the electrical motor itself and the external control by regulating the component's operations according to the feedback received by internal sensors. Those circuits integrate useful features, such as stall protection, detection of ventilation and current limiting, and can adjust on the fly operational parameters such as output power or shaft speed to guarantee smooth and reliable operations. On the other hand they introduce an extra layer between control architecture and physical actuators that make accurate modelling difficult. In those cases system designers can adopt a black-box approach, for instance using data driven methods, to overcome those limitations and still derive a useful set of characteristics that can describe the behaviour of underlying systems.

Aiming at a simpler modelling strategy, Hanai *et al.* [55] introduced an effective procedure to derive a model that relies on fewer parameters and employs a shorter identification procedure. This approach is better suited for using the derived model on-board as few modifications are required for control software and the use of external force sensors is required only during an initial training. The proposed model relies on three sets of equations: first, a thrust to voltage characteristic is fitted to training data, this represents the overall input-output characteristic of the actuator (e.g. voltage is the

control input for the motor controller), second, a current to thrust function approximation is derived and, finally, a velocity to thrust function has also been approximated from training data. Those are written as:

$$V(T) = \begin{cases} a_0 T + a_1 \sqrt[2]{T} + a_2 \sqrt[3]{T} + a_3 \sqrt[4]{T} & T \geq 0 \\ b_0 |T| + b_1 \sqrt[2]{|T|} + b_2 \sqrt[3]{|T|} + b_3 \sqrt[4]{|T|} & T < 0 \end{cases} \tag{2.25}$$

where T is the actuator's output thrust, a_x, b_x the set of parameters that identify the piecewise-continuous fit, respectively for the positive and negative thrust range, as:

$$T_{est}(I) = \begin{cases} c_0 I + c_1 & I \geq -\frac{c_1}{c_0} \\ 0 & -\frac{d_1}{d_0} < I < -\frac{c_1}{c_0} \\ d_0 I + d_1 & I \leq -\frac{d_1}{d_0} \end{cases} \tag{2.26}$$

where T_{est} is the estimated thrust, I the actuator's current draw, c_x, d_x the set of coefficients for two linear approximations with a dead zone in between, and as:

$$T_{est}(\omega) = \begin{cases} k_0 \omega^5 + k_1 \omega^4 + k_2 \omega^3 + k_3 \omega^2 + k_4 \omega & \omega \geq 0 \\ l_0 \omega^5 + l_1 \omega^4 + l_2 \omega^3 + l_3 \omega^2 + l_4 \omega & \omega < 0 \end{cases} \tag{2.27}$$

where ω is the propeller shaft velocity, k_x and l_x the set of coefficients for two 5-degree polynomials continuous through the origin. An example of thrust approximations functions is shown in Figure 2.3.

The availability of two thrust approximations is then exploited in the same work [55] to detect failures in a specific actuator. This is done, for instance in the case of thruster ventilation, by measuring the difference between estimated thrust values, the first derived taking into account the variable I and the other using the variable ω as measured by the vehicle. The calculated difference determines the state of actuator fault. While of great relevance from a practical point of view, the proposed approach does not explicitly address the effects of device latencies or motor controller delays. Those aspects, not of primary importance from the modelling point of view, gain their relevance if derived models are meant to be used in real-time, for instance, in fault detection systems or fault tolerant control schemas. In fact, precise knowledge of actuators characteristics allows the calculation on-the-fly of their expected behaviour and thus recognise the presence of any unwanted activity.

2.4 Fault Detection and Diagnosis

The goal of diagnostic methods is to use some *a priori* knowledge of a system (or process) to detect the presence of faults and to apply different *search strategies* to isolate their root

cause. This practice is also known in literature as Fault Detection and Isolation (FDI). The system's knowledge usually comes in the form of relationships between observations (or symptoms) and failures (or effects). Correct definition of those usually requires an understanding of the underlying process properties and it is necessary to correctly classify and identify behaviour not experienced during normal operations. Venkatasubramanian

Method	Advantages	Disadvantages	Examples
Quantitative	Fast, robust, efficient	Require accurate identification of models	Observers, Parity Space, Kalman Filters
Qualitative	Simple, can provide fault explanations	Often limited to few fault hypothesis	Fault Tree, Causal Models, Qualitative Physics
Process History	Accurate process model not usually required	Require large amount of quality training data	Expert Systems, Neural Networks, Statistical Classifiers

Table 2.5: General classification of diagnostic methods.

et al. [56]–[58] conducted a comprehensive review of different diagnostic strategies and identified three broad families of methods that are available for system's designers. These differ in the way process knowledge is represented, for instance with the support of analytical models, statistical analysis or historical experience, and how search strategies are used to make decisions on the state of a system. An overview of FDI methods is shown in Table 2.5.

Method	Advantages	Disadvantages
Parity Equations	Suitable for additive faults, small computational effort	May show less robustness to external disturbances
State Estimation	Robust to external disturbances, balanced reaction times	May require more effort for non-linear processes
Parameter Estimation	Suitable for multiplicative faults, requires only knowledge of model's structure	Larger computational effort as on-line estimation is required

Table 2.6: Overview of *quantitative* diagnostic methods using a *model-based* approach.

The first set of methods is known as *quantitative model-based* [56], where the monitored system is compared against its analytical representation, such as a mathematical functional relationship between process's inputs and outputs. The second one is known as *qualitative model-based* [57]. In this case the *a priori* knowledge is represented by means of qualitative expressions, such as fault trees, causal models or abstraction hierarchies. The third one is known as *process history based* [58], where knowledge is derived from analysis of past system's behaviour without relying on a specific understanding of the underlying input-output relationships. A detailed classification of *quantitative* diagnostic methods is found in Gertler [59] and Isermann [60] where industrial applications are also described. In this case diagnostic strategies are broadly classified in *model-based* and *model-free* approaches,

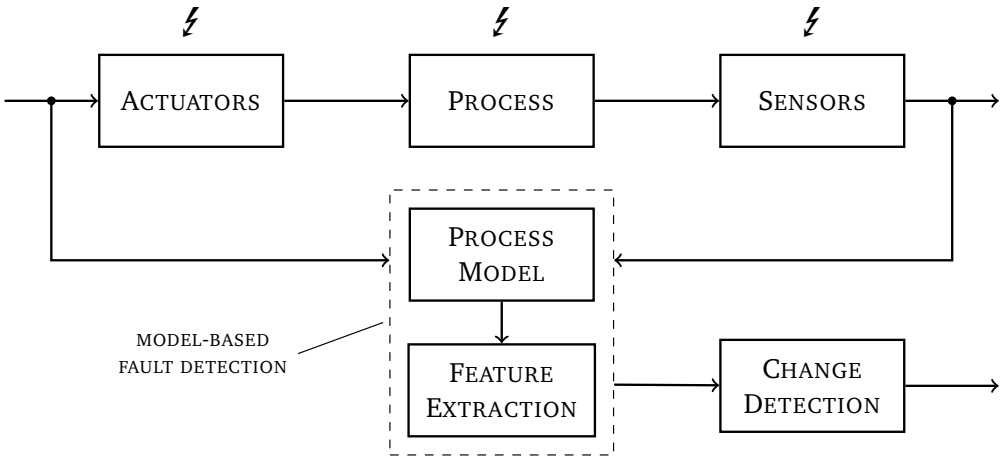


Figure 2.4: Generalised schema for a *model-based* diagnostic system. Failures can be related to the actuators, the sensors or, in specific cases, to the process itself. These are detected, for instance, by comparing the expected and real behaviour using an analytical model for the process under analysis.

highlighting the importance of analytical representations for the implementation of real-time diagnostic systems.

Model-based approaches often represent the methodology of choice for diagnostic system designers. This is because of the availability of much experimental work in different type of domains, for instance, in the context of aerospace or automotive industries and, more in general, of industrial process control. Detailed reviews are found in Isermann [61] and Hwang *et al.* [62], where different subsets of methods are discussed together with their applications on real systems. Table 2.6 shows a comparison among common fault detection methods. A general schema for a model-based diagnostic system, instead, is shown in Figure 2.4. This highlights the principal elements of typical systems such as fault detection and fault diagnosis aspects. The latter is often implemented in a *change detection* module that operates on the features extracted during the detection phases. This element is responsible for the diagnosis task that extracts information about a given fault, such as its size (or severity), location, time, and correctly identifying its type. Two main methodologies are usually followed when implementing such a module: *classification methods* and *inference methods*. The former relies on a set of techniques such as statistical classification (decision trees, Bayes classifiers), pattern recognition, approximation methods or artificial intelligence methods (fuzzy or neural network classifiers). The latter, instead, relies on binary or approximate reasoning and tends to rely on expert knowledge to correctly recognize a specific fault.

In the marine domain relevant work on fault detection and diagnosis topics has been developed in the past focusing around the propulsion subsystem. Previous studies show, in fact, that the runtime detection of unwanted behaviours is a key aspect to guarantee reliable and safe operations when limited or no human intervention is available. Such systems find successful application in different operational environments from Remotely

Operated Vehicles (ROVs) [63] to larger marine vessels [64].

2.4.1 Model-based Methods for Underwater Systems

In the context of autonomous underwater systems several model-based strategies have been proposed to estimate the health status of the on-board actuators. These are usually focussed on a common set of actuator's control and feedback signals, such as input voltages, throttle commands, propeller speeds, load currents. They differ, on the other hand, in the way fault information is extracted (e.g. selecting specific *features*) or in the way decisions are taken (e.g. using alternative *change detection* methods) about the presence of failures. Other differences are in the number of tuning parameters or in the presence of extensive validations in real sea conditions.

Advanced diagnosis techniques have been investigated in several works [65]–[68]. These tend to provide a good trade-off between accuracy and precision in identifying multiple fault hypotheses and for this reason they are objectives of interest in the autonomous system community. In the work of Alessandri *et al.* [65] the model's dynamics are taken into account in a diagnostic loop closed around the vehicle's motion. A series of exclusive filters evaluate the platform's behaviour under different fault hypotheses and residuals are evaluated to detect the presence of a specific fault condition. Such an approach has been validated in controlled conditions demonstrating its effectiveness. The work of Antonelli *et al.* [66] is based, instead, on the use of Support Vector Machine (SVM). This core functionality of the proposed system allows it to benefit from the robustness offered by a well-understood data-driven approach and, at the same time, to manage the residual uncertainty usually resulting from unmodelled non-linear vehicle's dynamics. On the other hand, this approach requires the use of a clean and an extensive dataset of fault-free trajectories with an off-line training procedure in order to obtain a well tuned system. The contribution of Miskovic *et al.* [67], instead, relies on a Principal Component Analysis (PCA) technique applied together with a simplified thruster model. This strategy offers the capability of recognizing a broad range of faults classifying their typology using a specific set of logical relationships. Its implementation relies on the availability of a precise statistical model for the control input signals. This aspect, together with the need of codifying expert knowledge as a static set of rules and lack of extensive field validation, makes this method less suitable for implementation on a long-term autonomy platform. The work of Wang [68] suggests the use of a fuzzy neural network (FNN) to model the vehicle's dynamics. Detection of faults is done comparing the expected behaviour, suggested by the FNN output, with measurements taken from vehicle's sensors. This approach, while flexible because of its fuzzy representation, is presented only with simulated results. Moreover, together other machine learning approaches, it relies on the availability of good training data for the vehicle's dynamics. On the other hand, a simpler yet effective diagnostic strategy is presented in the work of Hanai *et al.* [55] where the general vehicle's dynamics are not taken into account. In this case a lower level model-based thruster

fault detection scheme is introduced. The actuator's model is derived through a series of simple experiments conducted in controlled environments. Such procedures identify the relationship between input and output signals for each vehicle's actuator. Knowledge of these models provides an effective way to assess the correct behaviour of propulsion system and to detect problems even if small deviations from nominal characteristics are introduced. This aspect represents an advantage compared to other architectures as small and subtle changes in the propulsion system are more difficult to detect if taking into account a larger motion model. In fact, underwater dynamics are of non-linear nature [69] and highly affected by environmental disturbances or measurement noise.

More in general experimental validations of fault detection systems during field missions [18], [55] suggest that model-based techniques represent good candidates for building a real-time diagnostic system that monitors the effectiveness of propulsion subsystems. Such methodologies, even if dealing with multiple actuators, do not require heavy computations and can be efficiently integrated inside control software architectures.Such an approach offers accuracies proportional to the quality of underlying models and the precision of on-board sensors. Furthermore, it does not employ parameters that require tuning for a specific type of underwater mission. These aspects give model-based strategies the required robustness and effectiveness to be integrated in autonomy architectures for long-term deployments.

2.4.2 Alternative Methods and Operational Experience

Other fault detection strategies [60] rely on model-free methods, such as spectrum analysis, limit checking or, more in general, qualitative approaches. While still of relevance from a diagnostic point of view their application to the AUV domain requires extensive tuning, expert knowledge and a strict set of parameters that limits their scope to a specific vehicle's configuration or environment. This is in contrast with the flexibility often required by operators, for instance, allowing different payloads to be exchanged between consecutive deployments. Moreover, on-the-fly mission reconfiguration and automatic tuning capabilities are often desired features that are currently the objectives of research in the context of long-term autonomy operations.

An example of a model-free method is the limit checking approach. This analyses a set of signals and it applies a series of checks, for instance, on upper and lower bounds, to detect the possible occurrence of a fault condition. Such a methodology, while not requiring design efforts and being easy to implement, has a limited fault specificity and its requires a reduced sensitivity to accommodate the expected variations (e.g. system noise) in monitored signals. This strategy is often employed at higher system levels, for instance, in the software domain, or when designing hybrid diagnostic systems (e.g. combining multiple strategies). This methodology found often application together with a classification method known as fault-tree analysis (FTA). This technique relies on expert knowledge and analysis conducted during the system's design to represent the

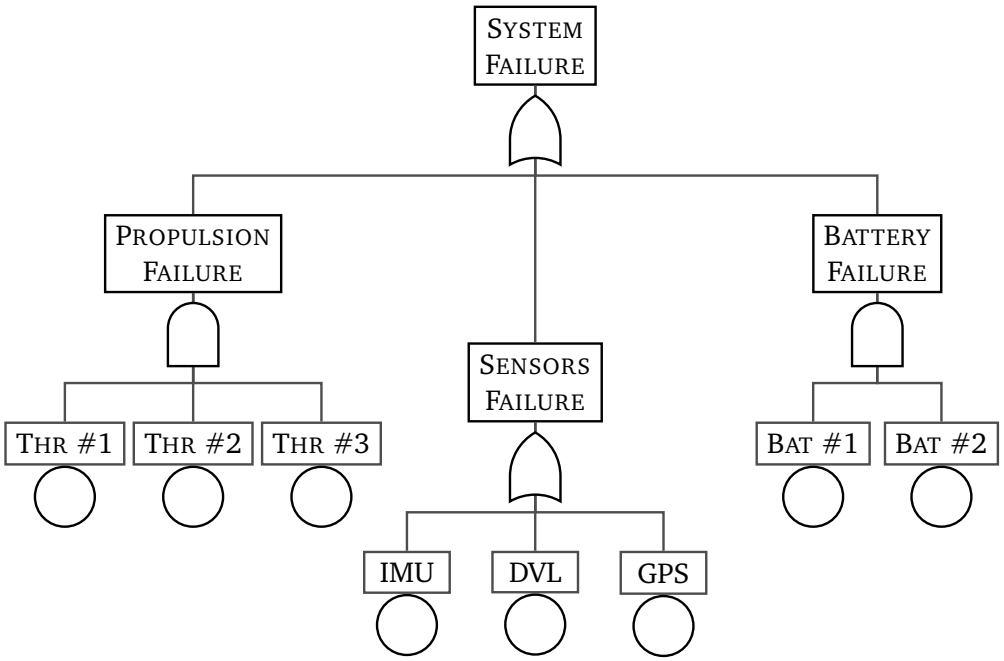


Figure 2.5: Example of fault-tree diagram for an underwater vehicle. Three main subsystems are represented in the tree along with their principal components. Logic relationships (AND, OR) define how faults are propagated from components to the overall system.

relationships between components and failures in terms of hierarchical trees. An example of FTA applied to a generic AUV system is shown in Figure 2.5. This highlights the dependencies among principal components (sensors, propulsion, battery) and introduces logical functions (AND, OR, XOR, etc.) to describe how faults can propagate through the AUV at runtime. Such an approach is often not enough to capture all the possible failure modes that real hardware components may show during their life-time. This is because a comprehensive representation of all fault possibilities makes the fault-tree grow exponentially. On the other hand, FTA and limit checking techniques are often employed in hybrid diagnostic systems that make use of a *strong* method (e.g. quantitative model-based) together with a simpler (or higher level) method to represent the overall system's state.

More general strategies have been explored in the area of integrated health management and control. Relevant work has been done in the context of ground [70] and aerial [71] autonomous robots. In these domains availability of affordable sensors and high communication bandwidths allow designers to develop distributed schemas where some diagnostics and mitigation modules are implemented outside the vehicle's architecture (e.g. in ground control centres or with the support of larger cloud platforms). Such an approach is often not a viable alternative in the underwater domain. This is because of limitations with the acoustic communications and the relative short coverage of low latency communication links in the maritime domain. Those aspects suggest the development of integrated on-board architectures [72] where most of the sensors data has to be processed by the vehicle itself in order to assess its health status. Such decisions can then

Component	Failure	Severity	Probability
Battery System	Failed to charge	minor	occasional
	Protection fuse blown	moderate	occasional
	Not sourcing current	moderate	remote
	Not switching on	moderate	remote
Propulsion system	Loss of propeller speed	minor	occasional
	Thruster stall	minor	occasional
	Thruster flooding	critical	remote
	Entanglements (sea weed, objects, lines, etc.)	moderate	frequent
Sensors	GPS poor accuracy	minor	frequent
	GPS lost initialization	minor	frequent
	GPS antenna fail	moderate	remote
	DVL poor accuracy	moderate	occasional
	DVL failure	critical	remote
	IMU failure	critical	remote
Control Surfaces	Not responding to commands	moderate	occasional
	Intermittent drive faults	moderate	occasional
	Actuator feedback fault	moderate	occasional
	Actuator offset from zero	minor	occasional
Communications	Antenna failure (WiFi, Iridium, GSM, etc.)	critical	remote
	Loss of connection (intermittent, reduced bandwidth)	minor	frequent
	Acoustic modem failure (low sensitivity, errors)	minor	occasional
Science Payload	No sonar data collected	minor	remote
	No CTD data collected	minor	remote
Emergency Abort	Failure to abort	critical	remote
	Premature abort	moderate	remote
Software	Collision avoidance failure	critical	remote
	Control failure (remote control, telemetry, etc.)	minor	occasional
	Navigation failure (loss of guidance, map drift, etc.)	moderate	occasional
	System exception (software error, restart)	minor	occasional
Others	Power switch failure	critical	remote
	Short due to failure (mechanical, electrical)	moderate	occasional

Table 2.7: Overview of common observed failures on AUVs during long-term deployments.

be shared over low bandwidth acoustic communication links [10] to notify neighbouring vehicles and human operators about the effective capabilities of a given autonomous unit.

An interesting aspect of the operational experience collected in the context of long-term missions [18], [73], [74] is the identification of a broader range of possible system's failures for underwater platforms. Table 2.7 shows the most common problems seen in practice and highlights their impact on the vehicle's effectiveness and their occurrence's probability. Such a list is only a reminder for the large number of possible fault conditions that a proper diagnostic system should handle in order to provide autonomous platform good levels of self-assessment and fault management capabilities.

2.5 Fault Mitigation

Beside detection and isolation aspects, other relevant work has been done in the area of mitigation and fault accommodation. This practice is often referred in literature as Fault Detection, Isolation and Recovery (FDIR) and it can be considered as natural extension of previous diagnostic techniques with the capability of reconfiguring a system after the identification of problems. In the autonomous systems domain such procedures are often

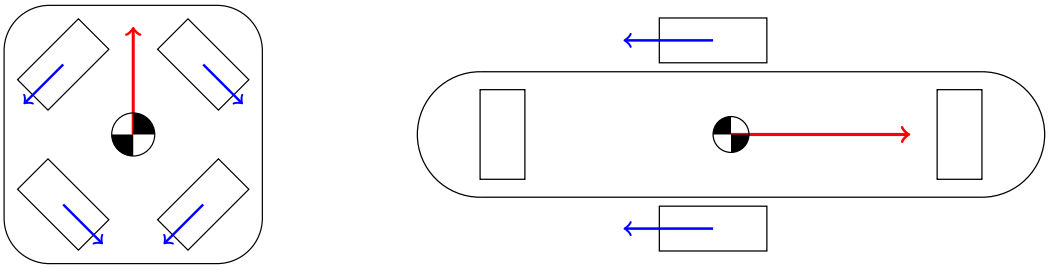


Figure 2.6: Example of thruster configurations. An X-shaped configuration (left) uses all thrusters to provide locomotion along the vehicle's principal axis. A traditional approach (right) uses pairs of thrusters directly along those axis and it does not fully utilise the set of thruster for cruise control. Lateral motion, used for manoeuvring, is controlled with a different subset of actuators.

implemented with changes in the vehicle's operating conditions, for instance by selecting a different navigation mode or an alternative use of available actuators, in order to reduce the impact of faulty components on the correct vehicle's behaviour. The feasibility of such an approach is highly affected by the platform's design and, in the case of component failures, by the presence of hardware redundancy.

An example of this is the use of redundancy in the propulsion subsystem, for instance, in a vehicle designed with an X-shaped actuator configuration. In this case each thruster has a 45 degrees offset from the vehicle's principal axes. Such an approach relies on the simultaneous use of actuator's pairs to produce the necessary thrust to move an underwater vehicle. This is shown schematically in Figure 2.6. A similar choice represents a trade-off between control capabilities and platform's survivability in presence of actuator failures. Focusing on this subsystem relevant work has been conducted in the past for optimising the use of the remaining actuators in the event of thruster failures. Such approaches research alternative *thrust allocation* solutions by remapping the distribution of generated forces among the efficient actuators with the intent of partially restore the platform's control capabilities according to the original vehicle's design.

2.5.1 Classical Control Strategies

A similar mitigation strategy has been initially proposed by Yang *et al.* [75] where an experimental study has been conducted relying only on the thruster redundancy. Later, the contributions of Sarkar *et al.* [76] and Omerdic *et al.* [77] reviewed the original

strategy with the use of weighting coefficients for actuator's forces and introduced a more generic formulation for underwater systems. The resulting mitigation strategy relies on a proportional redistribution of forces done according to the original thrust allocation schema among available actuators. A general example of control architecture is shown in

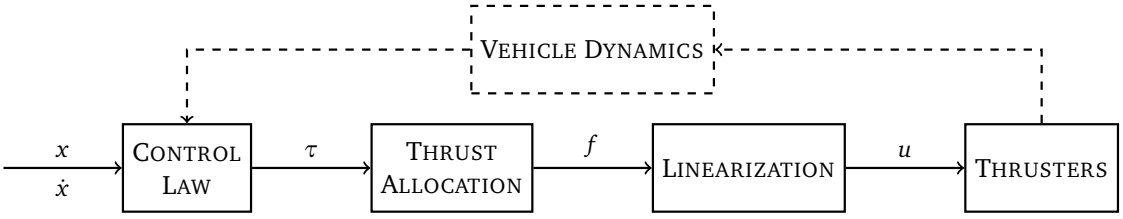


Figure 2.7: Overview of control schema for a generic autonomous underwater vehicle. Desired position x and $\dot{x}$ velocity vectors are provided to a control law that calculates the generalized thrust vector τ. An allocation procedure maps this to actuator's forces f. Finally, a linearization procedure converts forces in low-level thruster's commands u.

Figure 2.7, where the main components, such as *control law*, *vehicle dynamics*, *thrusters*, are represented.

The *thrust allocation* module takes care of mapping the generalized force vector τ into local actuator forces u. Mitigation strategies, often, operate at this stage preventing failures from affecting the vehicle's navigation performance. In such cases a modified architecture is introduced, for instance, as in the FDAS approach [77], by replacing it with a pair of accommodation and diagnosis modules. A possible resulting schema is shown in Figure 2.8. The concept of *thruster efficiency*, on the other hand, has been introduced and coupled with a diagnostic framework in the contribution of Hanai *et al.* [78]. This represents a comprehensive FDIR schema where a diagnostic system coupled together with a mitigation strategy is implemented in a real AUV. In this work the diagnostic part provides health status estimations for all vehicle's actuators. Such estimates are then used to dynamically adjust the force's redistribution weights in the event of fault detection.

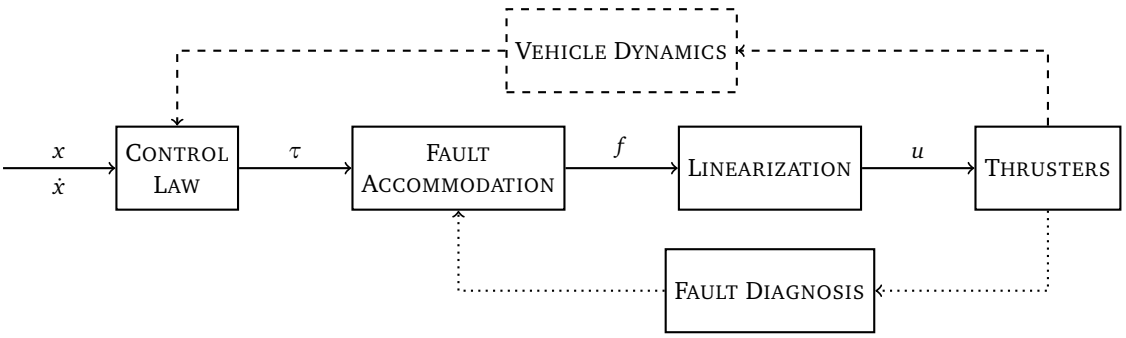


Figure 2.8: Overview of control schema with fault accommodation capabilities. A diagnosis module is added to the general schema to monitor the actuators' control feedback. A modified allocation procedure utilise the diagnostic output to provide accommodation in presence of failures.

In the work developed on ODIN AUV [76] the introduction of weighting requires the

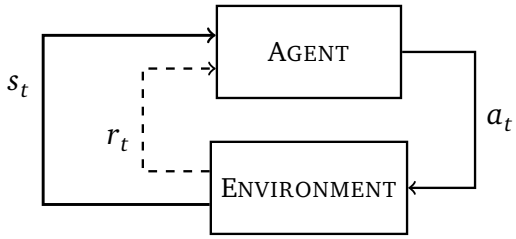


Figure 2.9: General schema of *reinforcement learning* (RL) methods. An agent acts on the environment using an action a_t. Reward r_t and state update s_t are propagated back to the agent after interacting with the environment. Reinforcement learning approaches aims at finding an optimal sequence of actions that maximises the overall return ($R = \sum_t r_t$) of the generated behaviour.

use of a modified controller schema to prevent thruster saturation. This aspect, while solving elegantly the problem of saturation, introduces some restrictions in the design of customized control architectures. An example of those is seen in [79], [80] where a common control schema has been adapted to a specific operating mode that an AUV may encounter during its long-term operations. In such cases the fault adaptation should be attempted even if in presence of multiple control strategies, selected according to the current task. Furthermore, in the work developed on the SAUVIM AUV [78] the integration of a fault accommodation system with an higher level software architecture (e.g. autonomy modules) is not discussed. Two other aspects are also left aside in the above works: the effects of force redistribution on the navigation performance and on the energy usage profile [81] of the autonomous platform. These aspects, while not of primary importance from the control point of view, influence the behaviour and effectiveness of vehicles over the duration of long mission. Incorporating these concepts into a unified architecture can help decisions taken by higher level autonomy modules, for instance, at planning and mission execution levels. This, which represent an active area of research [82], enables modern autonomy architectures to adapt and optimise on-the-fly the current execution plan in terms of resource usage and task selection during long-term missions.

2.5.2 Alternative Strategies

Other fault-tolerant techniques exist and do not rely on classical control methods. In those cases mitigation solutions are found using machine learning (ML) or evolutionary strategies [83]. These methodologies, usually, rely on the availability of good quality training data, for instance, recorded from human operations in *learning by demonstration* studies or collected with specific controlled environment experiments. An example of ML approach for intervention AUVs is the use of *reinforcement learning* (RF) techniques as shown by Ahmadzadeh *et al.* [84], [85]. A general schema for reinforcement learning approaches is shown in Figure 2.9. These works introduce a system to learn a new control policy in case of actuator's loss. Such a strategy is effective in recovery from hard failures. Experimental validations shows how the platform manages to adjust its

navigation capabilities to the new operational state. In those specific cases this is achieved with changes in navigation heading and with the use of lateral motion and more complex trajectories. On the other hand, RF techniques require a great deal of simulation and experimentation before a valid policy is derived for a specific fault condition. Such a constraint is not always applicable for vehicles operating at sea because of the great range of unexpected events that can possibly be encountered during a long-term mission. Further more immediate reactions are often needed, for instance to avoid dangerous behaviours, or when a minor failure or degradation is detected.

2.6 Mission Reliability

Another topic of interest when analysing the *effectiveness* of autonomous systems is the concept of *reliability* applied to mission management. In fact, FDIR capabilities alone are not often sufficient to guarantee the correct development of long duration missions executed without human supervision. Those methodologies, while necessary for assuring quick reactions to sudden events, are not responsible for researching or implementing long-term accommodation strategies that should be taken into account in presence of unexpected scenarios (e.g. failures) or changes in the operating conditions not originally considered in the mission plan.

Different aspects are often investigated when approaching those issues. One is the analysis of failure's effects on the mission's development (e.g. *can the platform continue with future tasks?*) as well as the evaluation of original goal's viability (e.g. *can still the mission's objective be achieved?*). Several studies and architectures [86] have been investigated on this topic focusing on providing an effective way of assessing the platform's reliability in conducting complex missions. An example of this is the phase mission reliability analysis [87] approach studied in the UAV domain. This methodology aims at evaluating those effects by combining smaller fault-tree analyses for each operational phase that compose a long or complex autonomous mission. Such an approach allows for a quick numerical evaluation. In the underwater context, on the other hand, field missions have been evaluated using Markov-chain models [74], where sequential missions (often found in this domain) are described as a set of discrete states with respective state transition probabilities. For example, a mission can be represented analytically as:

$$Pr(X_{n+1} = x \mid X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = Pr(X_{n+1} = x \mid X_n = x_n) \tag{2.28}$$

where $X_i \in S$ and S is the set of possible states, also known as the *state space*. The (2.28) is usually defined under the assumption that conditional probabilities are well defined. Such an approach allows simplified analysis of the mission by considering the probability of moving onto a next state as conditional only on the present state. The aim of such a strategy is to correlate the mission's states with a risk and survivability analysis derived from previous operational experience as well as expert knowledge. An

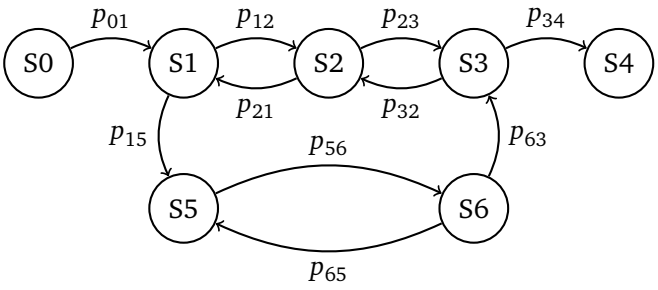


Figure 2.10: Markov-chain representation for a generic mission process. State S0 and S4 represents the *initial* and *recovery* phases of the mission. States S1 to S3 are associated with general control tasks (e.g. diving, surfacing, trajectory following). States S5 and S6 with target recognition and identification tasks (e.g. sonar processing, target reacquisition).

example of a possible Markov-chain representation for a simple underwater mission is shown in Figure 2.10. In this case a plan with 7 states is taken into account. The *initial* ($S0$) and *recovery* ($S4$) phases are identified as edge cases while *intermediate* states ($S1, S2, S3, S5, S6$) are connected together in a cyclic graph representing the sequence of repeated operations usually performed by an underwater vehicle. This approach, while of relevance, is difficult to apply in architectures that employ more complex planning [1], [2] or optimization strategies, where often, a degree of uncertainty is embedded into higher level planning procedures. In those cases neither Markov-chain models nor expert knowledge can exhaustively handle all the possible outcomes that an autonomous platform can encounter during its operations.

Another methodology that evaluates the *reliability* of autonomous mission is the analysis of its evolution when the availability of on-board resource is scarce or uncertain. An example of relevant work in the underwater domain is the study of evolutionary techniques for adapting on-the-fly survey missions [88]. This work highlights how energy management issues are a relevant aspect to guarantee reliable operations when away from human supervision in unknown environments. Such a characteristic puts underwater vehicles in the same context of other mobile systems, where relevant studies [13]–[15], [89]–[91] have been conducted extensively on energy management aspects and on the effect of environmental disturbances on the reliability of missions. In the works of Saha *et al.* [89] and LeSage *et al.* [14] effort has been put, first, in understanding and modelling the effect of different discharge or load profiles on the behaviour of stored energy resources in the context of realistic missions. Such aspects are relevant if the mobile system is capable of carrying out complex tasks (e.g. survey areas, manipulating objects, navigating in cluttered environments) and a wide range of energy demands are usually experienced during their development. Those studies investigate how the usage of energy resources is influenced by the health status of on-board batteries and by specific mission profiles. For instance, frequent executions of highly demanding tasks (e.g. take-off, landing, diving, surfacing, etc.) result in a reduced availability of energy resources due to an excessive stress on the battery system, leading, in extreme cases, to premature termination of

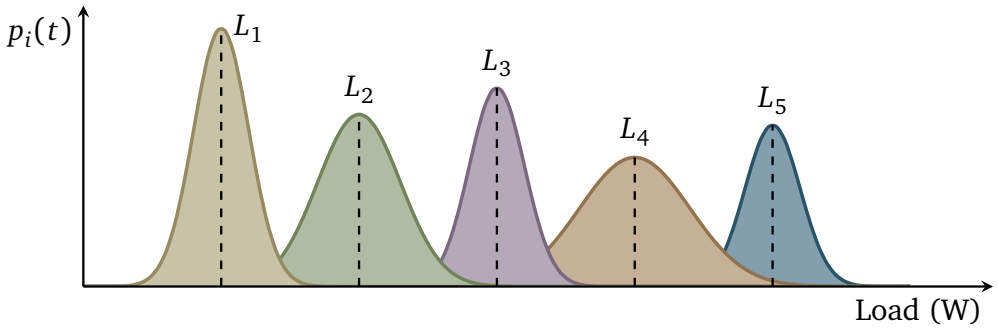


Figure 2.11: Example of load distribution for an autonomous vehicle (adapted from [14]).

missions.

A detailed analysis [14] has been carried out on UGVs's systems using identification procedures to isolate and quantify different types of power load configurations experienced during field missions. Advanced platforms, in fact, can be fitted with a broad range of auxiliary sensors, complex algorithms and on-board computer systems that can process great amount of data while operating in the field. Figure 2.11 and Table 2.8 show an example result for the proposed methodology. Different clusters are identified analysing the platform's performance when deployed in different operating scenarios. This loads classification, together with occurrence probabilities derived over the course of typical missions, allowed researchers to obtain statistical models (e.g. Gaussian Mixtures, Markov processes) that can be used to simulate the possible future evolution of realistic missions. Such an approach is then used to estimate the effective residual time that on-board battery system can sustain when experiencing mission-like stimulations.

Cluster	**Name**	**Mean** (A)	**Std** (A)
L_1	Hotel Loads	1.0	0.50
L_2	Sensors	2.3	0.65
L_3	Navigation	3.2	0.40
L_4	Navigation + Sensors	4.1	0.75
L_5	Diving + Sensors	5.0	0.45

Table 2.8: Example of load clustering for an autonomous underwater vehicle.

Other efforts have been focused in estimating the mission's energy requirements [13] using real-time measurements and *a priori* knowledge gathered from previous deployments and comparing it to the available on-board resources. This strategy represents an useful technique because real field operations are often affected by external disturbances or temporal changes of the mission environment that require constant adjustments of on-board predictions. Such high variability in the operating conditions is, often, enough to invalidate most of the original assumptions made during initial planning phases. Failing to take into account those aspects often results in shorter mission times, vehicles getting stuck away from recovery zones and, more in general, the necessity of introducing large

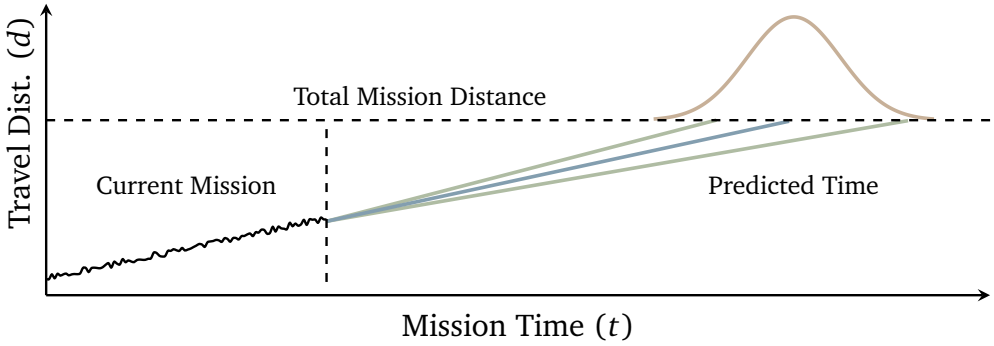


Figure 2.12: Example of travel time prediction for UGV missions (adapted from [15]). Process history is used to characterise the mission's behaviour, for instance, calculating the probability density for the expected locomotion time with respect to the mission's travel distance.

safety margins in the resource planning phases.

A further extension of the load characterization study introduces the use of energy estimations in evaluating the feasibility of an autonomous mission [15]. This procedure is carried out while a vehicle is operating in the field by correlating the available energy resources with requirements derived from its current mission plan. Among those are the estimation of vehicle's remaining mission time, that in case of ground systems operating on unknown terrains is a function of many factors (e.g. driving style, friction, wheel grip, etc.) as shown, for instance, in Figure 2.12, and the prediction of residual battery runtime (i.e. time before on-board resources depletion) using samples collected during mission's execution. Experimental results on UGVs validate the use of such a methodology in proving a more robust assessment of autonomous missions. Altogether, the work done on UGVs and UAVs suggests how the use of runtime energy frameworks is beneficial to quantify effects of the external environment on the mission's execution, both in term of feasibility (e.g. the probability of successfully completing the current plan) and estimation of future resource usage.

2.6.1 Dynamic Environments and Mission Planning

In the underwater domain vehicles operating in presence of dynamic environments and unknown disturbances are a well-known problem. Relevant work has been done in the past to handle this operational uncertainty using different approaches. Generic and more comprehensive strategies include the use of extensive semantic frameworks, both to address the presence of unexpected events [20] (e.g. failures) and to improve the vehicle's situational awareness [10] in presence of complex missions.

On the other hand, more specific work aims at improving the planning of AUV missions by embedding environmental effects and user's constraints in the problem formulation. In this domain the concept of on-line mission adaptation [88] has been explored for adjusting the mission's operational parameters (e.g. duration, survey area, speed) in

presence of resource constraints. Further attention has been given also to the concept of contingency planning [92], where the use of overly-conservative plans is discussed together the implications on suboptimal use of on-board resources. Finally, an example of planning for autonomous inspection tasks [82] is discussed in presence of temporal constraints. A relevant application in the same context is the problem of autonomous intervention [93], where a vehicle is envisioned to replace divers or remotely operated systems to interact with subsea panels.

More generally, the problem of planning [94] is often extended to the underwater domain [95] using the experience gained in other domains, such as mobile ground systems and aerial robotics. Solutions are often researched using probabilistic approaches [96] (e.g. using a Markov decision process) by employing reinforcement learning [97] strategies or more classic dynamic programming techniques.

Path-Planning Problem

Beside global mission planning aspects other relevant work has been done on the path-planning problem [98]–[102], where evolutionary approaches [98], search procedures [99] or gradient based techniques [100] are discussed in presence of ocean current maps and large variable environments. The use of adapting sampling and mixed integer linear programming is discussed in [101] while energy-aware aspects in context of dynamic path-planning are also mentioned in [102]. All of these works highlight the necessity of including environmental effects in the planning domain of long-term deployments as their analysis can substantially improve the vehicle's effectiveness when operating away from human supervision. Furthermore, work done on path planning shows how navigating along optimal paths improves the vehicle's efficiency in terms of resource usage and how this contributes to extend the vehicle's persistence out in the field.

2.6.2 Probability of Completion

A common approach in analysing reliability aspects is to introduce a compact representation for the underlying processes that characterise a given dynamical system. This allows the definition of the reliability integral (2.29): an expression that quantifies the expectation of a system working correctly and in normal conditions. The integral [103], [104] is generally defined over a set of edge conditions which are known to trigger a failure or to exceed the system's specifications:

$$R = 1 - \int_{\mathbb{F}} p(\theta) d\theta \tag{2.29}$$

where $\mathbb{F}$ is the failure domain defined by a limit state function $g(\theta)$ as $\mathbb{F} = \{\theta \in \Theta : g(\theta) \leq 0\}$ and $p(\theta)$ is the probability density function of θ. Several numerical methods [105] exist to evaluate this expression and often combine domain knowledge with Monte

Carlo strategies to both quantify the probability terms and to identify the failure domain. Alternatively to this expression it is common to formulate such analyses focusing only on the right-hand side of (2.29). This is known as *probability of failure* (P_F) and it is defined as:

$$P_F = \int_{\mathbb{F}} p(\theta) d\theta \tag{2.30}$$

where the integral term is solved on a domain exceeding the failure boundary. In a general case θ represents a random vector of multiple variables and $p(\theta)$ takes the form of a joint probability density.

In the autonomous systems domain underlying processes of interest are, for instance, defined at mission or task levels, according to the level of detail and the scope of supporting analysis. In the case of mission level abstractions the process is described as a sequence of intermediate steps (e.g. actions or tasks) that platforms need to carry out when trying to reach their specific goals.

$$M = \{A_0, A_1, \ldots, A_n\} \tag{2.31}$$

Failure domains are identified, for example, as combinations of limits derived by the use of on-board resources (e.g. battery energy, fuel consumption, availability of recharging stations, etc.), the allocation of temporal or spatial resources (e.g. maximum mission time, coverage areas, etc.) and user-defined operational constraints. Two common variables, used in previous works [13], [15], are the *mission time* (t_m) and *mission energy consumption* (e_m). Considering just these two variables the *probability of failure* can be expressed in the form of:

$$P(t_m > t_T, e_m > e_T) = \int_{t_T}^{\infty} \int_{e_T}^{\infty} f_M(t_m, e_m)\, dt_m de_m \tag{2.32}$$

where f_M is the joint probability density (given t_m, e_m), derived from the mission process M (2.31), and the parameters t_T, e_T are the upper-limits, respectively, of mission time and energy consumption. When considered from the point of view of reliability the (2.32) is then expressed as:

$$R = 1 - P(t_m > t_T, e_m > e_T) \tag{2.33}$$

and it is referred in literature [15] as *Probability of Mission Completion* (PoMC). This characterises the chance of successfully conducting the underlying mission M within the platform's energy and time constraints. Such a formulation implies the temporal dependency of the f_M term. This, in fact, is usually determined by several factors (e.g. changes in the environment) and by the presence of advanced decision making processes (e.g. on-line or adaptive planning) that can modify on-the-fly the process itself. An example of the joint density term, calculated for a generic sequential mission of repeatable tasks, is shown in Figure 2.13. In this case f_M follows a bivariate normal distribution:

$$f_M(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} e^{-\frac{z}{2(1-\rho^2)}} \tag{2.34}$$

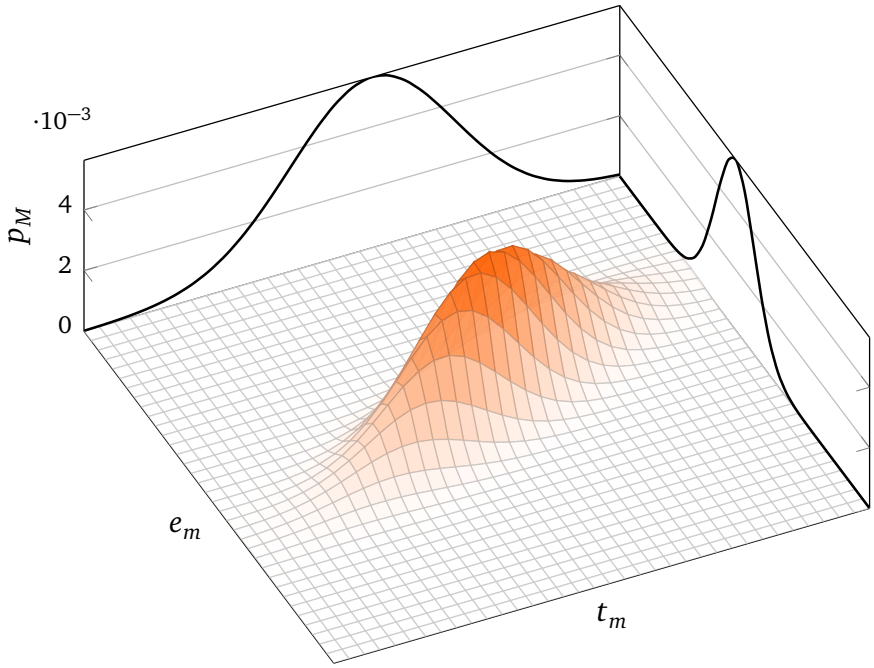


Figure 2.13: Example of bivariate normal probability density $f_M(t_m, e_m)$ for a generic mission process with repeatable tasks. Uncertainty of energy consumption and time duration characterise the shape of the surface with respect to a correlation coefficient ρ.

where z is a location term defined as:

$$z = \frac{(x_1-\mu_1)^2}{\sigma_1^2} - \frac{2\rho(x_1-\mu_1)(x_2-\mu_2)}{\sigma_1\sigma_2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2} \tag{2.35}$$

and ρ is the correlation coefficient between random variables, also defined as:

$$\rho = \mathrm{Cor}(x_1, x_2) = \frac{\mathrm{Cov}(x_1, x_2)}{\sigma_1\sigma_2} \tag{2.36}$$

In the case of PoMC the above model is identified for the variables t_m, e_m by the parameters μ_t, μ_e and σ_t, σ_e, respectively, means and variances of mission time and energy usage. Those values are usually derived using models formulated for intermediate tasks A_i that compose the mission process M. The identification of the coefficient ρ, instead, is a process that requires experimentation[1] and, as shown also in previous works [15], is affected by operational uncertainty[2] and external disturbances.

2.6.3 Range and Energy Evaluations

Availability of accurate state estimation techniques, for example in the battery context, allowed researchers in the mobile systems community to devise several methodologies

[1] Both in terms of simulations (e.g. using Monte Carlo methods) and of real mission's analysis.
[2] Uncertainty in autonomous operations is given by non-ideal execution behaviours of intermediate tasks. It can be modelled as additive error for the underlying statistical models of vehicle's actions.

for evaluating the energy requirements and the effective range of battery-powered systems [106]–[108] operating in real scenarios. Analyses have been done, for instance, conducting experiments with human operators, evaluating the effects of their decisions and control styles on the performance of mobile systems, or with the identification of repeated behaviours from previous deployments. These aspects, often underestimated in power-rich scenarios such as indoor or large ground systems, plays an important role in the marine domain where the access to resources, vehicles or support structures is often not practical or totally unavailable.

Previous work on performance evaluation of underwater vehicles has been done by Willcox *et al.* [81]. In this work researchers analysed the behaviour of AUVs conducting scientific surveys in marine environments while deriving simple metrics to calculate their expected energy requirements. Their formulation is effective when power loads can be assumed constant and without variations for most of mission's time. Such conditions are often found in regular data gathering tasks where the vehicle is required to follow regular trajectories and maintaining a constant attitude (e.g. speed, depth, heading, power usage). Considering the survey of a large mission area the energy usage can be approximated[3] as:

$$e_{surv} = \left[\frac{\rho C_d S v_{surv}^3}{2\eta_p} + H \right] t_{surv} \tag{2.37}$$

where v_{surv} represents the survey speed, t_{surv} the survey duration, η_p the propulsion efficiency, C_d the drag coefficient, ρ the density of water, S the vehicle's surface area and H the platform's hotel load. Knowledge of (2.37) allows planners to correlate energy requirements with operational parameters, like v_{surv} and t_{surv}, often objective of interest for operators when analysing the feasibility of such missions. Another relevant aspect of (2.37) is that it allows finding an approximate expression for the minimum required survey energy given the survey's parameters. In fact, assuming being in presence of a known square survey area A, the total linear distance can be expressed as:

$$L \approx \frac{A}{2\lambda} - 2\lambda \approx= \frac{A}{2\lambda} \tag{2.38}$$

where λ is the survey resolution (e.g. the distance between parallel tracks of a lawn mower pattern) set by operators. In such a case the survey speed is thus expressed as:

$$v_{surv} \approx L / t_{surv} \tag{2.39}$$

and by dividing the (2.37) by L, one gets:

$$\varepsilon_{avg} = \frac{e_{surv}}{L} = \frac{\rho C_d S v_{surv}^2}{2\eta_p} + \frac{H}{V} \tag{2.40}$$

[3]The equation (2.37) does not takes into account any environmental effect, such as sea currents, wind variations and, more in general, the effects of vehicle's manoeuvres.

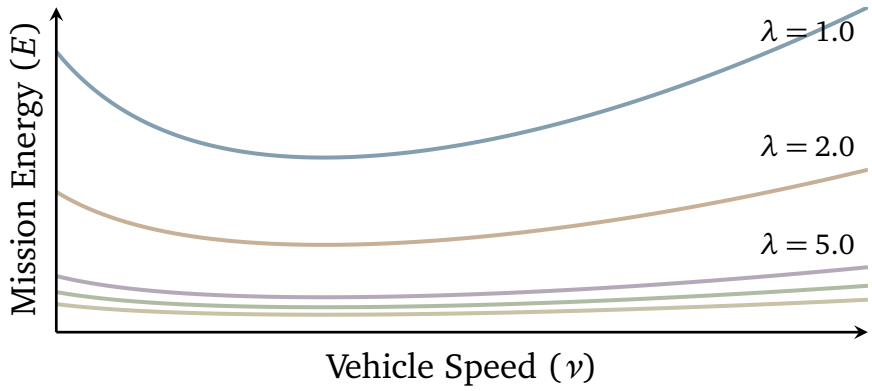


Figure 2.14: Example of survey profiles for a generic AUV. Curves are calculated for different spatial resolutions λ and speeds v within the vehicle's allowed range (e.g. $[0.25, 2.5]$ m/s). Minima are identified once the λ parameter is further restricted taking into account the resolution of on-board sensors.

the expression of the *energy usage per unit distance* (ε), a relevant metric[4] to evaluate the survey's performance. Deriving this quantity with respect to v_{surv} and setting it to zero gives an estimate of the optimal survey speed:

$$v_{opt} = \sqrt[3]{\frac{H\eta_p}{\rho C_d S}} \tag{2.41}$$

The use of (2.41) in the (2.37) allows, thus, the calculation of the minimum energy requirement for the survey under analysis:

$$e_{min} = \frac{3LH}{2v_{opt}} = \frac{3}{2} t_{opt} H \tag{2.42}$$

Figure 2.14 shows an example of resulting survey profiles for a generic AUV surveying a given area A. Curves are calculated from (2.37) by varying the vehicle's speed v within thruster's capabilities $v \in [0.25, 2.0]$ and using different survey resolutions $\lambda \in [1, 10]$. Results shows how fast survey speeds require more energy to complete a mission and, at the same time, a selection of slower ones leads also to a raised energy requirement. Such a behaviour is given by the consequent increase of total navigation time t_{surv} while being in presence of a fixed hotel load H. Despite its exploratory relevance the expression (2.42) represents only a rough approximation of the actual energy requirement, more correctly its lower-bound. Such a quantity finds most use during the initial stages of the mission planning process, where the limited knowledge of operational environments may require the use of rough assumptions or idealised behaviours.

[4] In this case such a metric is averaged along the full length of the survey and represent the overall behaviour of the platform rather then its local performance.

2.7 Summary

This chapter analysed some of the main aspects affecting *reliability* and *survivability* issues of modern underwater platforms. Initially the battery subsystem has been analysed introducing the work done by researchers in the power systems community that provides analytical models to accurately represent the internal status of battery elements. Beside pure modelling aspects, work done in the prognostic and health management domain suggests that more complex techniques are often needed to better characterise the temporal evolution of such a subsystem. In fact, when considered as part of a more complex platform, on-board power sources are affected by specific usage patterns that may affect their efficiency during field operations. Later, after discussing vehicle's power sources, the propulsion subsystem has been also analysed. This, in fact, represents another relevant component that limits the capabilities of unmanned platforms when in presence of failures or other unexpected events, such degradations of on-board actuators.

Energy usage and propulsion systems have been related to *survivability* issues while discussing existing diagnostic and repair methodologies applied in the context of fault detection and mitigation. Those, commonly employed in presence of more complex industrial scenarios, find more and more usage in autonomous robotics applications where advanced or hybrid techniques are often needed to discover, evaluate and respond at runtime to unexpected events that may be encountered while operating in unknown environments and while away from human supervision. Several approaches, such as ones based on machine learning, statistical analysis and unsupervised techniques, are commonly found in literature. On the other hand, in the marine domain fewer applications are reported and model-based techniques are often preferred by vehicle's designers given the constraints of underwater operations (e.g. limited communications) and the ones of underwater platforms (e.g. limited computational capabilities). Along with survivability aspects, *reliability* issues have also been discussed while analysing autonomous missions together with the use of on-board resources and vehicle's capabilities. In this context, more approaches are found for the underwater domain and several strategies are followed by vehicle's designers to make sure complex missions can be conducted efficiently without human intervention. Probabilistic methods are introduced in such a scenario. Some of those aim at modelling the complete evolution of missions, identifying those are a set of known states. On the other hand, work done on unmanned ground vehicles shows also that more focused solutions, built around analysis of vehicle's performance in unknown environments, are a valid alternative to estimate the reliability of a given mission.

Chapter 3

Energy-Aware Architecture

It is important to realize that in physics today, we have no knowledge what energy is. We do not have a picture that energy comes in little blobs of a definite amount.

Richard P. Feynman

This work introduces an architecture focused on improving *reliability* and *survivability* features of modern underwater vehicles. This is focused on the concept of *energy awareness*: the analysis of on-board energy usage and its correlation to the vehicle's effectiveness while operating in the field. Such an energy-aware architecture complements existing autonomy modules by improving the assessment of vehicle's status, both in terms of self-observation, assuring that internal components are working correctly, and analysis of external factors, recognising the effects of mission's environment, such as sea currents, on a vehicle's capabilities.

Internal assessment is achieved with the introduction of an *automatic fault mitigation* framework. This is used for augmenting the vehicle's fault-tolerant capabilities and for detecting the presence of problems within the vehicle's propulsion subsystem. Such a component is especially useful in presence of partial failures, such as performance degradations of on-board actuators, because of their subtle effects on vehicle's control capabilities. In those cases the platform's reconfiguration or task adaptation strategies can be proposed as valid alternatives to premature interruption of unmanned operations.

On the other hand, external interactions are evaluated using a *runtime performance estimation* framework. Such a component monitors the effective vehicle's performance while operating in the field. Such a procedure is needed because assumptions taken before starting a mission may be improved using information collected during execution. In fact, as discussed in section 2.4.2, the operational experience [18] of long-term mission suggests that external disturbances, such as sea currents, and, to some extent, changes in the health status of on-board components are important sources of uncertainty during

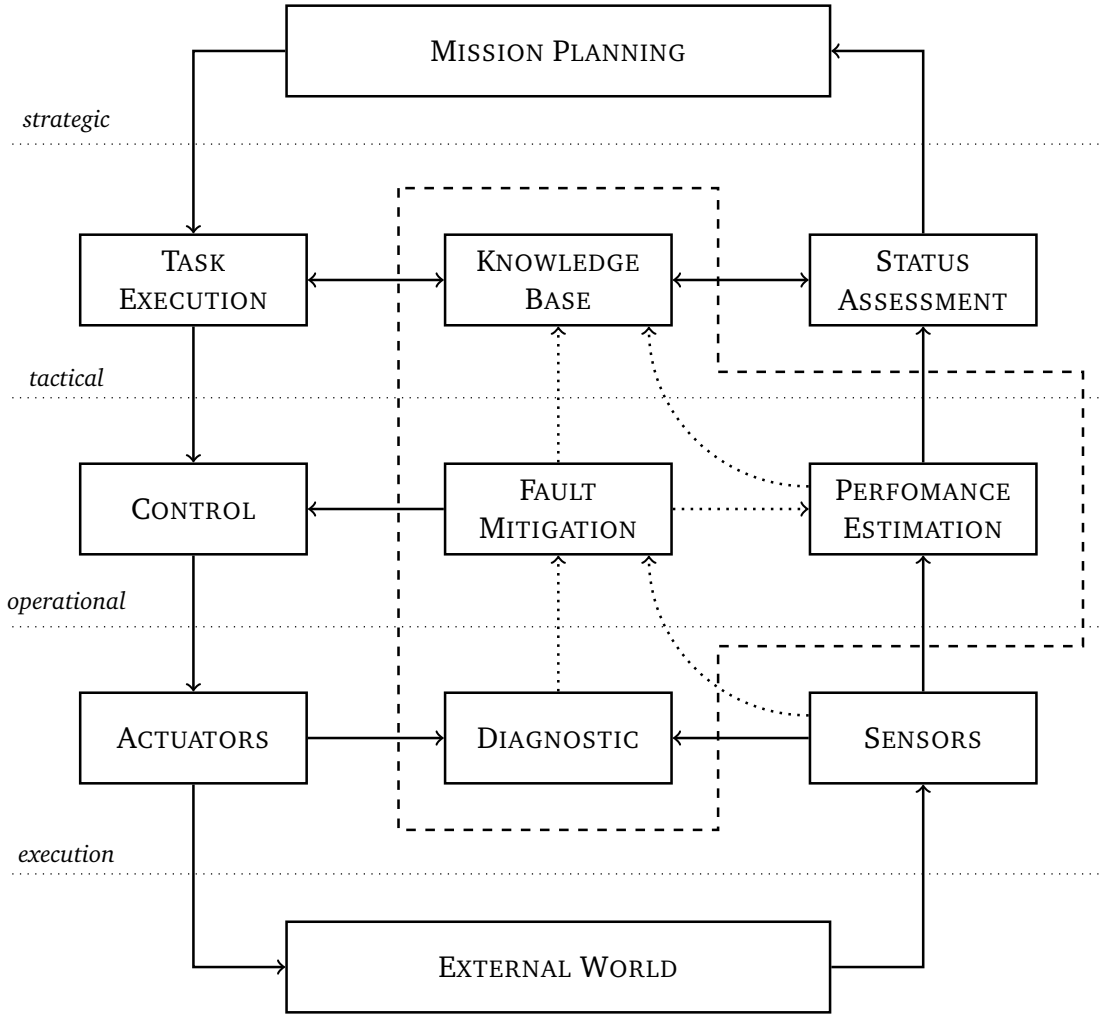


Figure 3.1: Overview of energy-aware architecture. Core elements are delimited by dashed lines. Other modules represent a broader vehicle's autonomy framework. Connections indicate the flow of information within the schema. Energy-aware services are exposed to the autonomy framework at multiple levels of its hierarchical structure.

field operations. Gathering knowledge about those aspects, while conducting a mission, allows the autonomous vehicle to optimise its behaviour, for instance, selecting more appropriate navigation paths or adjusting the sequence of mission's tasks when considering the limitations [88] of on-board resources.

Components of this architecture, when combined together, aim at extending the capabilities of higher level autonomy modules[1] by replacing initial assumptions with correct and up-to-date information about the vehicle's behaviour collected while operating in the field. A possible use for the proposed work is shown in Figure 3.1. In such a schema the *energy-aware architecture* is integrated in a broader autonomy framework, taken, for example, from the PANDORA project [1] that served as test bench for most of the experimental work shown in later chapters. The original schema uses a layered approach to separate logical elements operating at *execution*, *operational*, *strategic* and *tactical*

[1]Those are *task execution* or *mission planning* modules that on-the-fly adjust the vehicle's behaviour in response to unexpected events or changes in the external environments.

levels. Those are characterised by an increasing level of abstraction: transforming sensors measurements into higher level concepts, for example, by employing reasoning modules. Following a similar approach the energy-aware subsystem, shown within dashed lines, is thus integrated in the vehicle's architecture offering *fault mitigation* and *performance estimation* services to existing control and execution modules.

In the following sections the main elements of the *energy-aware architecture* are introduced together with description of their behaviour and the interaction with other autonomy modules. Their design, while following the layered logical approach, also separate fast and slow operations that are executed during the mission. On one hand, fast operations are synchronous with the update rate of vehicle's control architecture, evaluating, for instance, the state of internal components multiple times during navigation. On the other, slow ones are executed asynchronously according to the evolution of mission's tasks. These are, for example, logical evaluation about the effects of detected failures or the research of optimal solutions for locomotion actions.

3.1 Automatic Fault Mitigation Framework

The introduced fault mitigation framework differs from past proposals seen in literature [10], [19], [72] by employing energy consumption as the primary metric for underlying diagnostic methodologies. Such an approach allows automatic runtime operations without relying on a complex set of diagnostic signals. In fact, a small set of parameters, that characterise the behaviour of on-board actuators, is identified through a short self-testing procedure conducted after deploying the vehicle in its mission's environment. Such a framework utilises computational resources available on-board while avoiding the interaction with remote operators and integrating seamlessly with existing vehicle's operations. It makes use of a generic representation for component's health status and failure levels, making it compatible with a broad range of possible vehicles' designs, for instance, with ones featuring control fins or rotatable thrusters for precise navigation or accurate station keeping capabilities.

All those features represent a difference from other fault management systems that often require extensive off-line training and detailed platform's knowledge to be tuned efficiently. Such a characteristic is proper of more complex architectures [10], [72] making those less suitable for on-board implementation on low-cost underwater systems or in cases that have frequent changes in payloads or vehicle's configuration are expected among consecutive missions.

3.1.1 Temporal Approach

Aiming at long-term operations the proposed system analyses the vehicle's behaviour over two different time frames: a short one that is related to the execution of a current task, and a long one, related to the entire duration of the autonomous mission. If failures

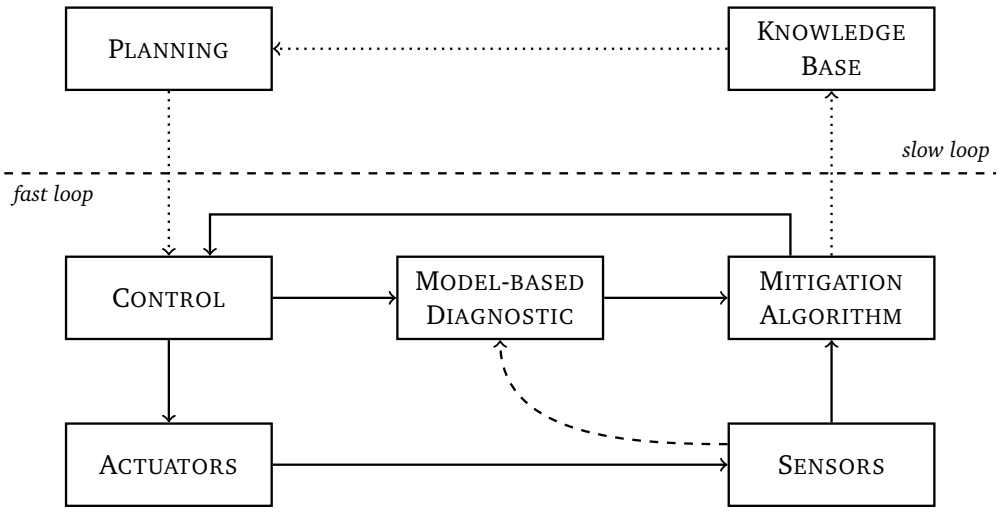


Figure 3.2: Overview of the fault mitigation system. Two feedback loops acts on the control software. A slow loop operates asynchronously to propagate fault knowledge. A fast synchronous one, instead, deals with short-term mitigation reactions.

are detected during normal operations a mitigation strategy is introduced over the short time frame. This allows the vehicle to complete its current task. Concurrently effects of detected failures are evaluated over the duration of a longer time frame. This is done, as shown schematically in Figure 3.2, by propagating the fault's information, collected at runtime, in a knowledge graph that describes relationships among vehicle's components. Such an approach allows the analysis of detected failures together with the evaluation of alternative capabilities that ensure the completion of remaining mission's tasks.

A relevant component for this framework is the *model-based diagnostic* module. This monitors the energy usage of vehicle's actuators and it employs analytical redundancy in order to assess their runtime behaviour. Modelling of these components is achieved with the use of a state-of-the-art non-linear regression technique known as Locally Weighted Projection Regression (LWPR) [109]. Such a methodology is used in this architecture also for performance estimation purposes and it is described with more detail in section 3.3.1. Another relevant component is the *fault mitigation* module. This, upon detection of failures, reconfigures the control software using the estimated actuator's efficiency calculated by the diagnostic module and adapting the vehicle's behaviour to its remaining capabilities. Reconfiguration is achieved with the use of a modified control schema that allows runtime adjustments for the use of on-board actuators. In the case of reconfiguration an optimization procedure is employed for calculating a valid thrust allocation solution that takes into account the constraints of available actuators and prevents them to operate beyond their *saturation* threshold.

Beside modelling and mitigation aspects, a specific feature of this approach is the use of an energy-based diagnostic metric that is derived at runtime using measurements provided by on-board monitoring sensors as further described in Chapter 4. Sensors able of monitoring the vehicle's energy usage are commonly found in existing designs as power

system probes or in the form of more advanced Battery Management Systems[2] (BMS). Independently from their nature, existing sensors can be often re-purposed to be used with the proposed energy-driven diagnostic approach used in this architecture.

3.2 Knowledge Representation Module

As previously mentioned, a *knowledge base* module is also introduced in the proposed architecture. This is used for representing information gathered by low level modules and propagates refined concepts using different levels of abstraction. Such a component collects decisions taken by *fault mitigation* modules and estimations done at *fault detection* level to evaluate the impact of failures on the vehicle's effectiveness. This allows high-level autonomy elements, like *mission planner* or *task execution* modules, to better adapt the vehicle's remaining activities if a more suitable execution strategy can be computed. This behaviour is usually influenced by the severity of detected faults and, in extreme cases, could require the use of different navigation modes, in terms of trajectory planning or motion control, if normal capabilities are made unavailable.

An overview of this module is shown in Figure 3.3 where an example knowledge graph is presented for a hover-capable vehicle in the context of inspection missions. The proposed schema is a layered structure which connects and organises vehicle's internal beliefs at different abstraction levels. This isolates basic and measurable knowledge elements from complex features that can be derived after a simple reasoning procedure.

3.2.1 Components and Capabilities

Concepts at *component* level represent a physical device, such as an actuator or a sensor, together with its internal characteristics (e.g. health status or usage constraints). When combined together those are represented by concepts at *capabilities* level. These are introduced for propagating the effect of component's availability on specific vehicle's characteristics. At this level navigation capabilities are modelled as a set of logical relationships, for instance (3.1) and (3.2), among the control forces and lower level components.

$$Thr(x) \wedge Thr(y) \wedge Thr(z) \wedge \ldots \implies Dof(k) \tag{3.1}$$

Concepts at *action* level are used to represent elementary behaviours that can be executed by the vehicle. Those make use of underlying concepts but are not necessarily aware of all lower level relationships (3.3). Elements at this level can be seen as building blocks that compose more complex items or *tasks*.

$$Dof(x) \wedge Dof(y) \wedge Dof(z) \wedge \ldots \implies Cap(k) \tag{3.2}$$

[2]The BMSs are usually integrated in electric vehicles to monitor the health status of on-board battery packs. These manage the charge and discharge processes while evaluating the internal battery state (e.g. status of charge) at any point during vehicle's operations.

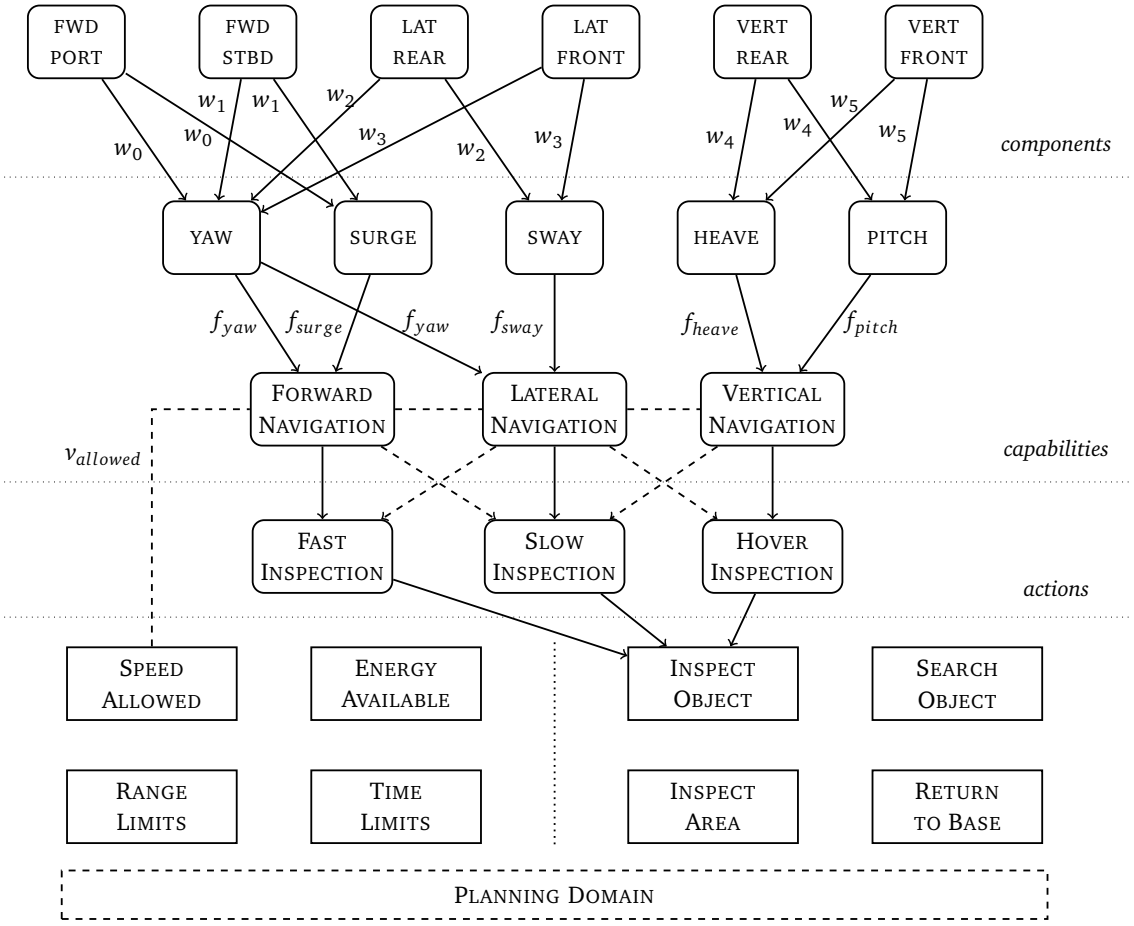


Figure 3.3: Knowledge representation of vehicle's status inside the *knowledge base* module. Relationships between high level concepts such as tasks and low level parameters such as health coefficients estimated by the *fault mitigation* modules are shown in terms of vehicle's *components*, *capabilities* and *actions* abstractions.

The sequence of *tasks* characterise the evolution of an autonomous mission. These concepts are also made available in the planning domain, where elementary behaviours, such following a trajectory or scanning a portion of seabed, are consider as atomic elements that can be either complete successfully or fail in achieving their intended scope. A list of principal logical relationships used in this framework is shown in Table 3.1.

$$(Cap(x) \wedge Cap(y)) \vee (Cap(z) \wedge \ldots) \implies Action(k) \tag{3.3}$$

The use of these abstractions allows the *knowledge base* module to correlate numerical values, such as health status estimates, to qualitative effects characterising more and more complex vehicle's capabilities. Following the example of Figure 3.3 a typical behaviour is detailed in the context of an inspection-type mission, for instance, like the ones conducted during the PANDORA project. In this case a *planning system* [137] has been given the goal of conducting a survey of mission's area for detecting the presence of sunken objects. Such a mission is composed by a sequence of simple tasks, for instance the *inspection object* one. Each task is executed selecting the most appropriate *action* among the ones available at

Level	Type	Relationship
components	*linear*	$Thr(1) \wedge Thr(2) \implies Surge$
components	*linear*	$Thr(3) \wedge Thr(4) \implies Sway$
components	*linear*	$Thr(5) \wedge Thr(6) \implies Heave$
components	*angular*	$Thr(5) \wedge Thr(6) \implies Pitch$
components	*angular*	$Thr(1) \wedge Thr(2) \wedge Thr(3) \wedge Thr(4) \implies Yaw$
capabilities	*navigation*	$Yaw \wedge Surge \implies Fnav$
capabilities	*navigation*	$Yaw \wedge Sway \implies Lnav$
capabilities	*attitude*	$Heave \wedge Pitch \implies Vnav$
actions	*inspection*	$Fnav \vee Lnav \implies FastInsp$
actions	*inspection*	$Lnav \vee Vnav \implies SlowInsp$
actions	*station-keeping*	$Vnav \vee Lnav \implies HoverInsp$

Table 3.1: List of principal logical relationships used in the *knowledge base* module.

the time of its execution. Actions are characterised by different *execution costs* derived by evaluating the state of vehicle's *components* and *capabilities*. In the event of failure, such a *thruster degradation*, fault knowledge is propagated from lower layers (e.g. *components*) to upper ones (e.g. *actions*). In this way execution costs and action's availabilities are updated to reflect the estimations done by *diagnostic* modules and influence the choices of planning modules. This is done by translating the logical relationships shown in Table 3.1 into PDDL (Planning Domain Definition Language) statements that are used to describe the planning problem for a given mission. Such a procedure is used at any stage of a PANDORA mission that require the solution of the planning problem in response to a mission event (e.g. completion of a task, detection of an object) or a vehicle event (e.g. failures, loss of capability).

3.3 Performance Estimation Framework

Beside internal assessment and knowledge related aspects another relevant component of the proposed energy-aware architecture is the introduction of a *runtime performance estimation* framework. This, as described previously, is given the task of monitoring at runtime the vehicle's performance while operating in unknown environments or when only limited information has been provided by operators. In this context performance is evaluated using a set of metrics, such as the *energy usage per unit distance* $\varepsilon_{nav}(\psi)$ and effective locomotion speed $v_{cruise}(\psi)$ that are derived from sensors' measurements taking into account the vehicle's heading ψ with respect to external disturbances, such as sea currents. Metrics are formulated taking into account previous studies [81], [88], [98] on performance evaluation and mission optimization in marine environments, and more recent works [13], [15] focusing mission's feasibility evaluation in the domain of ground robots. Runtime knowledge of vehicle's performance allows a better evaluation of the current mission's status and, when coupled with optimization procedures, to improve

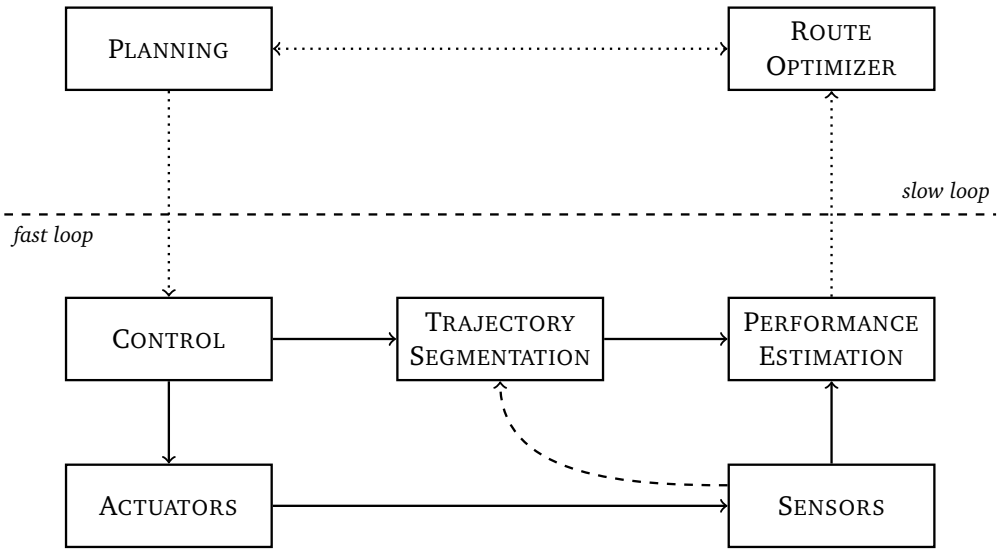


Figure 3.4: Overview of the runtime estimation system. Two feedback loops are implemented. A fast loop estimate vehicle's performance using sensors and navigation data. A slow one, instead, optimizes the vehicle's path while navigating in the mission's area.

several aspects such as locomotion efficiency, total execution time or final mission's outcome.

As mentioned in section 2.6 modelling of environmental aspects is a difficult task and many different approaches are found in literature. They focus mostly on the evaluation of disturbances at ocean scale aiming at long-range scientific experiments such as environmental surveys. In this context, however, the analysis is focused on operations conducted in small areas, such as littoral environments, internal waters or areas around human-made structures. Such operational scenarios are typical of small-size low-cost survey-class or intervention-class vehicles equipped with a small sensors package. These are often fitted with a Doppler Velocity Log (DVL) dedicated for navigation purposes but not always with additional Acoustic Doppler Current Profilers (ADCPs) that can provide sea current's measurements. Behind these limitations is the need of reducing the operational costs and, at the same time, extending their operating range by reducing the on-board payloads to strictly cover the mission's requirements (e.g. acquisition of imaging data). With these considerations in mind a data-driven approach is followed in this work to derive environmental knowledge from samples collected at runtime by on-board sensors. Data is analysed incorporating knowledge of different operating modes (e.g. navigation strategies, manoeuvring constraints) followed by the vehicle while executing a specific task.

An overview of the proposed runtime estimation framework is shown in Figure 3.4. A few modules are added to the vehicle's software architecture to collect motion data, identify trajectory segments and estimate the runtime metrics over a short time scale related to the current task execution. Concurrently derived metrics are used in a route optimization procedure that aims at improving the vehicle's efficiency if changes in the

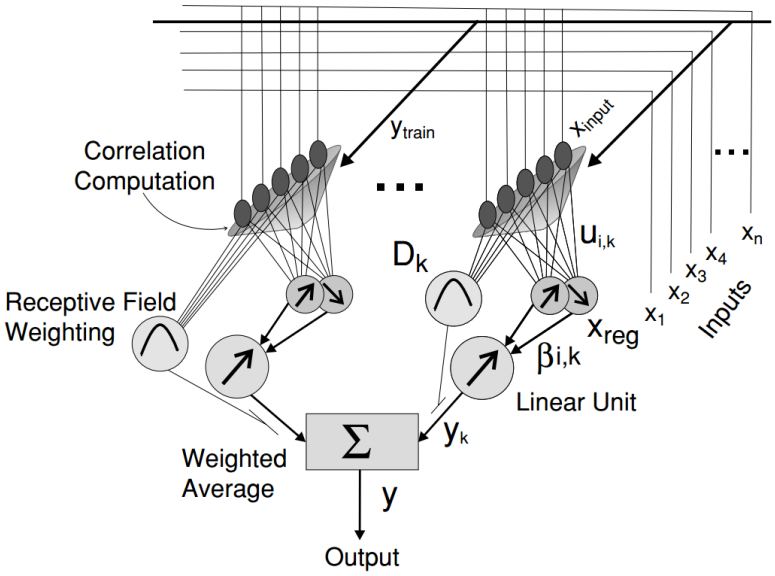


Figure 3.5: Schematic representation of LWPR algorithm. Adapted from [109].

operating conditions are detected. Those are, for instance, deviations from the initial assumptions [88] done at mission's planning stage about vehicle's navigation capabilities, execution time and energy consumption. Runtime estimations rely, even in this context, on the use of a regression analysis procedure implemented using the LWPR method and already employed for modelling internal aspects of vehicle's actuators. Such an approach allows, on the one hand, the characterisation of the behaviour of external disturbances, where changes in intensity and direction are not necessarily represented using simple linear models, and, on the other, the reuse of existing components already available in the vehicle's software architecture.

3.3.1 Regression Analysis

As mentioned in previous sections, the regression analyses presented in this architecture rely on the Locally Weighted Projection Regression (LWPR) algorithm [109]. This is a state-of-the-art non-linear regression technique that employs multiple linear models to approximate, on a smaller domain, high dimensional non-linear functions. This approach allows efficient capture and representation of complex models where often only their approximations are known in linear form. Such a method finds many applications in the robotics domain. In the underwater context it has been recently applied [110] to improved motion modelling, introducing a corrective term learnt during field operations to existing motion models (e.g. derived with classical identification procedures) or as non-linear model adaptation term [111] that aims at adjusting the vehicle's model in case of actuator failures.

This algorithm allows for incremental on-line learning through the introduction of a forgetting factor, which adjusts the learnt model as fresh input samples are provided. This method is best featured when a good number of starting samples ($N_t \geq 2000$) are

available, making its application ideal for the use of cases presented in this work, for instance, the extraction of thruster's characteristics and, as will be introduced in a later chapter, the identification of vehicle's performance metrics. In the context of thruster modelling the incremental learning capabilities are employed in updating the extracted models among vehicle's deployments. This aims at compensating for ageing effects of marine actuators by proving fresh input samples using the thruster's testing procedure previously introduced.

The LWPR algorithm, shown schematically in Figure 3.5, applies a dimensionality reduction to identify the most important directions in the input space. The domain in which each local model is activated is known as *receptive field* (RF) and it is defined using a Gaussian kernel. In each receptive field the function is locally approximated using a lower dimensional linear model as fitted with Partial Least Squares (PLS). LWPR produces the final result $\hat{y}$ for the target function as the weighted sum of all the predictions of local models. The weights w_k are taken from the kernel of each receptive field:

$$\hat{y} = \frac{\sum_{k=1}^{K} w_k \hat{y}_k}{\sum_{k=1}^{K} w_k} \tag{3.4}$$

$$w_k = \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{c}_k)^T \mathbf{D}_k (\mathbf{x}-\mathbf{c}_k)\right) \tag{3.5}$$

where $\mathbf{c}_k$ represents the centre of k-th receptive field, $\mathbf{D}_k$ the inverse covariance that specifies the receptive field's width, K is the total number of fields, $\hat{y}_k$ their local prediction and $\mathbf{x}$ the input query point. The *receptive fields* are adjusted on-line as more data is provided to the algorithm with no other intervention required.

Parameter	Description
$init\_D$	Initial width of a *receptive field* (rf)
$init\_alpha$	Step of the stochastic gradient descent
$update\_D$	On-line adaptation of width (on/off)
w_gen	Generation threshold for *receptive field*
$penalty$	Penalization for introducing a new *receptive field*

Table 3.2: Principal LWPR hyperparameters.

The identification of algorithm's parameters is an autonomous procedure known as *training*. A few *hyperparameters* control this behaviour. The most important are shown in Table 3.2. These are tuned to a problem's dynamics and their ranges are selected as a trade-off between under and over-fitting resulting models calculated on the initial experimental data. During this stage combinations of input parameters are use to generate different output models. These are evaluated using a k-fold cross-validation technique. After model training their predictions x_{pred} are calculated and a Mean Squared Error (MSE) metric is computed using samples x_{val} from cross-validation datasets. Analytically

this is expressed as:

$$MSE = \frac{1}{N_s}\sqrt{\sum_{i=1}^{N_s}(x_{pred}^{(i)} - x_{val}^{(i)})^2} \tag{3.6}$$

where N_s is the total number of samples used during *training*. The model that minimise the MSE metric is then chosen as output of the regression procedure.

3.4 Summary

This chapter is focused on the description of a novel *energy-aware architecture* that aims at improving the underwater vehicle's effectiveness when operating away from human supervision. First, a high level overview of this proposal has been presented. This describes the main architectural elements and shows how the proposed work is integrated within a more complex autonomy framework used for real underwater missions. Later, two of principal components, focusing respectively on *automatic fault mitigation* and *runtime performance estimation*, have been discussed. Those are both built around the analysis of energy usage while the underwater platform is operating in the field. Measurements are collected at runtime and are further employed to evaluate the status of the vehicle using a set of logical relationships among the derived metrics. Abstract concepts and relationships are contained in a *knowledge base* component. This refines lower level elements into higher level logical aspects providing other modules, such as *mission planners* and *task executors*, with up-to-date estimations of the vehicle's current capabilities.

Finally, a relevant non-linear regression technique has been introduced. This, used in both components, it allows the representation of the behaviour of vehicle's actuators and the modelling of the effect of external disturbances while relying on self-testing or automatic training procedures implemented on the vehicle itself. Internal details for the *automatic fault mitigation* framework are presented in Chapter 4. The *runtime performance estimation* framework is detailed in Chapter 5. Experimental validations, done in real environments for both components, are presented in Chapter 7.

Chapter 4

Fault Mitigation Framework

Failures are not something to be avoided. You want to have them happen as quickly as you can so you can make progress rapidly.

Gordon E. Moore

As outlined in previous sections the automated fault mitigation framework, introduced by the energy-aware architecture, relies on different components that combined together are used to model, detect and react to the presence of unexpected failures in the vehicle's propulsion subsystem. This is achieved with the support of a modified control schema that allows other modules to adjust at runtime the use of remaining healthy actuators. Such a schema, presented with more details in Figure 4.1, describes how estimations done at the diagnostic level are used to adjust the behaviour of the control subsystem in the event of actuators' failures.

4.1 Thruster Model

A first relevant module of this schema is the *Thrusters* element. This models the runtime behaviour of actuators when operating in standard conditions. Such a module is trained with samples collected using on-board sensors by means of a self-testing procedure. In this framework it used to represent the runtime power consumption of modelled actuators calculating their output currents given input commands taken from control modules. Analytically the internal behaviour of such a component is represented as:

$$I(t) = g_m(u(t)) \tag{4.1}$$

where $I(t)$ is the actuator's current draw, $u(t)$ is its throttle input command at time t and $g_m(.)$ is a non-linear function that represent the *throttle-to-current* characteristic for given actuator. The latter, described with more details in section 6.4, is identified using a

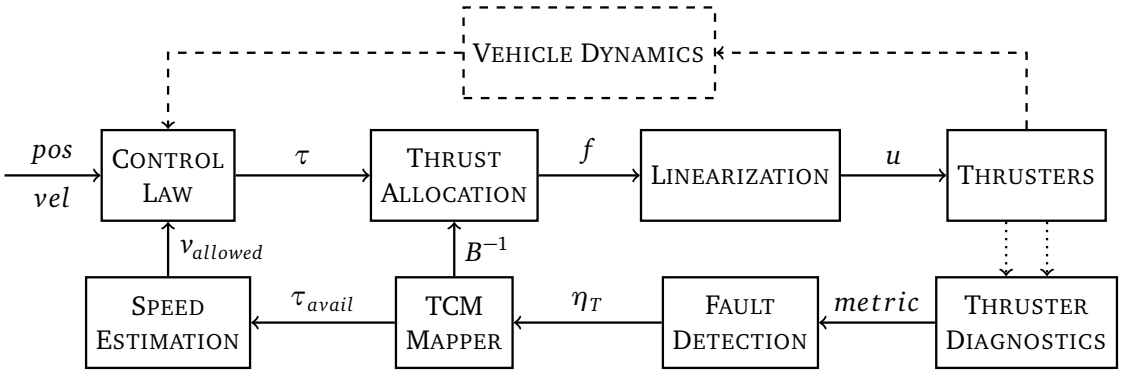


Figure 4.1: Schema of the fault-tolerant control architecture for Nessie AUV. The fault mitigation algorithm estimates at runtime the thruster efficiencies η_T and the allowed cruise speed for the actual configuration $v_{allowed}$. This influences the control law which takes into account any limits on the navigation speed when computing the control forces τ to move the vehicle. Moreover the fault-mitigation algorithm calculates an updated version of the *thruster configuration matrix* (*TCM*) used for allocating the control forces on the available actuators.

LWPR regression analysis executed on a training dataset. Knowledge of (4.1) allows a straightforward calculation of the actuator's instantaneous power:

$$P(t) = V(t) \cdot I(t) \approx V_{bus} \cdot I(t) \tag{4.2}$$

where $V(t)$ is the measured actuator's bus voltage, approximated as a constant V_{bus} for the experimental platform used in this work, and, by integration, to the evaluation of its energy consumption:

$$e(t) = \int_0^t P(t)\,dt = \int_0^t V(t) \cdot I(t)\,dt \tag{4.3}$$

The use of a non-linear regression procedure, such as LWPR, allows representing each device with a precise actuator model that takes into account minor differences in their underlying characteristics.

$$e(t) \approx V_{bus} \cdot \int_0^t I(t)\,dt \tag{4.4}$$

4.2 Thruster Diagnostics

A second relevant module is the *Diagnostics* component. This combines samples generated at the actuator's model level with measurements collected at runtime. It calculates a diagnostic metric that highlights deviations from standard behaviours. Such a metric is then used to detect the presence of failures in a following *fault detection* module.

As mentioned in previous sections while developing this work a *model-based* diagnostic approach is employed. This, described by the schema shown in Figure 4.2, consists of a real-time residual generator that produces a zero output metric if vehicle's actuators

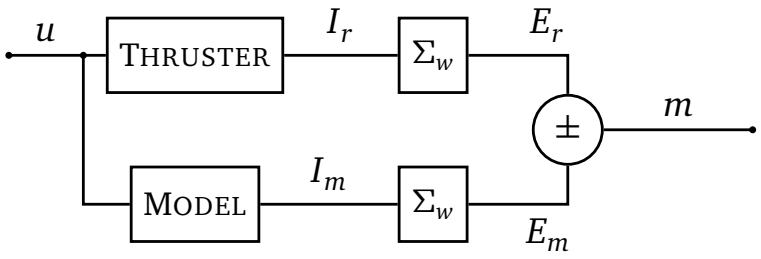


Figure 4.2: Thruster diagnostic schema. The u signal represents actuator's commands sent by control modules. A thruster model estimates the current usage I_m given its control input. Dedicated sensors measure the actual current I_r for the real device. Two integrators extract STE features E_m, E_r and a comparator calculates the diagnostic metric m used for detecting the presence of faults.

are operating like their analytical counterparts (e.g. following nominal characteristics) in presence of known control commands. The metric m is derived from a feature called *short-term energy* (STE). This is calculated by integrating the energy usage $e(t) \geq 0$ of a given actuator over a short time window W_e. This choice has been suggested by initial results of early experiments [50], [112] with an integrated energy measurement framework. The STE can be written in discrete time for both the real actuator and its model as:

$$E_j[n] = \sum_{k=0}^{W_e} V_{bus}\, T_s\, I_j[k] \qquad j \in \{r, m\} \tag{4.5}$$

where V_{bus} is the thruster nominal voltage, T_s the sampling time, I_r the measured current and I_m the current calculated using the thruster model derived in the previous section.

The diagnostic metric m is calculated as sum of residuals Δr of short-term energy features normalized and filtered using an exponential smoothing over a temporal window $W_f > W_e$.

$$\Delta r[n] = E_r[n] - E_m[n] \tag{4.6}$$

In this expression E_r represent the measured short-time energy and E_m the same feature derived from the thruster model's output. The metric m is thus given by:

$$m = \sum_{l=0}^{W_f} w_l \left(\frac{\Delta r[l]}{\Delta r_{max}} \right) \qquad m \in [-1, 1] \tag{4.7}$$

$$w_l = \frac{e^{-\alpha l}}{\sum_j^{W_f} e^{-\alpha j}} \qquad \sum_{l=0}^{W_f} w_l = 1 \tag{4.8}$$

where w_l are exponential weights given the parameter α, which controls the smoothing (e.g. $\alpha = 0.1$), and Δr_{max} is the maximum value of the short-term energy residual. This is derived taking into account the characteristics of underlying actuators assuming one

term of (4.6) as maximum and the other as zero. Analytically this is written as:

$$\Delta r_{max} = W_e \, V_{bus} \, T_s \, I_{max} \tag{4.9}$$

where I_{max} represent the maximum designed current usage of the device and W_e is the length of the integration window. Quantities W_e and W_f are chosen with the trade-off between accuracy and latency. They are considered as parameters for tuning the

Parameter	Value
W_e	2.0 s
W_f	20.0 s
T_s	0.1 s
V_{bus}	28.0 V
I_{max}	7.5 A
Δr_{max}	42.0 J

Table 4.1: List of parameters for the thruster diagnostic module.

responsiveness of the proposed diagnostics module. In evaluating this work those values have been chosen using operational experience from indoor trials and are reported in Table 4.1.

4.3 Thruster Failures

In the marine environment thruster failures can be related to several causes [113], for instance, the presence of objects blocking the propeller, tangled ropes or water leaks that disrupt the functionality of the actuator itself. In these cases previous field experience [18], [73], [74] as well as modelling studies [54] suggest that the *performance degradation* failure, as well as the complete loss of the actuator, is also a relevant problem that should be addressed during real sea operations. This failure mode, as the name implies, is related to a loss of efficiency for the faulty component, which operating at reduced capacity produces a different effect on the vehicle's navigation to what is expected by its control architecture. This fault, often difficult to detect in presence of feedback control loops, can be modelled as an actuator that presents a *thrust efficiency* η_T lower than one.

$$\eta_T = \frac{T_{real}}{T_{nominal}} \qquad 0 \leq \eta_T \leq 1 \tag{4.10}$$

Efficiency is defined as the ratio between effective thrust T_{real} (e.g. the one produced during real operations) and the design thrust $T_{nominal}$ (e.g. the one achievable in ideal conditions). The parameter η_T can thus be used to represent the health status of the actuator. In this framework the severity of a degradation is controlled by adjusting η_T: from the complete loss of efficiency ($\eta_T = 0$) to standard operating conditions ($\eta_T = 1$).

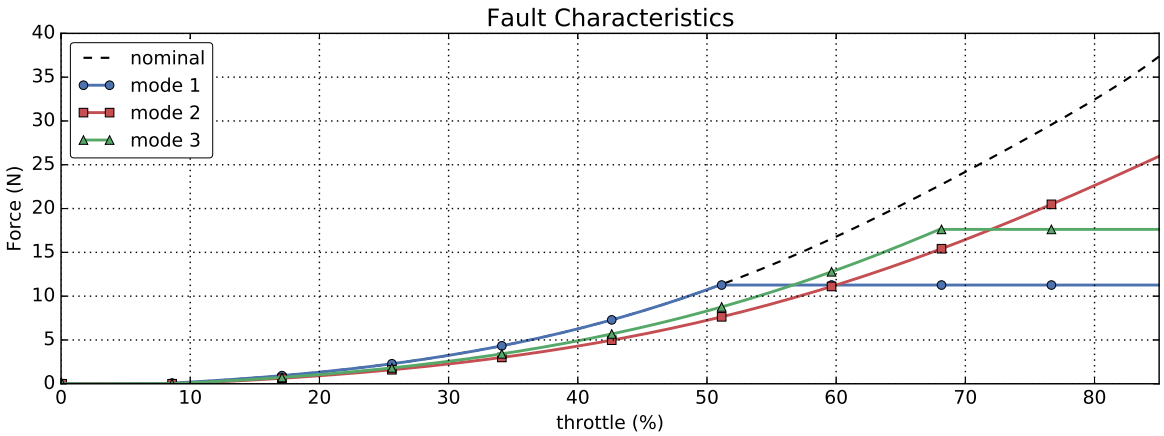


Figure 4.3: Example of thruster's fault characteristics. In this first case the thruster is operating with a limit on its maximum power, capping its effort above 50% of the input command. In the second case a reduced thrust efficiency ($\eta_T = 0.8$) is introduced. In the last case a combined effect of the previous modes is shown with $\eta_T = 0.9$ and capping above 68%.

This work focuses more on the effect of degradation failures $\eta_T > 0$ rather than on the complete loss of actuators. In such a case the vehicle is still capable of controlling a *degraded thruster* without necessarily excluding it from the thrust allocation problem. This approach is followed until higher level autonomy modules identify which long-term adaptation needs to be implemented. A total failure $\eta_T = 0$, while still relevant, can generally lead to the loss of a DOF if no redundancy is available in the vehicle's design (e.g. few actuators are available). This is the case for the *heave* DOF of the experimental platform considered in the rest of this work. Because of these aspects experiments involving vertical thrusters are not taken into account. In case of actuator's loss an alternative solution, which does not rely solely on the remapping of control forces, can be researched, for instance, by adapting the vehicle's motion or its navigation trajectory [84], [85] to compensate for the lack of control along a specific axis. Implementation of similar strategies is outside the scope of this work as those rely more on reinforcement leaning and experience driven techniques.

Another relevant aspect is that the resulting characteristic of a degradation fault is often unknown and only a qualitative description can be estimated. Figure 4.3 shows some of the possible models that can be used to represent degradation scenarios. In all of those cases the actuator's behaviour is disrupted and the nominal thruster characteristic is not followed along the full range of input commands. From a low-level point of view the degradation fault can be seen, for instance, as the effect of a damaged ball-bearing, propeller or nozzle which affect the flow of water around an actuator and thus its output performance. Analytically this characteristic can be represented as:

$$T_i(u) = \eta_{T_i}\, G_{m_i}(u) \qquad\qquad i \in \{0,\ 1,\ \dots\ ,\ N\} \tag{4.11}$$

where G_{m_i} is the *throttle-to-thrust* relationship for the i-th thruster, η_{T_i} its efficiency and N the number of thrusters. The $G_m(u) \propto g_m(u)$ term is derived from the *throttle-to-current*

characteristic (4.1) using the relationship between electrical power and output thrust[1].

This work considers explicitly the first degradation failure mode as shown also in Figure 4.3. Its effects are reproduced by limiting the maximum power output of a target actuator. In the experimental platform this is achieved by manipulating on-the-fly the input commands to reduce its output thrust. This procedure is known as *fault injection* and it is described in details in section 6.6.1. Once such a modification is in place the actuator's characteristic deviates from its standard condition triggering, then, detections in the diagnostic system. This behaviour has been observed also in presence of a real failure (e.g. the loss of a propeller blade) as reported in section 7.1.1. Despite this focus, other failure modes could be handled by the proposed framework. This feature is given by the use of a diagnostic metric that highlights the presence of a unexpected behaviour as long as a change in the energy usage is detected.

$$\eta_{T_i}[n] = \begin{cases} \eta_{T_i}[n-1] - \delta_{\eta_T} & \text{if fault detected} \\ \eta_{T_i}[n-1] & \text{if nominal} \end{cases} \tag{4.12}$$

The severity of detected failures is represented using a qualitative approach. This allows other modules, such as fault mitigation ones, to react to generic fault conditions and yet implement a possible correction over a short time horizon. For this reason a counting procedure has been also introduced in the proposed framework. This translates the series of diagnostic decisions for each actuator into an estimation of their thrust efficiency. This is assumed to be optimal at the beginning of a mission and, in presence of a sequence of positive detections for t_f consecutive time instants, it is gradually decreased until the actuator's functionality is restored (e.g. the metric falls below the λ_d threshold) or its use is declared not feasible by an higher level reasoning module.

$$\eta_{T_i}[n] = \begin{cases} \eta_{T_i}[n-1] - \delta_{\eta_T} & \text{if fault detected} \\ \eta_{T_i}[n-1] & \text{if nominal} \\ \eta_{T_i}[n-1] + \gamma_{\eta_T} & \text{if fault cleared} \end{cases} \tag{4.13}$$

A recovery capability is also modelled in such a procedure. This relies on another coefficient $\gamma_{\eta_T} \ll \delta_{\eta_T}$ that allows the system to restore the use of a given actuator if no further mismatch is found after a specific time $t_r \gg t_f$ during normal navigation and its actual efficiency is above a certain threshold $\eta_T \geq 0.2$. This choice allows the system to cope with transient faults, such as the ingestion of sea weed in tunnel thrusters or larger debris in open actuators. The settling time t_r is specific to a particular type of thruster and it is usually identified experimentally at design time. Nonetheless, in case of severe failures (e.g. $\eta_{T_i}[n] < 0.2$) the faulty actuator is first excluded from the system (e.g. to prevent further damage) and a notification is escalated from low level diagnostic modules

[1]For commercial thrusters, like the ones used in this work, the relationship between electrical power and output thrust is provided by the manufacturer. Alternatively, it can be calculated experimentally with any of the methods discussed in Section 2.3.

to the higher level knowledge representation components.

4.4 Fault Detection

A third module is the *Fault Detection* component. This analyses the energy-based metric provided by the *diagnostics* module and detects the presence of faults. The output is a sequence of decisions d that signal the detection of an underlying problem with a specific actuator. Moreover, this module keeps track, as introduced in the previous section, of efficiency estimations for the thruster subsystem, allowing other modules (e.g. *mitigation*, *knowledge base*) to reason and adapt in presence of specific fault conditions.

One common method to detect the presence of a fault given the computed metric is the use of a single threshold. Such an approach relies on the detection theory and it relates the threshold's value, used to generate binary decisions, with given a detection performance expressed, for instance, in terms of probability of false alarm P_{fa} (i.e. assuming the presence of faulty condition in a fault free case).

$$d = \begin{cases} 0 & \text{if} \quad |m| < \lambda_d \\ 1 & \text{if} \quad |m| \geq \lambda_d \end{cases} \tag{4.14}$$

Different strategies can be used for deriving the λ_d threshold. One approach is to assume the metric as a random variable and to express the generic λ_d threshold as:

$$\lambda_d = \mu_m + k\sigma_m = k\sigma_m \qquad\qquad k = 1,\ 2,\ 3,\ \ldots \tag{4.15}$$

where k is a scaling parameter, μ_m and σ_m the metric's mean and standard deviation. In a fault free case the mean is assumed to be zero while the term σ_m takes into account the residual uncertainty from real-time measurements noise σ_{sens} and the precision of the extracted LWPR model σ_{model}.

$$\sigma_m^2 \propto \sigma_{sens}^2 + \sigma_{model}^2 \tag{4.16}$$

If the metric m is assumed to be following a normal distribution $N(\mu_m, \sigma_m^2)$ the probability of false alarm is given by:

$$P_{fa} = \int_{\lambda_d}^{\infty} \frac{1}{\sqrt{2\pi\sigma_m^2}} e^{-\frac{x}{2\sigma_m^2}} dx \tag{4.17}$$

This suggests a straightforward procedure to calculate the value of threshold λ_d as function of the chosen P_{fa} and the uncertainty of the diagnostic metric σ_m. While developing this work the performance of detection system has been tuned to allow a probability of false alarm of 1% or $P_{fa} = 0.01$. This choice while conservative, allows the system to reject outliers (e.g. given for instance by synchronization errors in the ROS framework) without

reducing the system's responsiveness when in presence of the degradation faults as shown in the experimental section.

Under these assumptions the residual term Δr can be seen as difference between two normal distributions, respectively, $N(\mu_{sens}, \sigma^2_{sens})$ and $N(\mu_{model}, \sigma^2_{model})$ generated from the measurement and model process of a given actuator. In an ideal case, neglecting the model bias and assuming a fault free scenario, their means can be assumed to equal $\mu_{sens} = \mu_{model}$. This lets the Δr term be represented with a normal distribution of zero mean and known variance: $N(0, \sigma^2_{sens} + \sigma^2_{model})$. Taking into account these aspects the (4.7) can be rewritten as:

$$m = \sum_{l=0}^{W_f} w_l X_l \tag{4.18}$$

where $X_l = \Delta r[l]/\Delta r_{max}$. Under the assumption of independence between residual samples the variance of the metric can be expressed as:

$$\begin{aligned} \sigma^2_m &= \mathrm{Var}\left(\sum_{l=0}^{W_f} w_l X_l\right) = \sum_{l=0}^{W_f} \mathrm{Var}(w_l X_l) = \sum_{l=0}^{W_f} w_l^2 \mathrm{Var}(X_l) \\ &= \sigma^2_x \sum_{l=0}^{W_f} w_l^2 = \frac{\sigma^2_{sens} + \sigma^2_{model}}{\Delta r_{max}} K_w \end{aligned} \tag{4.19}$$

where K_w represents the scaling term given by the w_l coefficients. The terms σ_{sens} and σ_{model} are numerically estimated from (4.5) using the training dataset employed to extract the actuator's model:

$$\sigma^2_{sens} \propto W_e \; V_{bus} \; T_s \; \sigma^2_i \tag{4.20}$$

$$\sigma^2_{model} \propto W_e \; V_{bus} \; T_s \; \sigma^2_{lwpr} \tag{4.21}$$

where σ^2_i represents the precision of current measurements (e.g. $\pm 0.05\,A$ for the validation platform) and σ^2_{lwpr} is given by the confidence intervals of the specific LWPR model for the actuator under consideration. Such an approach is motivated by the availability of a single fault hypothesis (e.g. healthy/faulty actuator) which allows a probabilistic analysis of the metric and by the operational experience collected on the experimental platform during field trials. In fact, as presented in the experimental sections, this strategy, despite its simplicity, allows satisfactory levels of performance when implementing the diagnostic system on a real underwater vehicle with limited[2] computational capabilities.

Beside the use of a single threshold other fault detection techniques have been successfully applied in the context of fault detection such as the use of Artificial Neural Networks (ANN) [61] or the use of Fuzzy Decision Making (FDM) methods [114]. The FDM ap-

[2]A relevant aspect of AUVs built around the ROS framework is their characteristic of sharing computational resources among different elements of the vehicle's control architecture. Often real-time modules are given only a small portion of CPU time and for this reason complex algorithms can be only evaluated using batch processing over larger time frames.

proach represents a relevant technique as it allows the detection and isolation of multiple faults using the same type of residuals presented in this section. Fuzzy methods are useful in the presence of noisy or imprecise measurements and when interpretation is highly dependent on human experience. In the case of FDM the use of simplified membership functions, aggregation operators supports a detection system using a common threshold for multiple fault hypothesis. The use of FDM is considered for the future work related to the proposed energy-aware architecture.

4.5 Fault Mitigation

A fourth module is the *Fault Mitigation* component. This is composed of smaller elements, such as *Thrust Allocation*, *TCM Mapper* and *Speed Estimation* also shown in Figure 4.1, that analyse the output of previous modules (e.g. estimated efficiencies and binary decisions) and introduce at runtime a short-term reaction to unexpected failures. At this level, a set of estimated quantities (e.g. allowed speed, available forces) is shared with the higher level *knowledge base* module and, concurrently, these are used to adjust the internal behaviour of the control subsystem. In more severe cases (e.g. when degradation disrupts the actuator's behaviour) a different allocation procedure is selected to maintain satisfactory navigation capabilities.

The proposed system implements mitigation strategies by adjusting the use of available actuators proportionally to their estimated efficiencies η_{T_i}. A well-known approach to achieve this is to modify the thrust allocation policy, for example, manipulating the *Thruster Configuration Matrix* (TCM) generally used in the control schema of underwater vehicles. The TCM matrix, also indicated as B, is defined as:

$$B_{(6\times N)} = \begin{bmatrix} e_{x_1} & \dots & e_{x_N} \\ e_{y_1} & \dots & e_{y_N} \\ e_{z_1} & \dots & e_{z_N} \\ (r_1 \times e_1)_x & \dots & (r_N \times e_N)_x \\ (r_1 \times e_1)_y & \dots & (r_N \times e_N)_y \\ (r_1 \times e_1)_z & \dots & (r_N \times e_N)_z \end{bmatrix} \tag{4.22}$$

where $e_i = [e_{x_i}\ e_{y_i}\ e_{z_i}]$ and $r_i = [r_{x_i}\ r_{y_i}\ r_{z_i}]$ are, respectively, the orientation and positions of each thruster with respect to vehicle's centre of mass, $c = [x_c\ y_c\ z_c]$. This matrix is used to distribute generalised force requests $\tau_{(1\times 6)} = [x\ y\ z\ k\ m\ n]$, expressed with respect to the centre of mass, in forces $f_{(1\times N)} = [f_0\ \dots\ f_N]$ specific to individual actuators. Analytically the relationship between thruster configuration matrix B and actuator's forces f is expressed as:

$$f = B^{-1}\tau \tag{4.23}$$

On the other hand, the thrust request τ can be derived, at least from a general point of

view, by taking into account the equation of motions [69] that describe the behaviour of a generic underwater vehicle. These are written as:

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau \tag{4.24}$$

$$\dot{\eta} = J(\eta)\nu \tag{4.25}$$

where M is the inertia matrix of the underwater vehicle, including rigid-body and added mass effects, C is the matrix of centrifugal and Coriolis terms, D is the matrix of damping terms, g is the vector of restoring forces (gravity and buoyancy) and $\nu_{(1\times6)} = [u\ v\ w\ p\ q\ r]$ is the vector of linear and angular velocities in vehicle's body-fixed frame. Using this approach the generalized force vector τ represents the forces acting on the underwater vehicle with respect to the vehicle's body-fixed frame.

In this case the thrust efficiency coefficients η_T can be introduced in the solution of the allocation problem (4.23). These modify the matrix B^{-1} weighting the contribution of individual thrusters proportionally to their estimated capabilities, reducing thus the efforts on damaged actuators and redistributing it on the healthy thrusters [78]. Such an approach has the ability to maintain standard navigation capabilities at least for less severe failure scenarios.

4.5.1 Thruster Remapping

This approach, also known as *weighted generalized inverse*, makes use of a weighting matrix W where the coefficients ($w_i = \eta_{T_i}$ and $0 \leq w_i \leq 1$) along its diagonal represent the estimated thrust efficiency for the platform's thrusters.

$$W_{(N\times N)} = \begin{bmatrix} w_0 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_N \end{bmatrix} \tag{4.26}$$

In standard conditions each coefficient is assumed equal to 1, meaning that the specific thruster is healthy and fully available for generating the requested thrust. In the case of a complete failure (i.e. $w_i = 0$) the use of this technique guarantees that a faulty actuator is excluded from the allocation problem's solution. With the introduction of matrix W the (4.23) can thus be rewritten as:

$$f = B_w^{-1}\tau \tag{4.27}$$

where the B_w^{-1} is known as weighted generalized pseudo-inverse of the thrust configuration matrix:

$$B_w^{-1} = W^{-1}B^T\left(BW^{-1}B^T\right)^{-1} \tag{4.28}$$

The use of a generalized inverse [69], [77] avoids the presence of singularities (e.g. when more thrusters are excluded from the allocation problem) and is general enough to deal with non-square thruster configuration matrices (e.g. when thrusters are greater

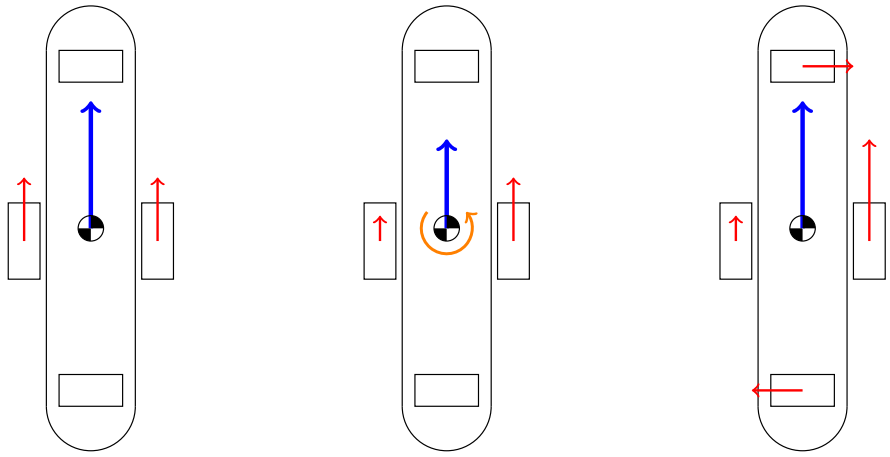


Figure 4.4: Thrust allocation in Nessie AUV. The diagram on left shows the reference case where a given amount of force has been requested along the surge axis. The middle case shows the effect of the degradation of the port-side actuator without any compensation. An extra torque is produced by the unbalance between thrusters and the total force request is not met. The right shows the effect of the thruster allocation algorithm where other actuators are used to compensate for the faulty thruster.

than the controlled degrees of freedom). It can be efficiently computed with the use of singular value decomposition (SVD) [115]. An example of thruster remapping is shown in Figure 4.4 where a standard allocation is compared to a faulty behaviour for a generic AUV with four actuators on the surge-sway plane. In this example the port-side forward thruster is affected by degradation. Under these conditions an additional torque is generated by the unbalanced operation of forward thrusters. By remapping the use of actuators the remaining thrusters are used to compensate this unwanted behaviour. If allowed the starboard-side thruster may increases its effort to match the original force request.

4.5.2 Thruster Saturation

Despite its flexibility the introduced approach can not guarantee that the allocated force vector f is within the saturation limit of each thruster in every fault scenario. In fact, the presence of a degradation failure introduces extra effort on the remaining actuators to compensate for the lack of thrust in the failed component. Such an aspect may push the requested forces above the capability limits of other actuators which under standard conditions are guaranteed to stay within the capability limits.

A solution for this problem has been proposed [76] in literature but it requires modifications of the controller architecture by introducing extra integrators at the input channels of the underlying controller system. On the other hand, while developing this work it has been decided to leave the low level controller unmodified and to follow a different approach. This does not increase the complexity at lower levels, allowing the reuse of existing implementations, but it optimises the force allocation policy, preventing the effect of saturations, by introducing a dynamic programming procedure that researches a valid

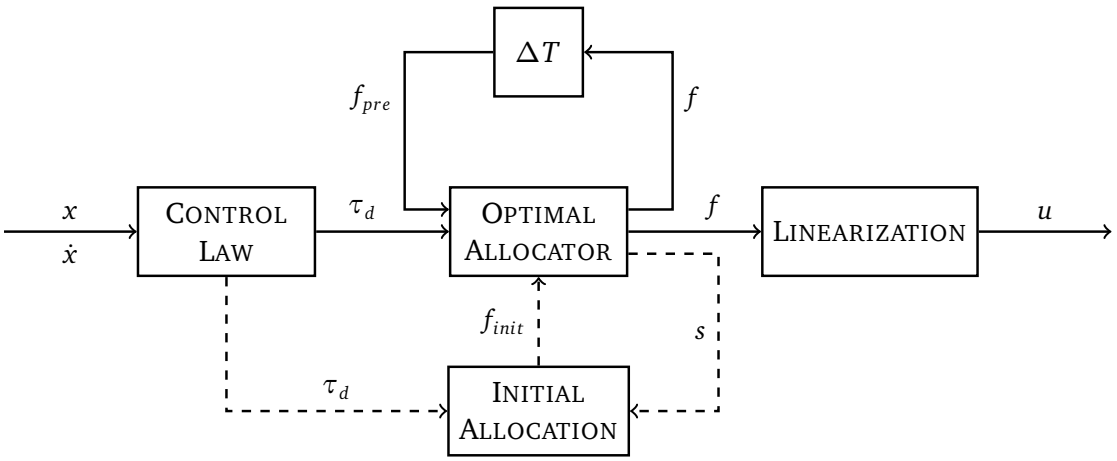


Figure 4.5: Optimal thrust allocation schema. The control law calculates a desired thrust vector τ_d. The solver, initialised with a solution calculated using the classical approach f_{init}, calculates the optimal allocation f and slack variable s. Subsequent iterations include the previous solutions $f_{pre} = f(t - \Delta T)$ to take into account the actuator's dynamics.

solution given the actuator's limits. A similar method has been explored in previous works [116] and applied in the context of marine vessels. This relies on the use of a quadratic programming (QP) formulation that derives a solution for the allocation problem. Other techniques [77] have been also used in the past, such as the S-approximation or T-approximation. Those are focused on mitigating the effect of force unbalance after allocation and for this reason those are not discussed in this work.

The approach followed in this work relies on a customized variant of the original QP formulation [116], implemented with the use of CVXPY [117], a Python-embedded modelling language for convex optimization problems, and the state-of-the-art ECOS [118] solver as Mixed Integer Quadratic Programming (MIQP) problem. These provide an efficient way of implementing a real-time optimization method that can be integrated in the software architecture of existing AUVs. An overview of this allocation schema is shown in Figure 4.5. The optimization problem, instead, is defined as:

$$\min_{f,s} \; f^T P f + s^T Q s + \beta \bar{f} + \xi \tag{4.29}$$

subject to:

$$s = \tau_d - B_w f \tag{4.30}$$

$$-f_{act} \leq f \leq f_{act} \tag{4.31}$$

$$-\Delta f \leq f - f_{pre} \leq \Delta f \tag{4.32}$$

where the first term $f^T P f$ takes into account the power consumption of actuators, with the use of a matrix P, the second $s^T Q s$ penalizes the error s between thrust request τ_d and allocated generalized force $B_w f$. The slack variable s is required to always allow a feasible solution to be found in the optimization problem.

The matrix Q is chosen as $Q \gg P > 0$ so that solutions are found with $s \approx 0$. The matrix P is built approximating the thruster characteristics as quadratic functions $h(f_i) = af_i^2 + bf_i + c$. This, differing from the original formulation, allows the analyses conducted on the real actuators to be taken into account. Quadratic approximations are used to simplify the optimization problem and to improve the time needed to research a valid solution. Furthermore, the term f_{act} is the available thrust for a given actuator and $\bar{f}$ is the maximum component of the f vector. These are defined as:

$$f_{act_i} = \eta_{T_i} f_{max_i} \qquad i \in \{0,\ 1,\ \ldots,\ N\} \tag{4.33}$$

$$\bar{f} = \max_i |f_i| \qquad \bar{f} > 0 \tag{4.34}$$

where f_{max_i} is the maximum thrust for the i-th thruster. These quantities are used in the above formulation to bound the actuator forces using the estimated thrust efficiency (4.31) and to penalize solutions where a single thruster is working at its maximum capacity (4.29) by introducing a coefficient $\beta > 0$. At the same time, actuator dynamics are enforced using (4.32), where f_{pre} is the solution found during the previous step and Δf represents the maximum rate of change for the actuators' output. The use of such constraint allows solutions that abruptly modify the throttle commands to be discarded. Lastly, ξ is a relaxation term that groups coefficients needed to avoid singularities and numerical errors.

Stable navigation in the presence of different failures is enforced by introducing a customised *force prioritization policy*. This allows the control schema to allocate forces in an ordered way, satisfying first thrust requests for high priority degrees of freedom (DOFs). An example of a possible policy ω for an inspection task is $heave > pitch > yaw > sway > surge > roll$. This prioritises adjustments of depth and orientation to maintain the alignment of on-board sensors while relaxing the control on forward and lateral displacements. This is implemented by adjusting the coefficients q_i of the diagonal matrix Q such as:

$$q_i > q_j \quad \text{if } \omega(i) > \omega(j) \qquad i \neq j \ \wedge\ i, j \in \{0, 1, \ldots, N\} \tag{4.35}$$

still under the condition $Q \gg P > 0$. Such an approach makes the solver selecting solutions where the error term s is lower for higher priority DOFs.

4.5.3 Vehicle Speed Adjustments

Beside saturation prevention the proposed mitigation strategy also introduces an adjustment to vehicle's *cruise speed* in case a moderate failure is detected. This aims at keeping the controller's requests closer to actuator's capabilities and to limit the overall compensation of unbalanced forces. Such a procedure, beside providing short-term adjustments, allows the fault mitigation framework to share its knowledge with other

autonomy modules about the recommended navigation speed (or allowed ranges) along each vehicle's principal axis. This information is then used at mission planning or task execution level where knowledge about vehicle's capabilities is used to estimate execution times or resource requirements for the current mission.

The adjusted navigation speed is calculated by estimating the available force on each DOF. This is derived from the remaining force of each thruster f_{act_i}, the estimated efficiencies η_{T_i} and the thruster configuration matrix B, which provides the relationship between actuators and degree of freedoms. In other words the available generalized force for a specific DOF can be written as:

$$\tau_{avail}(dof) = \sum_i f_{act_i} = \sum_i \eta_{T_i} f_{max_i} \quad \text{with} \quad i \in D_T(dof) \tag{4.36}$$

where $D_T(.)$ is the set of thrusters controlling a specific degree of freedom dof. After calculating (4.36) for all the controlled degrees of freedom the calculated navigation speed can be expressed using the heuristic:

$$v_{allowed} = v_{cruise} \sqrt{\frac{\tau_{avail}}{\tau_{nominal}}} \qquad 0 \leq v_{allowed} \leq v_{cruise} \tag{4.37}$$

where $v_{allowed} = [u\ v\ w\ p\ q\ r]$ is the adjusted cruise speed vector, v_{cruise} is the initial speed set at deployment time under the constraint $v_{cruise} \leq v_{max}$ (where v_{max} is the maximum vehicle's speed) and $\tau_{nominal}$ is the force available in standard conditions without any fault.

The use of such an expression is explained by physical effects of individual terms in the motion's equation (6.15). In particular for underwater vehicles, the damping term D is often approximated with the use only of *quadratic drag* coefficients [69]. Such an approximation assumes the drag force proportional to the square root of speed. In this case drag force represents one of the stronger contributions to forces acting against the vehicle's motion. Therefore by introducing a smaller navigation speed the requested force, calculated by the controller for cruise navigation phases, is also decreased accordingly thus limiting the effects of thrust unbalance and saturation limits.

4.6 Summary

In this chapter the *automatic fault mitigation* framework has been introduced. This is one relevant component of the proposed *energy-aware architecture* that deals with *survivability* aspects focused around the propulsion subsystem. Initially the actuator's model used in this work is discussed. This relies on a compact analytical representation for the *thrust-to-throttle* characteristic, extracted using a hardware-in-the-loop (HIL) self-testing procedure and LWPR regressions. After discussing the underlying component's models the use of an energy-based diagnostic procedure has been discussed. This is built around

the concept of analysing the short-term energy (STE) usage of real actuators with respect to their analytical counterparts. Such an approach takes the form of a model-based diagnostic schema where a fault detection procedure is implemented on output residuals. Together with diagnostic details, an analytical representation for marine thrusters' failure is briefly discussed. This is used in the proposal to model the effect of faulty actuators when in presence of degradation failures. These do not disrupt completely the actuator's functionality but, on the other hand, they affect the navigation capabilities of the vehicle in a subtle manner.

Beside those aspects another relevant element of the proposed framework is the presence of a mitigation feedback. This is implemented by introducing a modified control schema for the underwater vehicle that deals with thrust allocation when in presence of faulty actuators. Such an approach is based on the concept of *thrust efficiency*, also discussed in this chapter, that it is estimated during field operations using the output metrics of the introduced fault detection subsystem. Runtime knowledge of actuator's state allows the proposed system to proportionally adjust their use in order to maintain, within achievable limits, the control capabilities and to allow the current task to be completed. As mentioned in this chapter, classical approaches that rely only on weighting coefficients may not guarantee saturation-free operations when in presence of failures. For those reasons the proposed solution implements an optimization procedure that deals with the allocation problem while taking into account actuator's constraints. These are adjusted at runtime using the information collected during the diagnostic analysis and complemented with operational ones tailored to the vehicle's navigation modes. After solving the allocation problem new speed limitations are calculated using the estimated health status of on-board actuators. These are used to keep the vehicle's behaviour close to its effective capabilities, for instance, reducing the unbalancing effect of unhealthy actuators.

Along with this short-term mitigation other evaluations are conducted at higher levels to introduce a more suitable mitigation strategy on a longer term. Those are conducted with the support of a *knowledge base*, introduced in the previous chapter, where metrics calculated at diagnostic and mitigation levels, such as estimated efficiencies and speed limits, are transformed into high-level concepts that can be used to adjust the remaining part of a complex mission. Experimental validation for the proposed *automatic fault mitigation* framework is discussed in Chapter 7. Real sea and controlled environment experiments analyse the capability of such a proposal implemented on a real hover-capable AUV.

Chapter 5

Runtime Performance Estimation

If we have an atom that is in an excited state and so is going to emit a photon, we cannot say when it will emit the photon. It has a certain amplitude to emit the photon at any time, and we can predict only a probability for emission; we cannot predict the future exactly.

Richard P. Feynman

After discussing the energy-based mitigation aspects another relevant component of the proposed energy-aware architecture is the *runtime performance estimation* framework. This focuses on the runtime analysis of vehicle's operations when navigating in environments where external disturbances are known to affect its capabilities. As mentioned in previous chapters such a framework employs measurements with a non-linear regression procedure, built around the LWPR algorithm, and it evaluates the feasibility of missions using performance metrics calculated on-the-fly. The concept of feasibility, introduced in section 2.6.2, can be modelled using the Probability of Mission Completion (PoMC). Such an approach requires a probabilistic analysis of the sequence of tasks that compose missions conducted in unknown environments.

This work focuses on AUVs conducting inspection-type missions in areas where sea currents can be assumed slowly varying for large portions of the environment, for instance, as in the PANDORA project's scenarios. This assumption allows the modelling of sea current's behaviour as a stochastic process where only limited variations in terms of speed (e.g. $\Delta v_c \leq 0.1$ m/s) and direction (e.g. $\Delta \psi_c \leq 10°$) are taken into account. Furthermore, it is assumed that the underwater vehicle is given the goal of surveying a limited area where multiple inspection points (IPs) are provided together with a set of constraints on execution time and resource usage. No *a priori* information is available other than what has been recorded by the vehicle during previous operations in the same operational environment. Finally, the availability of a contingency plan (e.g. return to the recovery

point) defined, for instance, before deployment is taken into account as alternative in case of mission failure.

The random variables taken into account for this analysis are the execution time and energy usage of mission's task when in presence of external disturbances. These variables are often objectives of interest for other works on mission planning, for instance, where great importance is given to the temporal domain [82], the energy consumption [88] or their application [81] in the evaluation of large-scale survey scenarios. In this proposal, on the other hand, more importance is given to the concept of mission's feasibility and probability of completion. Those are used as a criteria for evaluating the opportunity of interrupting, adapting or restructuring the current mission and to estimate the feasibility of possible alternative plans that are calculated on-the-fly by the unmanned platform. Those efforts aim at improving the vehicle's *reliability* allowing a certain degree of adjustments to be made in order to guarantee a satisfactory development of unsupervised operations.

A proposed methodology for evaluating variations with respect to original plans is to employ an *energy-constrained route optimization problem* that re-calculates at runtime alternative sequences of inspection procedures. Such an approach has the goals of adapting the vehicle's navigation, by incorporating knowledge collected from the environment, and, concurrently, evaluating the possibility of discarding less favourable tasks if the vehicle is operating in presence of resource scarcity [88]. The proposed route optimization algorithm is derived from the Orienteering Problem (OP) [119], an optimization procedure that allows specific constraints, budgets and rewards to be considered in the set of solutions.

The OP can be seen as combination of the Travelling Salesman Problem (TSP) [120] and the Knapsack Problem (KP). In this work the original problem's constraints are customized for AUV scenarios, allowing vehicles to navigate along non-cyclic paths (tours). This approach is similar to the Open Vehicle Routing Problem (OVRP) [121], where each vehicle is not required to return to its starting point after visiting its intermediate targets. A detailed description of the OP is found in [119]. On the other hand, a new variation of the original problem has been presented recently as Correlated Orienteering Problem (COP) [122]. This takes also into account the capability of gathering information about neighbouring targets when conducting inspections. Such an approach is relevant for robotics and unmanned vehicle applications and it also taken into account in this work.

5.1 Mission Model

An inspection mission is defined by a series of consecutive tasks (or actions) that must be achieved without failures or exhaustion of resources in order to reach a final goal.

$$M = \{A_0, A_1, \ldots A_n\} \tag{5.1}$$

Actions are of different kinds and are affected by uncertainty of the environment. In this type of mission these are mainly *navigation* and *inspection* tasks. The first regulates

trajectory following, path generation and general navigation among different inspection points. The second, instead, allows the vehicle to turn on its sensors while in proximity of an inspection point and to acquire relevant data using the vehicle's sensor payload. In the case of hover-capable AUVs an alternative navigation mode may be employed for this type of task. This, for instance, can hold the vehicle's position fixed while close to an inspection area and control its orientation to identify possible objects of interest in the surrounding environment. Such an approach is often needed to steer and focus the vehicle's on-board sensors (e.g. stereo cameras, narrow view imaging sonars, etc.) in order to conduct accurate data acquisition procedures.

The mission is also characterized by constraints that are often defined at the planning stage or are derived by platform specifications. These are, for instance, restrictions on resource usage during navigation (e.g. maximum battery discharge rate, optimal speed, depth of discharge, etc.) or the introduction of a contingency energy reserve. The latter is often required for safety purposes, for example, when the vehicle is operating in environments where recovery operations are not available. Constraints are generally introduced on the mission duration and total energy usage, two quantities unknown before execution and objectives for the probabilistic analysis. These are affected by uncertainty of the environment and by the non-deterministic development of tasks. For those reasons the quantities above are well represented by stochastic processes associated with the overall mission's uncertainty.

Under these considerations the probability of mission completion (PoMC) can be represented by writing the probability of exceeding constraints on resource usage:

$$R = 1 - P(t_m > t_T, e_m > e_T) \tag{5.2}$$

The right-hand term, also known as *probability of failure*, it is expressed in integral form for the mission process M as:

$$P(t_m > t_T, e_m > e_T) = \int_{t_T}^{\infty} \int_{e_T}^{\infty} f_M(t_m, e_m)\, dt_m de_m \tag{5.3}$$

where t_m and e_m are the mission time and energy random variables, t_T and e_T are mission constraints on duration and energy usage and $f_M(t_m, e_m)$ is the probability density function relating *duration* and *energy usage* of the mission process M. This formulation introduces a simplification over the real mission process, where other sources of uncertainty (marine traffic, obstacles) and abrupt changes in the environment are ignored. Nonetheless, it provides a quantitative representation when dealing with stochastic environments.

5.1.1 Task Models

With such a mission model in mind the two stochastic quantities, taken into account when assessing the mission's feasibility, can be modelled analytically for a generic mission

process M as:

$$t_m = \sum_{i=0}^{N_A} t_{A_i} \tag{5.4}$$

$$e_m = \sum_{i=0}^{N_A} e_{A_i} \tag{5.5}$$

where t_m, e_m are, respectively, the total mission time and total energy usage, N_A is the number of tasks and t_{A_i}, e_{A_i} the expected *duration* and *energy usage* for the i-th task. These quantities, from a general point of view, are unknown before completing the mission and their value can only be evaluated using the knowledge available during execution. In this formulation each task is modelled according to its class (or type) considering the individual realisations independent from others ones.

Repeatable Task

For a generic task, with the exception of *navigation* type, the expected *duration* and *energy usage* are assumed to follow a normal distribution. This is a reasonable assumption for all repeatable tasks that are executed during a survey-like mission. These are tasks where the vehicle is executing a fixed action, for instance, adjusting its position with a constant displacement, that do not depend on the actual mission's state. These are still affected by external disturbances but their duration and/or energy usage is determined mainly by deterministic factors (e.g. fixed time durations, sensor constraints) rather than the environment itself. In those cases the expected *duration* and *energy usage* quantities are written as:

$$t_{C_i} \sim N(\mu_{t_C}, \sigma^2_{t_C}) \tag{5.6}$$

$$e_{C_i} \sim N(\mu_{e_C}, \sigma^2_{e_C}) \tag{5.7}$$

where the parameters $(\mu_{t_C}, \sigma^2_{t_C})$, $(\mu_{e_C}, \sigma^2_{e_C})$ describe, respectively, the expected values and variances for *duration* and *energy usage* of a generic task for the C class. In this case the knowledge of these parameters, for instance derived from previous experience, allows an efficient representation of the uncertainty associated with the execution of tasks.

Locomotion Task

Locomotion tasks, on the other hand, are more sensitive to environmental disturbances that affect both their energy consumption and execution time during navigation. Such tasks are thus better described by models that take into account the effective locomotion speed and navigation performance, estimated at runtime using measurements collected on-board, combined with an additional term that represents higher order effects. This approach allows the non-deterministic behaviour of the underlying control software to be

taken into account when in presence of external disturbances. These, in fact, may affect localization algorithms or trajectory following procedures by increasing the position errors or by perturbing the vehicle's navigation. Analytically locomotion tasks are written as:

$$t_{leg_i} \sim f(d_{nav},\ v_{cruise}) + N(\mu_T, \sigma_T^2) \tag{5.8}$$

$$e_{leg_i} \sim f(\varepsilon_{nav},\ v_{cruise}) + N(\mu_E, \sigma_E^2) \tag{5.9}$$

where the travel time t_{leg_i} is derived from d_{nav}, which represents the travel distance among IPs, and v_{cruise} that estimates the effective locomotion speed. This travel time is updated during the mission's execution if any corrective action (collision avoidance, trajectory adjustments) is taken during navigation by recalculating the d_{nav} term. The energy usage e_{leg_i}, instead, is derived by taking into account the effective locomotion performance ε_{nav}, also known as *energy usage per unit distance*, when navigating in presence of disturbances. Even for this type of task, those additional quantities are only partially known when starting a mission.

General Formulation

Given the above assumptions about mission's tasks the original equations (5.4) and (5.5) can, thus, be rewritten as:

$$t_m(t) = t_m^{(t)} + \sum_{i=0}^{L} t_{leg_i} + \sum_{i=0}^{K} t_{ins_i} \tag{5.10}$$

$$e_m(t) = e_m^{(t)} + \sum_{i=0}^{L} e_{leg_i} + \sum_{i=0}^{K} e_{ins_i} \tag{5.11}$$

where $t_m^{(t)}$ and $e_m^{(t)}$ are the measured mission duration and energy usage at time t, L and K are the remaining *navigation* and *inspection* tasks, t_{leg_i} and t_{ins_i} the expected *duration*, e_{leg_i} and e_{ins_i} the expected *energy usage* for the i-th task. The (5.10) and (5.11) describe the estimated *duration* and estimated *energy usage* for a generic time t during execution. These quantities are used to evaluate the mission feasibility (5.3) with respect to the user's constraints. Such a procedure is executed multiple times over the course of a single mission, for instance, after completing a pair of navigation and inspection tasks.

More in detail, by taking into account the introduced models for locomotion and inspection tasks, the equations (5.10) and (5.11) can be further expressed by calculating the expected values $E[X]$ and variances $E[(X - E[X])^2]$ for the estimated duration and energy usage variables. These are written, respectively, as:

$$\hat{\mu}_{t_m} = t_m^{(t)} + \hat{\mu}_{t_{nav}} + K \cdot \hat{\mu}_{t_{ins}} \tag{5.12}$$

$$\hat{\sigma}^2_{t_m} = \hat{\sigma}^2_{t_{nav}} + K^2 \cdot \hat{\sigma}^2_{t_{ins}} \tag{5.13}$$

and:

$$\hat{\mu}_{e_m} = e_m^{(t)} + \hat{\mu}_{e_{nav}} + K \cdot \hat{\mu}_{e_{ins}} \tag{5.14}$$

$$\hat{\sigma}^2_{e_m} = \hat{\sigma}^2_{e_{nav}} + K^2 \cdot \hat{\sigma}^2_{e_{ins}} \tag{5.15}$$

where $(\mu_{e_{ins}}, \sigma^2_{e_{ins}})$ describe the *inspection*-class tasks and the pairs $(\hat{\mu}_{t_{nav}}, \hat{\sigma}^2_{t_{nav}})$, $(\hat{\mu}_{e_{nav}}, \hat{\sigma}^2_{e_{nav}})$ describe the *navigation*-class task. The first two parameters are derived from experience of repeated tasks and usually known once the task has been defined. The last four, instead, are estimated at runtime and their formulation is derived in the following sections.

5.1.2 Task Tracking

After introducing the modelling aspects another relevant aspect is the use of a *task tracking* procedure. This allows the framework to update intermediate estimations for repeatable tasks using the experience collected at runtime while the vehicle is conducting its mission. Such an approach improves the initial beliefs used to characterise the mission's feasibility at early stages. In fact, once a sequence of tasks is defined (5.10) and (5.11) can be evaluated for the current mission. Expected t_m and e_m are thus recorded together with estimations $\hat{t}_{A_i}$ and $\hat{e}_{A_i}$ for all the remaining actions. This evaluation of mission time and energy consumption is repeated after each task is achieved. Upon completion of the task A_n the proposed framework stores its measured duration t_{A_n} and energy usage e_{A_n}. The actual travel distance d_{leg_n} is also recorded if the task is of *navigation* type. These values are compared with estimations done at a previous stage. If significant variations from the former *plan* are detected a mission assessment process is started. In a positive case (5.10) and (5.11) are evaluated taking into account the full mission process M including the completed tasks where a complete knowledge about their performance is now available. Besides incorporating the latest available runtime estimates this recursive procedure also allows the residual uncertainty estimations to be periodically updated. This is done by using the stored knowledge about recently completed tasks. Analytically those are written, respectively, as:

$$\hat{\mu}_T \approx \frac{1}{N_A - L} \sum_{i=0}^{N_A - L} \hat{t}_{A_i} - t_{A_{plan_i}} \tag{5.16}$$

$$\hat{\mu}_E \approx \frac{1}{N_A - L} \sum_{i=0}^{N_A - L} \hat{e}_{A_i} - e_{A_{plan_i}} \tag{5.17}$$

and:

$$\hat{\sigma}^2_T \approx \frac{1}{N_A - L} \sum_{i=0}^{N_A - L} \left(\Delta t_{A_i} - \hat{\mu}_T\right)^2 \tag{5.18}$$

$$\hat{\sigma}^2_E \approx \frac{1}{N_A - L} \sum_{i=0}^{N_A - L} \left(\Delta e_{A_i} - \hat{\mu}_E\right)^2 \tag{5.19}$$

where $\Delta t_{A_i} = \hat{t}_{A_i} - t_{A_{plan_i}}$, $\Delta e_{A_i} = \hat{e}_{A_i} - e_{A_{plan_i}}$, N_A is the total number of tasks, L the tasks not yet achieved and $t_{A_{plan_i}}$, $e_{A_{plan_i}}$ are the estimated performance metrics for the i-th task.

This tracking procedure is used to combine the observed behaviour (or execution experience) with the sequence of actions calculated for a given mission plan. It provides an up-to-date estimation of the total *duration* and *energy usage* while developing the mission without relying on outdated assumptions made at planning time. The use of (5.16) and (5.17) allows the removal of any systematic errors (e.g. offsets) from previous calculations. After evaluating a new pair of t_m and e_m values the probability of mission completion (5.3) is also updated. If a plan exceeds the required constraints or if the (5.3) falls below a given threshold the current sequence of tasks is declared infeasible and an optimization process is started to derive a more suitable execution plan.

5.2 Runtime Estimates

As mentioned in the previous sections this framework relies on two runtime estimations to improve the assessment of a generic mission process. The first is the *energy cost per unit distance* associated with vehicle's navigation. This, introduced initially in [81], is estimated as function of the vehicle's absolute heading $\varepsilon_{nav}(\psi)$ rather than globally for a single survey. The second is the average cruise speed $v_{cruise}(\psi)$ still as function of absolute heading. The $v_{cruise}(\psi)$ term is measured with the aid of a DVL sensor as speed over ground (SOG) during navigation. These two functions are periodically re-evaluated at runtime in order to incorporate small changes of external environment. The effect of disturbances, like sea currents, is to change the two functions proportionally to their intensity and the prevalent directions.

In order to compute $\varepsilon_{nav}(\psi)$ and $v_{cruise}(\psi)$ features such as the average speed $\bar{v}$, the absolute navigation heading $\bar{\psi}$, the cumulative travel distance Δd and energy consumption Δe for the current trajectory are collected at runtime over a time window of N_w samples. Together with these features, speed σ_v^2 and heading σ_ψ^2 variances are also computed over the same time window. Features with similar properties are grouped together in sets of M elements. The conditions (5.23), (5.24), (5.25) are used for the grouping procedure. Once a group is collected a single measure $\hat{\psi}$ is derived for the navigation heading. This is then used to identify the value of the two functions given the group's features. Analytically this is expressed as:

$$\hat{\psi} = \frac{1}{M}\sum_{i=0}^{M}\bar{\psi}_i \tag{5.20}$$

$$\varepsilon_{nav}(\hat{\psi}) = \frac{1}{M}\sum_{i=0}^{M} p_i \frac{\Delta e_i}{\Delta d_i} \tag{5.21}$$

$$v_{cruise}(\hat{\psi}) = \frac{1}{M}\sum_{i=0}^{M} p_i v_i \tag{5.22}$$

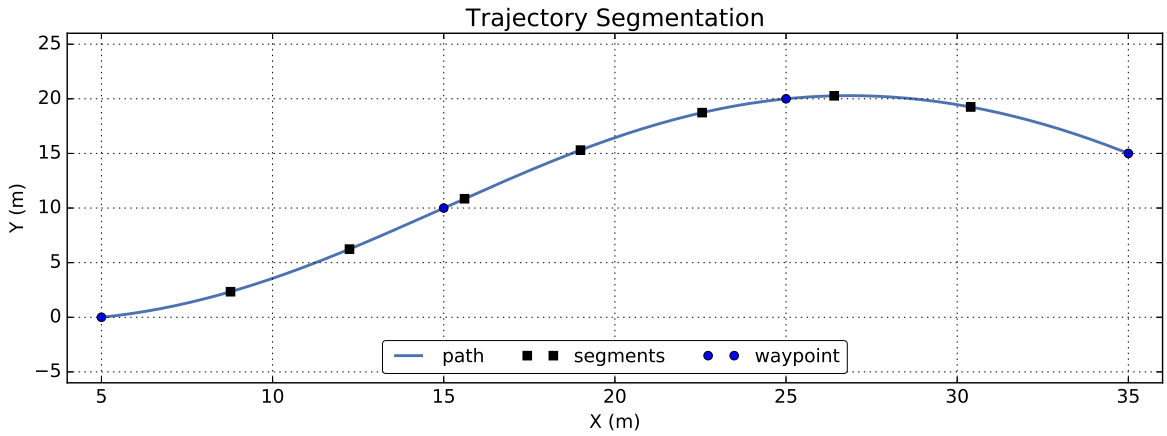


Figure 5.1: Segmentation result for a partial smooth trajectory used for inspection. Segments are isolated by limiting variations on their prevalent heading ψ and speed v. These cover approximately the same distance if the navigation is continuous and with small heading changes.

where M is the number of elements in the current group, $\bar{\psi}_i$ are the recorded headings, p_i are scaling weights, Δe_i and Δd_i are the energy usage and the travel distance for the i-th sample that satisfies the σ_v and σ_ψ conditions for the current group:

$$\sigma_\psi \leq \quad \Delta\psi \leq \quad 10 \text{ deg} \tag{5.23}$$

$$\sigma_v \leq \quad \Delta v \leq \quad 0.1 \text{ m/s} \tag{5.24}$$

$$|v - v_{cruise}| \leq \quad v_{thr} \leq \quad 0.1 \tag{5.25}$$

The condition (5.23) allows the proposed procedure to operate even in the presence of curved trajectories. This happens without loss of detail by splitting a large trajectory in multiple small segments. An example of this behaviour is shown in Figure 5.1 where smaller segments are employed as soon as the vehicle adjust its navigation heading. Such an approach allows the use of a finer resolution while in presence of smooth manoeuvres. The conditions (5.24) and (5.25) ensure that samples are collected during *cruise* navigation while filtering out any data related to *acceleration* or *deceleration* phases.

$$\sum_{i=0}^{M} p_i = 1 \quad \text{with} \quad p_i = \frac{e^{-\lambda i}}{\sum_j^M e^{-\lambda j}} \tag{5.26}$$

The coefficients p_i are chosen to weight the contribution of consecutive samples and decay with an exponential forgetting factor λ as defined in (5.26). The value M is bound between M_{low} and M_{high} limits. Features are collected in a single group as long as conditions (5.23), (5.24), (5.25) are met for all elements within the current group. If a group exceeds M_{high} samples a new group is created. If less than M_{low} samples are collected the group is discarded. Given these conditions each group identifies a single trajectory segment with homogeneous properties and the upper-bound M_{high} sets a limit for the length of segments to used in this procedure.

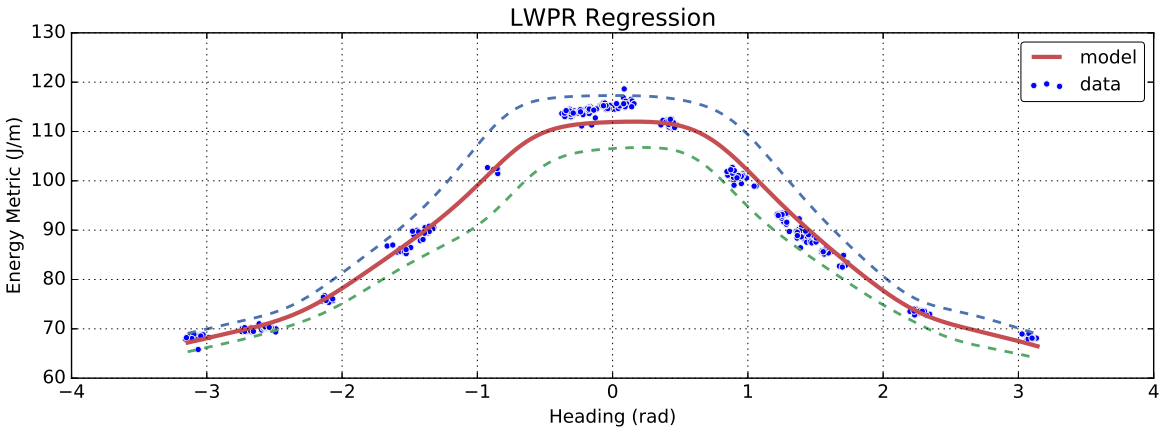


Figure 5.2: Example of LWPR regression with real data. Disturbances and measurement noise affects samples collected runtime. Nonetheless the LWPR algorithm extracts a smooth representation from noisy data. Dashed lines show the 95% prediction intervals associated with the computed model.

5.2.1 Data Pruning

Samples collected with the introduced procedure are stored in a dataset that is used to derive models for the $\varepsilon_{nav}(\psi)$ and $v_{cruise}(\psi)$ functions. As more data is made available during the course of an autonomous mission such a dataset is further expanded. On the other hand, to promptly capture variations of operating environments and to exploit temporal dependencies, older samples need to be progressively filtered out as they may not be representative from the current environment's state. Samples filtering is done using a binning procedure. These are efficiently grouped together in N_h bins representative for a range of possible headings. For each bin the latest K_h samples are kept. This retains the latest available estimations without discarding too many samples for less frequent navigation directions. Parameters N_h and K_h control the complexity of the resulting *data pruning* procedure and define the amount of output data available for further processing. The number of bins N_h is calculated taking into account the precision of heading sensors, with special interest for their accuracy and drift stability over long trajectory legs. In fact, assuming bins 12 to 4 degrees wide, operational values for the N_h parameter are found in the range from 30 to 90. This allows the procedure to work with bins large enough to include navigation errors when initiating or completing small heading adjustments (e.g. during hovering operations). The number K_h, instead, has been empirically chosen. This is because a minimum number of samples $N_s = K_h \cdot N_h \geq 1000$ is required for applying correctly a regression analysis on the derived dataset.

After obtaining a pruned dataset $\varepsilon_{nav}(\psi)$ and $v_{cruise}(\psi)$ functions are learnt using the Locally Weighted Projection Regression (LWPR) [109]. This method has been previously introduced in Section 3.3.1 and applied successfully also in the context of actuator's modelling. In the case of performance metrics few modifications are taken into account. A k-fold cross-validation (CV) technique is yet again employed during this phase, however, the number of folds is reduced to account for the presence of a lower input samples count

for the early stages of the estimation procedures. After *training* models are evaluated using a Mean Squared Error (MSE) metric, this is expressed as:

$$MSE = \frac{1}{N_s}\sqrt{\sum_{i=1}^{N_s}(x_{pred}^{(i)} - x_{val}^{(i)})^2} \tag{5.27}$$

where $N_s \sim K_h \cdot N_h$ is the total number of samples used during training, x_{pred} is the model prediction and x_{val} the corresponding samples from the CV dataset.

A relevant aspect of the LWPR algorithm is its capability of calculating *confidence intervals* given a generic query point x within the learnt function domain. These identify a range in which future observations will fall with a given probability according to what has been observed already. Confidence intervals are generically defined as:

$$\Pr_{\Theta}(h_l(X) < Y < h_u(X)) = \gamma_h \qquad \Theta = \{\theta_0,\ \theta_1,\ \ldots\} \tag{5.28}$$

where X is a random sample from a probability distribution defined using parameters in Θ, Y is a variable related to X, $(h_l(X), h_u(X))$ the lower and upper bounds, respectively, of observable values Y and γ_h is a value close but not equal to 1 that define the intervals. Knowledge of these allow a better characterisation of the residual uncertainty of $\varepsilon_{nav}(\psi)$ and $v_{cruise}(\psi)$ values calculated using the LWPR model. In fact, beside obtaining their expected values, also a range of other probable values is made available at runtime for further improving any subsequent analysis. Such an approach takes into account the uncertainty given by measurement noise and by the interaction of the vehicle's control subsystem with external disturbances. An example model for $\varepsilon_{nav}(\psi)$ is shown in Figure 5.2

Parameter	Value
init_D	0.01 – 1.0
init_alpha	1.0 – 10.0
update_D	*on*
w_gen	0.01
penalty	0.0001

Table 5.1: LWPR hyperparameters used for performance estimations.

together with its training samples and confidence intervals. This is extracted with the introduced procedure using the hyperparameters reported in Table 5.1.

5.2.2 Model of Trajectory Dynamics

Vehicle locomotion, together with previous estimations, is also represented using a simplified model. This describes the forward navigation using time-delayed third-order exponential functions to approximate the trapezoidal velocity profile, typical of a point-to-point navigation. This model is often used in trajectory generation problems [123] and

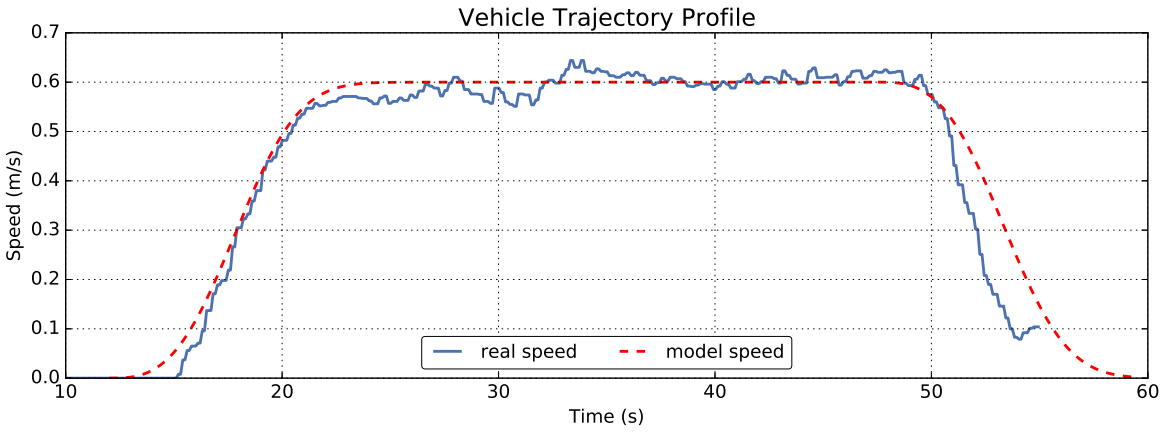


Figure 5.3: Nessie AUV's velocity profile. The exponential model (dashed line, $\alpha_{nav} = 0.15$) is overlaid on navigation data (solid line) taken from an inspection mission in real sea conditions. This represents the vehicle's behaviour during a *navigation* task between two inspection points while in presence of external disturbances.

guarantees the computed path has continuous derivatives up to the third order. Figure 5.3 shows how this approximates well the experimental vehicle's behaviour when operating in real environments, for instance, in presence of tidal currents.

Such a model assumes the normalized velocity profile to be described by the equation:

$$v(x) = \frac{v(t)}{v_{max}} = 1 - e^{-x^3} \quad \text{with} \quad x = \alpha_{nav} t \tag{5.29}$$

where v_{max} represents the vehicle cruise speed for point-to-point navigation, α_{nav} is a time-scaling parameters that controls the slope of the exponential functions and depends on the acceleration limits of the platform. This profile is used during planning stage as it provides time estimates for vehicle navigation. In fact, assuming a mission in which distant waypoints are visited one after the other, the time needed for an individual trajectory leg can be approximated as:

$$t_{leg} = t_{acc} + t_{cruise} + t_{dec} \tag{5.30}$$

where t_{cruise} is cruising time, t_{acc} the time needed to reach the cruise speed and t_{dec} the one to stop at the end of the trajectory leg. For reasonable sized legs (e.g. $d_{leg} \geq 10$ m) t_{acc} and t_{dec} represent a small percentage of the cumulative navigation time and can be assumed constant for all the intermediate legs. Given the platform's acceleration limits, the mission's navigation speed v_{cruise} and the configuration of underlying motion controllers t_{acc} and t_{dec} can be derived for a specific vehicle's configuration. A possible approach is measuring the time needed to increase the AUV's speed from 5% to 95% of the requested v_{cruise}. In the case of experimental platform considered in this work, the t_{acc} and t_{dec} are both estimated to be approximately 15 seconds each.

Under these assumptions only the t_{cruise} term needs to be periodically re-evaluated to take into account the effects of operating environments. In fact, assuming a uniform

motion for the *cruise* phase, this can be written as:

$$t_{cruise} = \frac{d_{leg} - d_{acc} - d_{dec}}{v_{cruise}} \tag{5.31}$$

where d_{leg} is the total length of the trajectory leg, d_{acc} and d_{dec} the navigation distance covered during *acceleration* and *deceleration* phases. These last values are calculated numerically by integration of the velocity profile (5.29), respectively, in the $[0, t_{acc}]$ and $[t_{brk}, t_{leg}]$ domains, where $t_{brk} = t_{leg} - t_{dec}$.

$$d_{acc} = \int_0^{t_{acc}} v(t)\, dt \tag{5.32}$$

$$d_{dec} = \int_{t_{brk}}^{t_{leg}} v(t)\, dt \tag{5.33}$$

This allows representing the leg navigation time t_{leg} as function of the navigation distance d_{leg} and the average cruise speed v_{cruise} according to the estimations calculated at runtime:

$$t_{leg} = \frac{d_{leg} - d_{acc} - d_{dec}}{v_{cruise}(\psi)} + t_{acc} + t_{dec} \tag{5.34}$$

where ψ is the line-of-sight angle between two consecutive waypoints that identify the given trajectory leg. A similar approach is followed to derive the required energy. This is done under the assumption that navigation among waypoints is conducted in a point-to-point fashion, typical of an inspection mission. In this case the energy requirement e_{leg} for a single trajectory leg is given by:

$$e_{leg} = d_{leg} \cdot \varepsilon_{nav}(\psi) + \xi_e \tag{5.35}$$

where $\varepsilon_{nav}(\psi)$ represent the *energy cost for unit distance* while navigating with heading ψ. The ξ term represents residual costs not taken into account by this formulation, like the initial acceleration effort or other second-order effects.

$$t_{nav} = \sum_{i=0}^{L} t_{leg_i} \tag{5.36}$$

$$e_{nav} = \sum_{i=0}^{L} e_{leg_i} \tag{5.37}$$

Using the introduced formulation cumulative locomotion (5.36) time and energy usage (5.37) for the navigation tasks are, thus, expressed as the summation of known terms.

On the other hand, under the assumption of Gaussian uncertainty the equations (5.34) and (5.35) allow expressing with more detail the variables (5.8) and (5.9). Analytically

these are described using their expected values and variances:

$$\hat{\mu}_{t_{nav}} = \sum_{i=0}^{L} \hat{\mu}_{t_{leg_i}} + \sum_{i=0}^{L} \hat{\mu}_T \tag{5.38}$$

$$\hat{\mu}_{e_{nav}} = \sum_{i=0}^{L} \hat{\mu}_{e_{leg_i}} + \sum_{i=0}^{L} \hat{\mu}_E \tag{5.39}$$

$$\hat{\sigma}^2_{t_{nav}} = \sum_{i=0}^{L} \hat{\sigma}^2_{t_{leg_i}} + \sum_{i=0}^{L} \hat{\sigma}^2_T \tag{5.40}$$

$$\hat{\sigma}^2_{e_{nav}} = \sum_{i=0}^{L} d^2_{leg_i} \cdot \hat{\sigma}_{\varepsilon_{nav}}(\psi_i)^2 + \sum_{i=0}^{L} \hat{\sigma}^2_E \tag{5.41}$$

where $\hat{\sigma}_{\varepsilon_{nav}}(\psi_i)^2$ is calculated from the *confidence interval* of the underlying regression model for the data point ψ_i and $\hat{\sigma}^2_{t_{leg_i}}$ is the numerically estimated variance for the navigation time of the i-th trajectory leg. The pairs $(\hat{\mu}_T, \hat{\sigma}^2_T)$, $(\hat{\mu}_E, \hat{\sigma}^2_E)$ are, instead, the estimated values for the residual uncertainty term of the two random variables (5.8) and (5.9).

5.3 Route Optimization

Along with estimation aspects the proposed framework implements a route optimization procedure based on the Orienteering Problem (OP). This technique, well-known in literature [119], allows an agent to optimize its travel path when visiting several points of interest while respecting constraints of budgets, duration and, at the same time, while maximising the final outcome. Many approaches have been proposed by researchers when applying the orienteering problem to autonomous sensing under the presence of constraints. A sub-modular OP formulation, solved with an approximation algorithm, is used to maximise information gathering in [124]. On the other hand, a similar formulation has been paired with a recursive greedy algorithm in [125] for maximising the additional information gain in a sensor network. For real-time use in large graphs the use of a linear approximation has been proposed in [126] for area coverage with a micro aerial vehicle while a sampling method has been presented in [127] to maximise the exploration of uncertain areas using genetic algorithms. A multi-robot sensing system has been introduced for agriculture applications in [128]. This solves the OP using a 4-approximation algorithm that guarantees at least a quarter of the points to be included in the problem's solution.

In this work the Orienteering Problem (OP) is tailored to the underwater mission's context employing the runtime estimations with a derived algorithm known as Energy-aware Orienteering Problem (EA-OP). This has the aim of maximizing the execution performance in presence of stochastic environments and to improve the utilization of on-board resources. Concurrently it allows the vehicle to calculate on-the-fly alternative

execution plans if assumptions made before deployment are invalidated by the performance metrics collected at runtime. For an inspection mission this algorithm provides an optimal sequence of tasks that aims at visiting a maximum number of inspection points while taking into account the task's models introduced in the previous sections. The resulting travel sequence characterizes an instance of the mission process M. Such an algorithm is first used at the beginning of a new mission when only an initial belief about the vehicle's performance, described in terms of (5.38) and (5.39), is available. This technique provides a *provisional plan* that does not yet include knowledge from the operating environment. Later, during execution, the plan is further refined as additional runtime measurements are collected.

5.3.1 Energy-Aware Orienteering Problem

As mentioned previously this framework employs a variant of the OP with the introduction of additional constraints and runtime estimations. This uses an open version for the routing problem that allows solutions to be non-cyclical paths (or tours) where user-defined starting and ending points are enforced. Such a characteristic is typical of inspection missions at sea where starting and recovery areas are distinct and delimited by operational constraints (e.g. presence of planned marine traffic or the use of a moving support vessel). This non-cyclical behaviour is more common of OVRP [121] formulations, where valid solutions are optimal routes that explore the full mission space. Later in this work, the proposed variant is compared to an existing OVRP formulation, used in the past with the same experimental platform to conduct field missions.

The EA-OP algorithm is described with a Mixed Integer Linear Programming (MILP) formulation where the navigation cost and time for exploring the mission space are derived, respectively, from (5.35) and (5.34). The aim of this is to maximise the objective function:

$$\sum_{i \in V} \sum_{j \in V} r_i x_{ij} \tag{5.42}$$

subject to the following constraints:

$$s.t. \sum_{i \in V_s} x_{is} = \sum_{i \in V_f} x_{fi} = 0 \tag{5.43}$$

$$\sum_{i \in V_s} x_{si} = \sum_{i \in V_f} x_{if} = 1 \tag{5.44}$$

$$\sum_{i \in V_m} x_{ik} \leq 1 \qquad \forall k \in V_m \tag{5.45}$$

$$\sum_{i \in V_m} x_{ki} \leq 1 \qquad \forall k \in V_m \tag{5.46}$$

considering the navigation path, and:

$$\sum_{i \in V_f} x_{ik} = \sum_{i \in V_s} x_{ki} \qquad \forall k \in V_m \tag{5.47}$$

$$\sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \leq e_{max} \tag{5.48}$$

where V is the set of inspection points, with s being the starting point and f being the ending point, r_i is the *reward* (or profit) of visiting the i-th point and x_{ij} is a binary variable denoting the existence of a path between two inspection points included in a proposed solution. Equation (5.42) represents the total number of IPs to be visited while maximizing the vehicle's profit along the proposed navigation path.

$$u_i - u_j + 1 \leq (n-1)(1 - x_{ij}) \qquad \forall i, j \in V \tag{5.49}$$

$$0 \leq u_i \leq n \qquad \forall i \in V \tag{5.50}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in V \tag{5.51}$$

Derivation of navigable paths is outside of the scope of this work and they are assumed to be known using geographic information about the mission area. The rest of the formulation is given by the additional constraints (5.43) to (5.51) where n is the total number of points, V_s is the set of inspection points excluding the starting s, V_f the set of points excluding the ending f, V_m the set of inspection points excluding both s and f. Furthermore, (5.52) denotes the energy-cost for the trajectory leg going from i to j, characterised by its length d_{ij} and its navigation heading ψ_{ij}. Altogether this term is derived using the runtime energy estimations introduced in the previous sections.

$$c_{ij} = d_{ij} \cdot \varepsilon_{nav}(\psi_{ij}) + \xi_e \tag{5.52}$$

In this formulation constraint (5.48) ensures that the maximum available energy e_{max} is not exceeded. Equation (5.43) ensures that there is no entry path to the starting point s and no exit path from the ending point f. This makes them, respectively, the first and the last point of the navigation route. Equation (5.44) enforces their presence in the solution. Constraints (5.45), (5.46), (5.47) allow the omission of inspections points from the solution. Equation (5.49) is a sub-tour elimination constraint forcing all the points to be visited by one path. Finally, constraint (5.50) bound the values of problem variables.

The use of those constraints allow enforcing the existence of a known ending point in the solution. This is chosen by end users, for instance, in an area suitable for the recovery of the vehicle. Furthermore, a solution is considered valid only if enough energy is available to complete inspections along the proposed route. If the full exploration of the mission space is not possible solutions with fewer inspection points are considered.

Energy and Time Constraints

The maximum allowed energy (5.53) is calculated at runtime before running this procedure. It includes the residual battery energy e_{bat}, derived using a battery model and energy measurements as discussed in section 6.3, some contingency reserve e_{res}, defined by operators, and a *worst-case* estimation ($\beta \geq 3$) for the remaining non-navigation tasks.

$$e_{max} = e_{bat} - \sum_{i=0}^{K} e_{ins_i} - e_{res} \tag{5.53}$$

$$e_{ins_i} \approx \hat{\mu}_{ins} + \beta \cdot \hat{\sigma}_{ins} \tag{5.54}$$

Along with this energy constraint the total navigation time can be derived using a similar approach. This is expressed analytically as:

$$t_{nav} = \sum_{i \in V} \sum_{j \in V} t_{ij} x_{ij} \tag{5.55}$$

$$t_{ij} = \frac{d_{ij} - d_{acc} - d_{dec}}{v_{cruise}(\psi_{ij})} + t_{acc} + t_{dec} \tag{5.56}$$

Knowledge of this quantity allows users to optionally introduce an additional constraint on the expected navigation time given a possible solution. Analytically this is written as:

$$\sum_{i \in V} \sum_{j \in V} t_{ij} x_{ij} \leq t_{max} \tag{5.57}$$

where t_{max} is the allowed navigation time for the current mission.

5.3.2 Energy-Aware Correlated Orienteering Problem

Along with a formulation derived from the standard OP this framework introduces also another variant based on the more sophisticated Correlated Orienteering Problem (COP). This introduces a term that models the capability of gathering information about neighbouring points of interest while conducting the navigation allowing, thus, to further optimize the resulting travel sequence. Incorporating such an aspect with energy estimations done at runtime leads to the implementation of another algorithm known, in this context, as Energy-aware Correlated Orienteering Problem (EA-COP).

The correlated problem is described with a MIQP formulation that tries to maximise the following objective function:

$$\sum_{i \in V} (r_i x_i + \sum_{j \in V_{N_i}} r_j n_{ji} x_i (x_i - x_j)) \tag{5.58}$$

Equation (5.58) represents the sum over the rewards of all visited points where V is set of all IPs, r_i the reward of visiting the point i and x_i is a binary variable denoting that the element i is visited in a solution. As mentioned previously this formulation takes into

account a number of points in the neighbourhood of i. These are represented with the set V_{N_i}. The utility gained from this spatial relationship is calculated as the sum of the rewards for visiting each point j in the neighbourhood multiplied by a weight n_{ji} and a quadratic term $x_i(x_i - x_j)$. The latter term ensures that extra reward, produced by correlation, is only added for vertices that are not visited in the current solution. This extra utility is then added to what is obtained by visiting the original point i. The coefficient n_{ji}, instead, regulates how much reward from j is gathered when observing j from i. These coefficients are calculated taking into account a sensor model that describes the capabilities of on-board payload. On the other hand, these can be also described incorporating knowledge of a specific inspection mode (e.g. long-range sensing versus short-range one). In this framework n_{ji} coefficients are defined using an exponential fading model:

$$n_{ji} = \frac{e^{-\alpha_n \cdot d_{ji}}}{\sum_{k \in V_{N_i}} e^{-\alpha_n \cdot d_{ki}}} \tag{5.59}$$

This takes into consideration the distance d_{ji} between point j and i and a parameter α_n which include the maximum sensing range d_{sens}.

$$\alpha_n = -\frac{\ln(0.01)}{d_{sens}} \tag{5.60}$$

Moreover, coefficients n_{ji} are normalised to balance the reward gained from a single neighbour with respect to one gained over the full set of point V_{N_i}.

In this variant of the COP the vehicle is assumed to start and finish its mission in points defined before deployment. This, as shown for the EA-OP, is enforced by extra constraints (5.61) and (5.62):

$$\sum_{i \in V_s} x_{is} = \sum_{i \in V_f} x_{fi} = 0 \tag{5.61}$$

$$\sum_{i \in V_s} x_{si} = \sum_{i \in V_f} x_{if} = x_s = x_f = 1 \tag{5.62}$$

where the binary variable x_{si} denotes the visit of i after the starting point s and x_{if} enforces that the finishing point is visited after i. From a general point of view, the binary variable x_{ij} describes the existence of a valid path between i and j. For the remaining points, instead, the constraints (5.63), (5.64) and (5.65) are considered:

$$\sum_{i \in V_m} x_{ik} = x_i \leq 1 \quad \forall k \in V_m \tag{5.63}$$

$$\sum_{i \in V_m} x_{ki} = x_i \leq 1 \quad \forall k \in V_m \tag{5.64}$$

$$\sum_{i \in V_f} x_{ik} = \sum_{i \in V_s} x_{ki} \quad \forall k \in V_m \tag{5.65}$$

Constraints (5.63) and (5.64) allow points to be left out from a possible solution, enforcing also that a inspection point is visited at most once. Constraint (5.65) guarantee that after visiting an IP navigation should continue towards another point, unless the finish location has been reached or other constraints are violated. Constraint (5.66) introduces an upper-bound for the vehicle's energy usage:

$$\sum_{i \in V} x_i c_i + \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \leq e_{max} \tag{5.66}$$

this, similarly for what described in the EA-OP case (5.53), guarantees to keep resource usage under control. In (5.66) the previous energy constraint is further extended: the term $x_i c_i$ takes into account the fixed cost of completing the inspection task for the point i and the term $c_{ij} x_{ij}$, instead, the cost of travelling between i and j as shown previously in (5.52). Fixed costs are not added for the initial and final points.

$$u_i - uj + 1 \leq (|V| - 1)(1 - x_{ij}) \qquad \forall i, j \in V, i \neq j \tag{5.67}$$

$$0 \leq u_i \leq |V| \qquad \forall i \in V \tag{5.68}$$

Finally, constraints (5.67) and (5.68) are introduced to allow only the evaluation of a single path solutions, preventing smaller disjoint tours from being considered.

5.3.3 Open Vehicle Routing Problem

Beside the Orienteering Problem (OP) and the Correlated Orienteering Problem (COP) formulations a simple Open Vehicle Routing Problem (OVRP) is also introduced with selected starting and exiting points. This tries to minimise the cost of travel through all inspection points as defined in the following objective function:

$$\sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \tag{5.69}$$

$$s.t. \sum_{j=1}^{n} x_{js} = 0 \tag{5.70}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = \{1, \ldots, n\} \text{ and } j \neq s \tag{5.71}$$

$$\sum_{j=1}^{n} x_{fj} = 0 \tag{5.72}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = \{1, \ldots, n\} \text{ and } i \neq f \tag{5.73}$$

$$u_i - u_j + n \cdot x_{i,j} \leq n - 1 \qquad \forall (i, j) \in A \text{ and } i \neq j \tag{5.74}$$

where c_{ij} is the cost of travelling from point i to point j, x_{ij} is a binary variable denoting the existence of a path between the two points in a proposed solution and V is the set of n

points. Equation (5.69), from a general point of view, describes the total cost of traversing all the inspection points. Constraint (5.70) ensures that there is no entry path to the start point, (5.71) enforces a single entry point to each of the other points and (5.72) ensures that there is no exit path from the extraction point, making it the final point of the route. Constraint (5.73), on the other hand, allows only one exit path from all the other points. Finally, the constraint (5.74), eliminates the presence of any sub-tours from valid solutions.

The OVRP formulation is used, in this framework, as a comparison term for the introduced energy-aware variants of the OP and COP methodologies. The OVRP relies only on few constraints and takes only into account a generic cost coefficient related to locomotion among inspection points. This can be seen as a first technique to identify a possible inspection route for a generic mission that does not take into account any other environmental effects. Results for problem's comparison are presented in section 7.3 in the case of inspection-class missions conducted with a simulated AUV where its energy model has been derived using field experiments.

5.4 Summary

In this chapter a *runtime performance estimation* framework has been introduced. This monitors the vehicle's behaviour while operating in unknown environments. A few relevant features are collected during navigation and a non-linear regression procedure is used to correlate measurements with performance metrics that takes into account the presence of external disturbances, such as sea currents. Runtime knowledge of the *energy usage per unit distance* and *average cruise speed* are then used to analyse the feasibility of missions. These are described using a discrete model where individual tasks are assumed independent and are represented using a stochastic approach. In this work inspection missions are focused explicitly. Those are characterised by two main types of tasks: *navigation* and *inspection*. The former deals with locomotion operations and it is mainly affected by external disturbances. The latter, instead, represents all the operations conducted when gathering information about the environment. This is described as a repeatable task or, in other words, as an action that shows the same behaviour every time is conducted. As such, *inspection* tasks are represented using a simplified model that describes their expected execution time and energy usage.

Beside modelling aspects further details have been discussed regarding the procedures followed in derived runtime estimations. Relevant is the use of a pruning procedure before conducting the regression analysis. Such an approach allows the derivation of a dataset of collected features that gives more priority to newer measurements while discarding older ones. Samples are collected during navigation using a trajectory segmentation procedure. This guarantees that features are grouped together only when they belong to a segment that has small variations in terms of heading and speed. Such an approach prevents

measurements taken during acceleration, deceleration or manoeuvring phases from being included in the presented estimations. Those phases, while still relevant, represent a small percentage of navigation time and are represented as higher-order effects. Experimental validation of the proposed procedure is presented in section 7.2 with the use of two real AUVs conducting field missions.

After introducing the runtime estimations these are employed in the context of route optimisation. This has been discussed in the final sections of this chapter. Route optimisation is used in this framework to improve the use of resources, allowing to adjust at runtime the sequence of tasks conducted by the vehicle. This is implemented using algorithms derived from the Orienteering Problem that includes estimations done at runtime. These are the Energy-aware Orienteering Problem (EA-OP) and Energy-aware Correlated Orienteering Problem (EA-COP) that are compared with a classical Open Vehicle Routing Problem in a set of simulation experiments discussed later in section 7.3. The introduced EA-OP includes a set of constraints customised for the underwater mission's context. Those allow end users to enforce the existence of an initial and a final point (e.g. for recovery purposes) in the calculated solutions. Moreover, a maximum energy budget and the concept of *reward* gathered when conducting an inspection task are taken into account when evaluating possible solutions. The EA-COP, instead, extends such an algorithm considering the capability of collecting information also from neighbouring points. This aspect, while not necessarily applicable for large scale inspection, becomes more important when considering inspections of underwater structures [129] where multiple observations of targets are taken into account. This variant modifies some of the original constraints, replacing them with a more expressive, yet computational intensive, condition. The EAOP is implemented using a MILP formulation, the EACOP, instead, with a MIQP one.

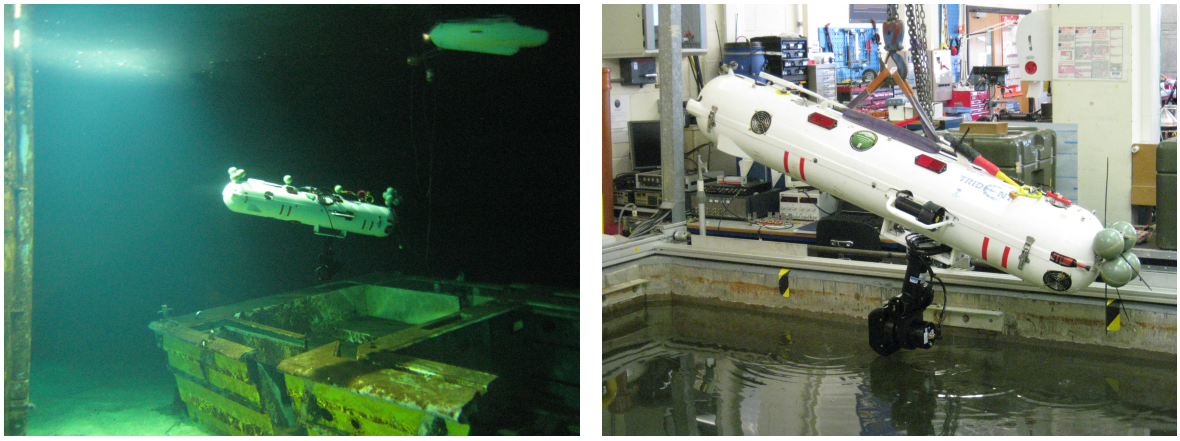
Chapter 6

Experimental Platform

"I'm very sorry", the drone said, without a trace of contrition.

Iain M. Banks – The Player of Games

The main experimental platform used in this work is the Nessie VII AUV, a research prototype originally developed in the Ocean System Laboratory [25]. This underwater platform is a torpedo-shaped vehicle with hover capabilities designed with inspection and intervention missions in mind. This platform has been upgraded over the last few years while being involved in several research projects [1], [2], [130] and European competitions [131] (e.g. SAUC-E and Eurathlon). Such a vehicle represents an ideal candidate for the practical implementation of an energy-aware architecture where dedicated hardware components and software modules are validated in the context of real field missions. Together with the Nessie AUV another underwater vehicle has been involved in the experimental validation of the proposed architecture. This is the IVER3 AUV [32], a commercial platform used for oceanographic surveys recently acquired by the Ocean



(a) **(b)**

Figure 6.1: Nessie AUV during indoor field trials (a) and during laboratory testing (b) with an external imaging sensor (multi-beam sonar) with pan and tilt capabilities.

Systems Laboratory. The availability of such a vehicle allowed the further analysis of the use of an energy-aware framework with an off-the-shelf platform designed for specific uses in the marine domain. In the following sections a detailed analysis of the Nessie AUV platform is introduced. First, its principal design features are discussed. Second, the experimental work done on this platform is described together with modelling and experimental validations. Finally, the existing vehicle's software architecture is analysed together with the integration of additional modules required by the proposed energy-aware architecture.

6.1 Vehicle Design

The Nessie AUV is a torpedo-shaped vehicle with hover capabilities designed with inspection and intervention missions in mind. This has a 0.30 m diameter, it is 1.75 m long and it weights almost 60 kg. An anodized aluminium hull allows a maximum operating depth of about 70 m to 100 m. A plastic outer shell provides protection for an internal frame and improves the vehicle's dynamic efficiency (e.g. reduced drag effects). The use of an internal hard plastic frame supports the attachment of wet-side components, such as actuators and external sensors, to the main pressure hull. The vehicle is powered by a set of four high-energy LiPo batteries, each with 25.9 V nominal voltage, 21 A h (about 543.9 W h) of stored charge and rated for 18 A maximum discharge current. This gives a total available energy of about 2.2 kWh, enough for conducting missions in a harsh environment for 6 h to 8 h with external payloads attached without the need of recharging.

Spec	**Value**
Length	1.75 m
Diameter	0.30 m
Weight	60.0 kg
Maximum Depth	70 m
Maximum Speed	2.0 m/s (3.8 kn)
Maximum Duration	6 – 8 h
Stored Energy	2.2 kWh
Operating Voltage	24 – 28 V

Table 6.1: Principal characteristics of Nessie VII AUV.

Propulsion is given by six brushless DC marine thrusters, Seabotix HPDC1502, each capable of about 4.5 kg of output thrust in ideal conditions while using 250 W of power. These thrusters are aligned with the principal axes of the platform, two for the longitudinal, two for the lateral and two for the vertical axes, and allow the control of five degrees of freedom (DOFs): surge, sway, heave, pitch and yaw. This configuration is a trade-off between operational requirements (e.g. hovering capabilities, lateral motion, precise station-keeping) and design constraints. Figure 6.1 shows Nessie AUV in its configuration

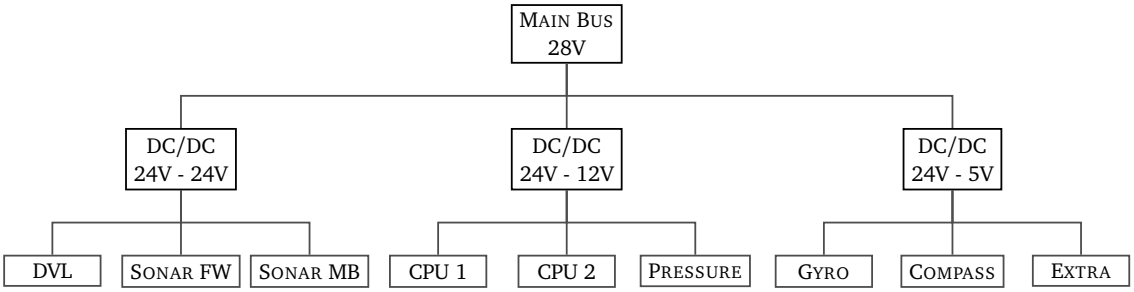


Figure 6.2: Overview of Nessie AUV's sensor and payload configuration.

for the EU FP7 PANDORA project. During field trials this vehicle has been fitted with an addition imaging sonar with pan and tilt capabilities attached to its bottom side. Such an addition allows the vehicle to carry out detailed analysis of underwater objects or structures with the capability of focusing [132] the imaging sensor on specific portions of the environment. This vehicle also supports tethered operations where end users can monitor or control specific aspects of the platform operating in the field. Tethers are mostly used for providing high bandwidth telemetry data during autonomous missions and are not required for standard operations.

6.1.1 On-board Sensors

This platform includes several navigation sensors, such as a Fiber Optic Gyroscope (FOG), a Doppler Velocity Log (DVL), a pressure sensor, a GPS unit and a 3-axis digital compass, as well as communication devices, such an acoustic underwater modem and Wi-Fi capabilities to allow remote on-surface operations. These are listed in Table 6.2, where their electrical specifications are also reported. These aspects are useful to quantify their impact on the platform's energy consumption during field operations. On the other hand, operational experience in the Scottish sea has shown that their effective performance is affected by environmental effects (e.g. temperature, humidity, etc.) other than by configuration parameters used for a specific type of mission (e.g. DVL ping delays, sonar refresh rate). Sensors are connected to the main electrical bus using DC-DC converters with thermal

Sensor	**Voltage** (V)	**Current** (A)	**Power** (W)
Teledyne Explorer DVL	24	0.20 - 1.50	4
Blueview P900	24	0.60 - 0.80	15
Blueview MB2200	24	0.85 - 1.05	25
KVH DSP-3000 FOG	5	0.40 - 0.60	2
PNI TCM Compass	5	0.20 - 0.22	1
Keller 33X Pressure	12	0.10 - 0.20	1.25
Embedded GPS Unit	5	0.08 - 0.12	0.75

Table 6.2: Overview of Nessie AUV main on-board sensors. Average power is calculated by taking into account sensors' duty cycles during normal operations.

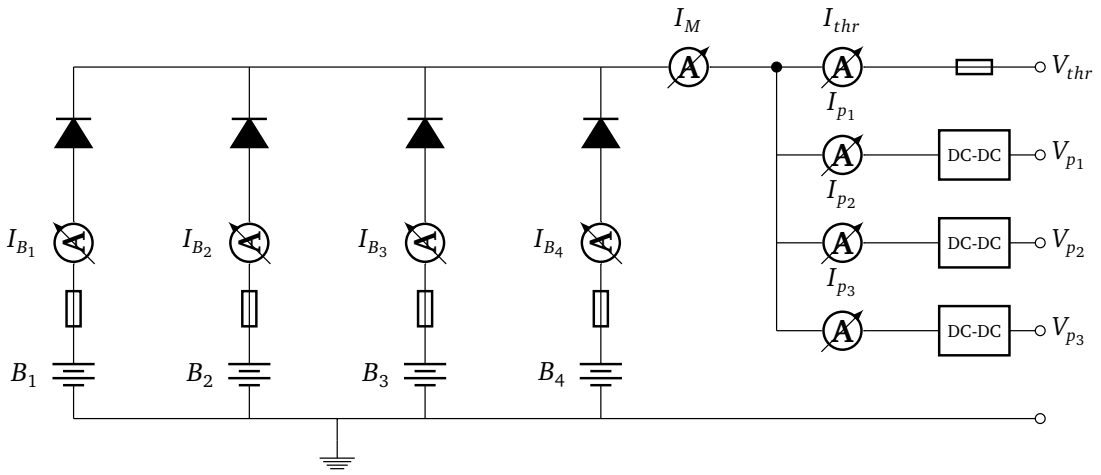


Figure 6.3: Overview of Nessie AUV's electrical schema. Highlighted are the current sensors used for monitoring individual batteries and other relevant electrical buses. Parallel diodes allow safe operations even in presence of unbalanced battery voltages.

protection. These adjust the supply voltages to power group of devices sharing a common power rail. An outline of device connections is shown in Figure 6.2. Navigation sensors are a requirement for correct operations and, for this reason, are considered part of the platform's baseline power consumption. On the other hand, external sensors, such as an imaging sonar or extra scientific payload, are considered additional loads that will be analysed in the context of mission's tasks.

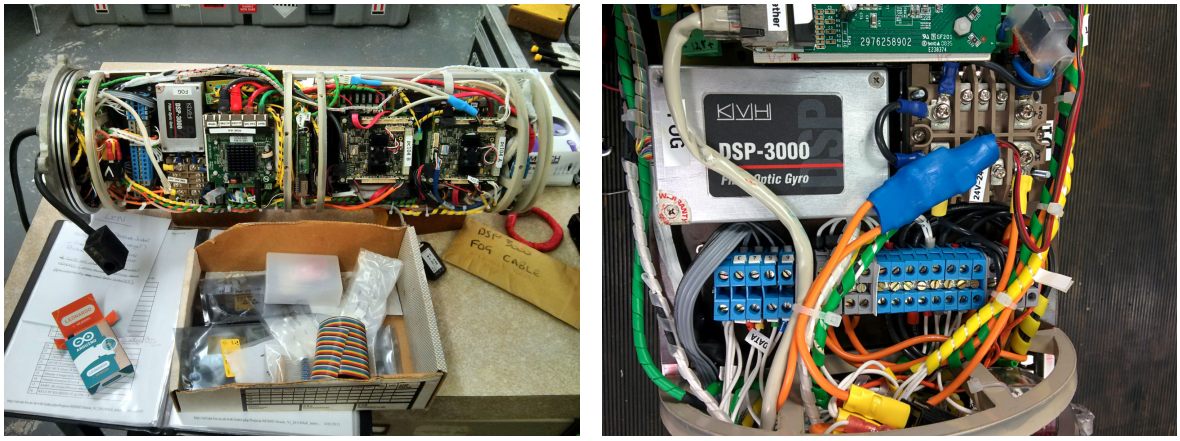
6.1.2 Electrical Schema

A relevant design aspect of the Nessie AUV platform is its electrical system. In this design four batteries are connected to the main bus using protection fuses and low voltage drop diodes. This approach allows feeding the thruster subsystem with an unregulated supply of about 28 V, well within the operating range of on-board actuators. At the same time, DC-DC converters are used to derive regulated supply for vehicle's sensors and control or on-board payload computers.

Such an approach, shown schematically in Figure 6.3, has been proven robust during several years of field operations and no severe failures have been traced back to this system. On the other hand, given the electrical bus design it can be seen that few metering points are effectively required to monitor the platform's overall energy consumption and the one of its subsystems. In fact, with Figure 6.3 in mind, one can notice that the I_M represents the overall current flow within the platform. This is given as a linear combination of currents I_{B_x} drawn from batteries and as combination of on-board loads:

$$I_M = I_{B_1} + I_{B_2} + I_{B_3} + I_{B_4} \tag{6.1}$$

$$I_M = I_{thr} + I_{p_1} + I_{p_2} + I_{p_3} \tag{6.2}$$



(a) (b)

Figure 6.4: Overview of Nessie AUV's dry section electronics (a) and on-board current sensor prototype integration (b). The sensing device is covered with heat shrink tubing to isolate its exposed high current connections.

where I_{thr} represents the current drawn from the propulsion subsystem and $I_{p_1}, I_{p_2}, I_{p_3}$ the current from other on-board devices. Knowledge of such currents allows the energy consumption to be derived for each subsystem using, for instance, a Coulomb-counting (CC) procedure [44]. This calculates the energy usage by integrating voltage and current measurement taken on a specific electrical bus. Analytically this is written as:

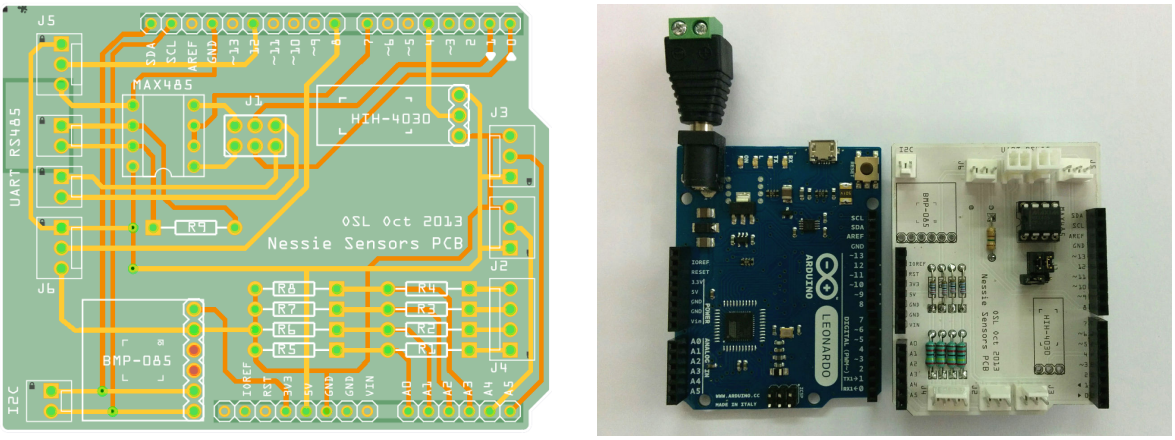
$$e[t] = \sum_{k=0}^{t} \Delta e[k] = \sum_{k=0}^{t} V[k] \cdot I[k] \tag{6.3}$$

where $e[t]$ represents the overall energy usage at time t and $V[k]$, $I[k]$ the bus voltage and current, respectively, at the instant k.

6.2 Energy Monitor

With those aspects in mind a dedicated energy monitoring system has been introduced in the Nessie AUV's design. This, known as *low-cost energy monitor* (LEM), is built around the use of an off-the-self microcontroller board (i.e. Arduino Uno) and few external current sensors placed on relevant sections of the vehicle's electrical bus. Current sensing devices are able to convert the current's flow into an output voltages proportional to its intensity. These are acquired together with bus voltage measurements using the microcontroller's internal ADC. Current values are later derived by applying the sensor's model (e.g. ideally a linear relationship between output voltage and sensed current) to collected samples. The LEM is connected to on-board control computers thus providing runtime measurements to the vehicle's software architecture[1]. Figure 6.4 shows the integration of a LEM prototype inside the dry section of Nessie AUV, where a current sensor, within its protective shielding,

[1]In this case a dedicated software module has been integrated to analyse and track collected samples such as voltages, currents and energy usage of vehicle's subsystems.



(a) **(b)**

Figure 6.5: Schematic (a) and manufactured prototype (b) for low-level data acquisition platform beside an Arduino board used for control purposes.

is connected in series with the main power bus. Figure 6.5 shows instead the LEM prototype manufactured as expansion board (i.e. shield) to be stacked on the existing microcontroller platform.

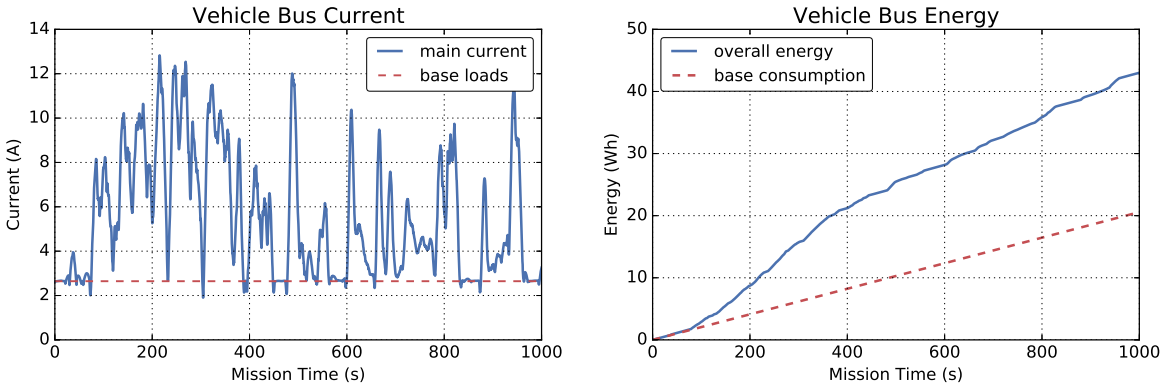
After an initial study of available current sensing technologies [47], also mentioned in Section 2.2.5, an automotive-grade Hall-effect linear current sensor (Allegro's ACS715 / ACS714) is chosen. This provides good isolation [48] between the high-current monitored bus and low-voltage instrumentation side, good sensing quality and it is manufactured in surface integrable package (i.e. 8-lead SOIC) and also available as small standalone board (i.e. 30 × 30 mm). The ACS715 component is rated for maximum 30 A and it has

Section	**Label**	**Current** (A)	
		avg.	*max.*
Main Bus	I_M	16.0	30.0
Thruster Bus	I_{thr}	10.0	24.0
Battery Bus	I_{B_x}	4.0	18.0
Payload Bus	I_{p_x}	2.0	7.0

Table 6.3: Nessie AUV's electrical bus currents. Average values are measured during low-speed operations in controlled environments. Maximum values are derived from loads specifications.

a 135 mV/A sensitivity. The ACS714, instead, is rated for maximum 7 A, a 188 mV/A sensitivity and it supports bidirectional measurements. The first sensor is employed to measure high currents, such as I_M and I_{thr}, while the second one is used on lower current buses as suggested by early measurements reported in Table 6.3. Sensors are used with a 10-bit ADC allowing resolutions of about 72 mA and 52 mA, respectively, after noise filtering. To further improve the precision of the CC procedure a trapezoidal integration is implemented with measurements taken at fixed sampling rate F_s of 10 Hz. Analytically this is written as:

$$e[n] = \sum_{k=1}^{n} V[k] \frac{I[k] + I[k-1]}{2} T_s \tag{6.4}$$



(a) **(b)**

Figure 6.6: Energy and current measurements of Nessie AUV. Plot (a) shows the main bus current I_M. Plot (b) shows the calculated energy consumption. Both plots highlight the effect of platform's base loads with respect to the overall trends.

where V is the bus voltage, I the sensed current and $T_s = 1/F_s$ is the sampling time. Such an approach is implemented inside the MCU and aggregated measurements (e.g. energy usage, average voltages and currents) are reported back to the vehicle's main control computer. Figure 6.6 shows measurements collected during a real sea experiment while the vehicle is navigating at low speed in presence of tidal currents. In this case the current I_M is measured and it is characterised by a constant load of about 2.6 A. Such an offset is given by the platform's navigation sensors, sonar and on-board computers. This characterise the *idle* behaviour of Nessie AUV, where a constant usage of about 80 W to operate the sensors is needed during autonomous missions.

6.3 Battery Model

The introduction of a standalone energy monitor, while providing useful information about vehicle's subsystems, allows better characterisation of the state of on-board batteries. In fact, once runtime measurements, such as bus voltages and currents, are collected for individual batteries they can be combined together with analytical models to obtain a robust estimation of their SOC. Such a behaviour represents an important feature for the energy-aware architecture because it allows autonomy software modules to evaluate the vehicle's behaviour in presence of accurate estimations of the availability of on-board resources.

With those aspects in mind after integrating the LEM inside the vehicle's architecture an identification procedure has been conducted for the vehicle's batteries. Figure 6.7 and 6.8 shows, respectively, an excerpt of the charge and discharge profiles collected during identification experiments with the Nessie AUV. These are conducted in a controlled environment using a HIL configuration and involving the full vehicle in the testing. Batteries are charged individually using an off-the-shelf Li-ion battery charger limited at 6 A, the same used for field operations. After reaching a charged state the vehicle's

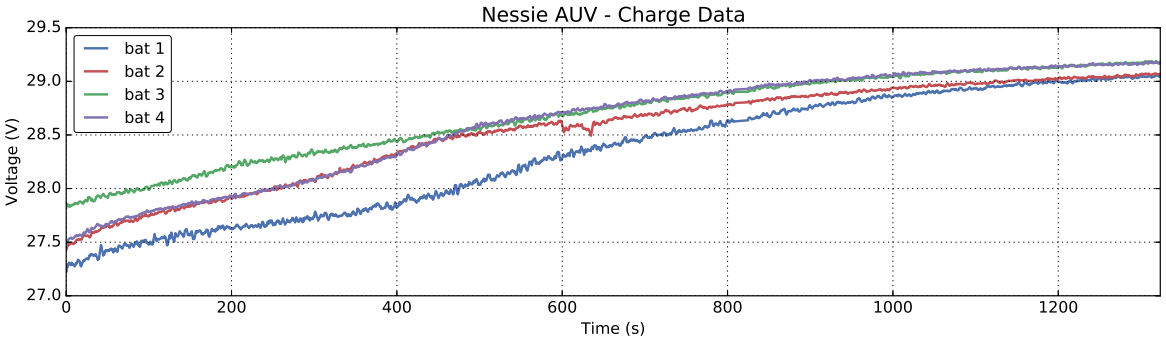


Figure 6.7: Nessie AUV's battery charging profile. Shown is the final or *top-up* phase covering from 75% to 100% of battery packs' capacity. Batteries behave differently according to their effective health status and internal resistance.

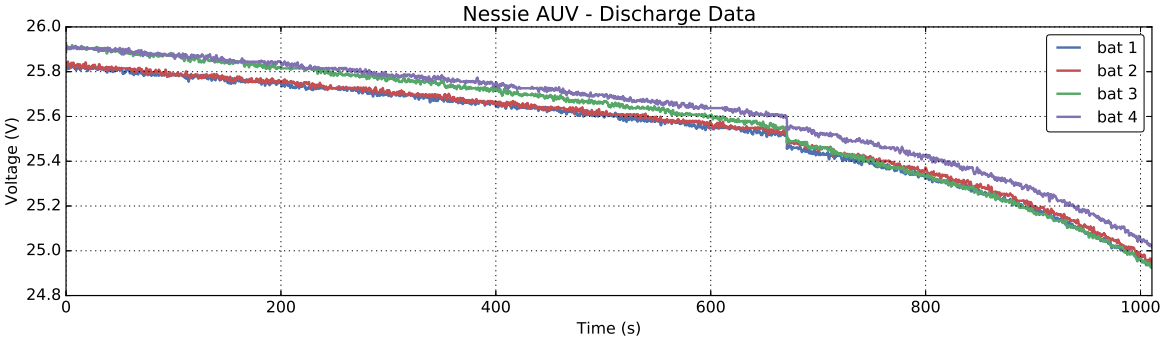


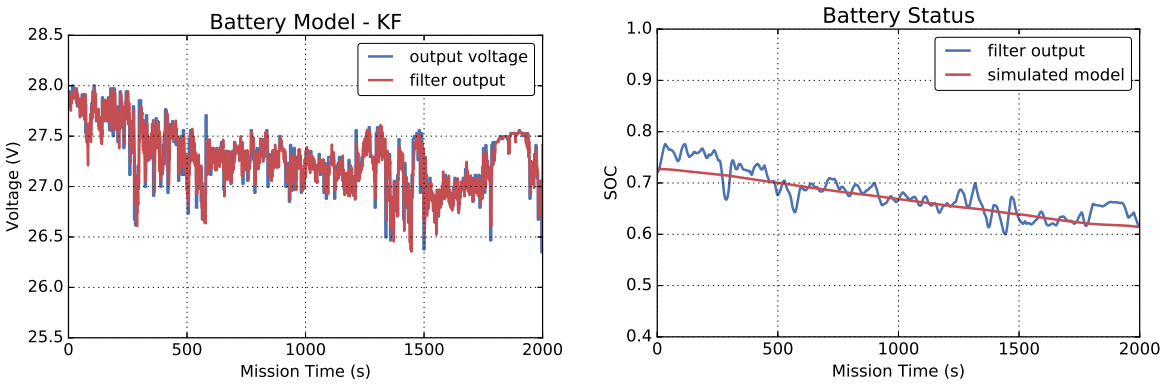
Figure 6.8: Nessie AUV's battery discharge profile. A constant load of about 3 A is applied to the battery system. The transition between linear and non-linear profile is clearly captured by the experiment.

is deployed in an indoor testing tank and left hovering at a fixed depth, simulating the submerged operations with all navigation and imaging sensors turned on. Concurrently, measurements are collected by the LEM and stored within the on-board computers.

Consequently a model fitting procedure is conducted on collected samples using an existing battery model adjusted for the vehicle's specification. As previously mentioned in Section 2.2.1 several models are available in literature, however many of those are focused on the analysis of a single cell battery and require the estimation of many parameters. In the case of Nessie AUV individual batteries are built by combining multiple cells together, specifically, in a 7S configuration (e.g. seven cells in series). This required the use of a battery model that can equally be applied in presence of multi-cell configurations and

Element	**Voltage** (V)			**Capacity** (A h)	**Energy** (W h)
	min	*nominal*	*max*		
Single Cell	2.75	3.7	4.2	21.0	77.0
Battery Pack (7S)	19.25	25.9	29.4	21.0	543.9

Table 6.4: Nessie AUV's battery specifications.



(a) **(b)**

Figure 6.9: Validation of Nessie's AUV battery model. Plot (a) shows the measured voltage together with estimations done with a Kalman Filter (KF). Plot (b) shows the estimated state of charge (SOC) together with offline model simulations. Noisy measurements increase the residual uncertainty in SOC estimations. Furthermore, the tuned model allows the representation of the battery system's behaviour with convenient accuracy also in simulated scenarios.

it can be identified only in presence of few[2] measured variables, such as open-circuit voltages, bus currents and voltages.

For those reasons the NREL battery model [37] is chosen to characterise the vehicle's batteries. This model, already presented in Figure 2.2, is extended to an 7S configuration and an equivalent lumped-parameters schema is assumed for an individual full pack. Table 6.4 shows single cell's characteristics together with the ones for the derived battery pack. Parameters of the resulting battery model are reported in Table 6.5. These are found using optimization procedures that minimise the resulting Mean Squared Error (MSE) for the output voltage. Using these parameters the model's average error is kept within a 0.5 V range.

Parameter	Value
R_c	0.40 mΩ
R_e	105.00 mΩ
R_t	98.00 mΩ
C_c	203.69 F
C_b	9309.59 F

Table 6.5: Nessie AUV's battery model parameters.

After obtaining a valid battery model a SOC estimation procedure is also implemented in the vehicle. This, as discussed previously in Section 2.2.2, is based on the use of a Kalman filter (KF) to combine the lumped parameters models with LEM measurements collected on-board during field operations. Figure 6.9 shows an experimental validation of the proposed procedure on Nessie AUV during a data collection mission. Chart 6.9a shows the filter's estimated battery output voltage together with real measurements taken

[2]Some models rely on internal variables that can not be directly measured when the battery is integrated in the vehicle and can only be estimated using dedicated experiments in laboratory conditions.

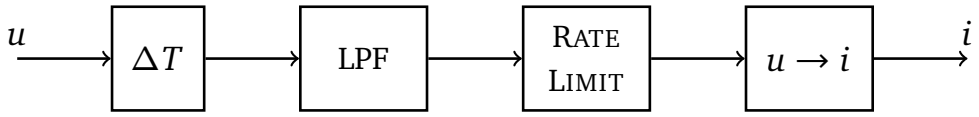


Figure 6.10: Thruster model schema. The *throttle to current* characteristic ($u \rightarrow i$) is applied after introducing the communication delay (ΔT) and non-linear effects of internal speed controllers for a generic brushless DC motor.

by the LEM device. Chart 6.9b shows, instead, the runtime estimated SOC comparing it with an off-line simulation done with recorded data using only the lumped parameters model.

6.4 Thruster Model

Following the identification of battery models further attention is given to the propulsion subsystem. As mentioned in previous sections, such a domain represents one source of operational uncertainty for autonomous underwater vehicles and detailed knowledge of component's characteristics is required to properly assess the vehicle's behaviour and effectiveness when outside in the field. For these reasons an accurate actuator's model has been also identified for the Nessie AUV. As discussed in Section 2.3 several techniques are known in literature to provide analytical representations for marine thrusters. Most assume correct knowledge of internal parameters (e.g. motor's resistance, inductance, etc.) for the actual components and do not include any operational considerations that arise from the use of integrated motor controllers. In the case of Nessie AUV oil-filled off-the-shelf thrusters are used in the vehicle. Those are built with an embedded ESC controller that provides advanced features like stall and ventilation protection. Being in presence of commercial components internal parameters are assumed unknown[3] and a black-box identification approach is followed.

In developing this work a data-driven strategy has been integrated in the vehicle's software architecture. This employs a short self-testing procedure that extracts actuator's models from the data collected by the platform itself. Such a procedure is also employed for calibration purposes, for instance before starting a long-term mission, and allows adjusting the analytical models in accordance to the actual health status[4] of the actuator under test. The built-in identification procedure relies on a discrete elements thruster model, also shown in Figure 6.10. This includes a time delay term (ΔT), representing the communication latency between vehicle's computers and motor controllers, a low-pass filter (LPF) followed by a rate limiter, reproducing the effects of ESC's control logic, and a *throttle to current* characteristic ($u \rightarrow i$). Actuators are controlled using a throttle

[3]Those are often not disclosed by the manufacturer or are of difficult characterization, for instance, requiring invasive procedures and laboratory testing on the actuator's internal components.

[4]Operational experience has suggested that oil-filled actuators, like the ones available on Nessie AUV, degrade sightly their performance within the maintenance's interval of their oil reservoirs.

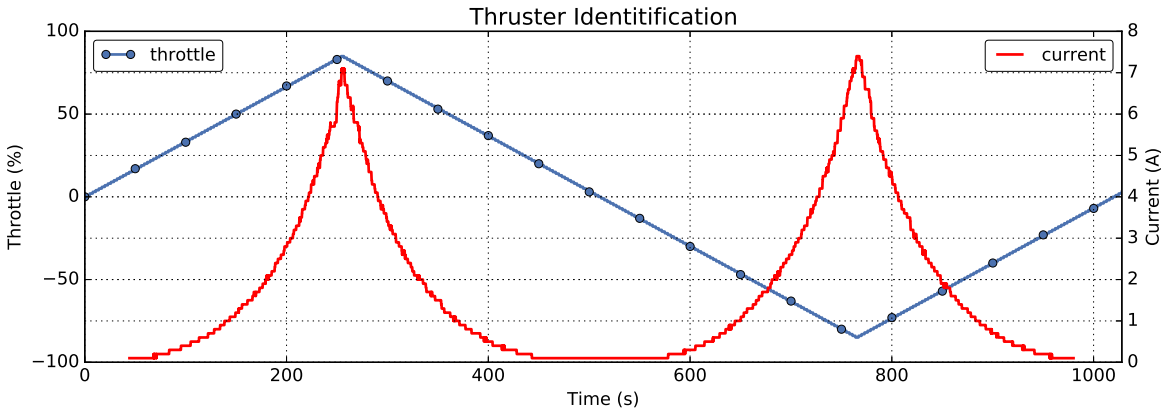


Figure 6.11: Nessie AUV's thruster identification. The plot shows a triangular command signal (blue) together with the current's feedback (red) from the motor controller.

command while their embedded motor controllers provide feedback signals regarding the actuator's behaviour, such as actual throttle (or relative speed), motor currents and temperatures.

A set of known input signals is used to stimulate the on-board thrusters and, at the same time, the vehicle's computers collect these feedback. Such an approach generates a calibration dataset that is later employed to derive the analytical model used to represent each actuator. The input command u is varied across the allowed range a with a repetition period p.

$$u_{step}(t) = a\,\text{step}(t - T) \tag{6.5}$$

$$u_{trig}(t) = \frac{2a}{\pi}\arcsin\left(\sin\left(\frac{2\pi}{p}t\right)\right) \tag{6.6}$$

$$u_{sin}(t) = a\sin\left(\frac{2\pi}{p}t\right) \tag{6.7}$$

Each command is held steady for a small time interval (e.g. 3 s). This allows the actuator's under test to reach its requested throttle command and transient effects to settle. The input signals used in this procedure are: a step function (6.5), a triangular signal (6.6) sweeping between full-forward and full-reverse commands and a slowly changing sinusoidal one (6.7) with full range amplitude and period comparable to the update rate of the control architecture used in the vehicle. Figure 6.11 shows an example of such a training procedure. In this case the forward port-side actuator of Nessie AUV is commanded using a triangular signal while the vehicle is held submerged inside a test tank. The plot shows how forward and reverse operations slightly differ in terms of current usage while in presence of same operating conditions. Such a behaviour is explained by the mechanical design of the actuator itself and it is taken into account during the identification procedure.

6.4.1 Non-linear Model

After collecting a dataset for all actuators a regression analysis is employed to extract their *throttle to current* characteristics. This is conducted using the LWPR algorithm, previously introduced in section 3.3.1, and it allows capturing even small differences between vehicle's actuators without requiring a manual curve fitting procedure. Together with this base characteristic other parameters, such as communication delay or actuator's latency, are estimated at this stage using a time analysis [133] done on signals (6.5) and (6.7). Table 6.6 shows, instead, the set of LWPR hyperparameters used during thruster models identifications.

Parameter	Value
$init\_D$	5.173
$init\_alpha$	1.450
$penalty$	0.0001

Table 6.6: LWPR hyperparameters used for thruster models.

Using the LWPR approach a thruster characteristic is represented as:

$$I(t) = g_m(u_{lim}(t)) \tag{6.8}$$

where i is the current used by the thruster, u_{lim} the input throttle and $g_m(.)$ is the extracted LWPR model. The u_{lim} term is derived from the original control input u by taking into account the effect of communication delay ΔT and actuator's dynamics. First a control delayed signal u_d is defined:

$$u_d(t) = u(t - \Delta T) \quad \Longrightarrow \quad \Delta u = u_d(t) - u_d(t-1) \tag{6.9}$$

such a term helps in the characterization of parameter Δu that represents the maximum allowed difference between two consecutive input commands. Δu is related to internal dynamics of the real actuator which in the presence of fluid friction requires some time to react to commanded throttle requests. Real operations are further influenced by the presence of internal ESC circuits. Their non-linear effects are modelled in this context using the concept of rate limiter, a hard filter that limits the variation of input signals. Given those aspects the u_{lim} term is analytically defined as:

$$u_{lim}(t) = \begin{cases} u_d(t) & \text{if} \quad \Lambda_{low} \leq \Delta u \leq \Lambda_{high} \\ u_d(t) + \Lambda_{high} & \text{if} \quad \Delta u > \Lambda_{high} \\ u_d(t) - \Lambda_{low} & \text{if} \quad \Delta u < \Lambda_{low} \end{cases} \tag{6.10}$$

where Λ_{low} and Λ_{high} represent the maximum rate of change for the original input command u, respectively when increasing or decreasing the thruster output power. Such

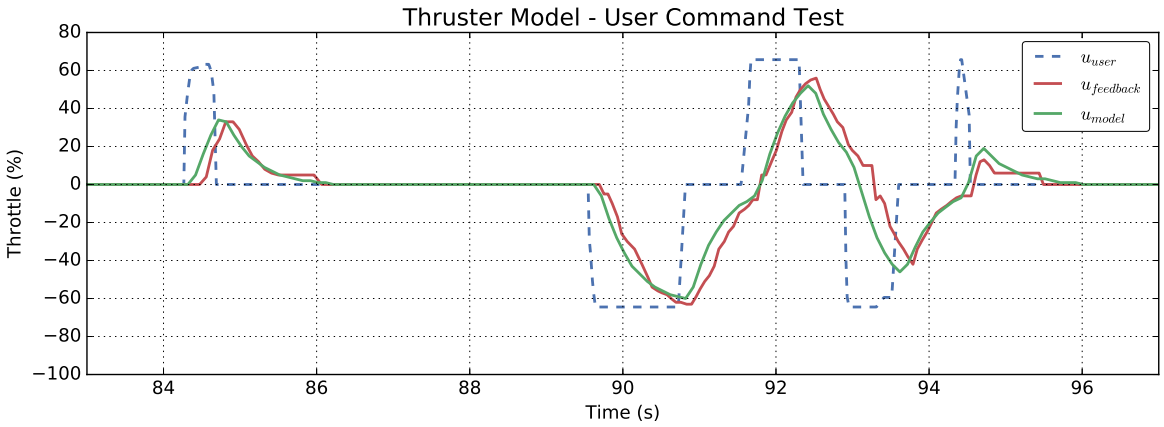


Figure 6.12: Thruster model evaluation with user input commands through the joystick interface. The dashed line represents the user input command, the green solid one represent the model predictions and the red one the delayed feedback from the real actuator.

parameters are numerically estimated during the thruster's testing procedure by using the command signal (6.5). In this case throttle feedback from embedded motor controllers is employed to calculate the *average rate of change* (RC) of u_d.

Parameter	Value
Λ_{high}	3.250
Λ_{low}	4.125

Table 6.7: Average rate of change normalised parameters used in thruster models.

The RC term is defined as:

$$RC = \frac{u_d(t+T_h) - u_d(t)}{T_h} \tag{6.11}$$

where T_h is the measured time needed to correctly change the throttle setting between 5% and 95% of the allowed ranges $(0, a]$ or $[-a, 0)$ when characterising, respectively, forward and reverse thruster's operations. Estimated parameters for the actuators used in this vehicle are shown in Table 6.7 while an example of input command analysis is shown by Figure 6.12 in presence of discrete user control commands.
The estimated characteristic and its calculated LWPR model is shown in Figure 6.13 for one of the Seabotix HPDC1502 thrusters used in Nessie AUV. The LWPR model is calculated by collecting training samples with the vehicle operating fully submerged in a controlled environment thus simulating stable environmental conditions. Raw measurements are characterised by a quantization error, given by finite precision of on-board sensors, and by a low noise figure. Despite this the collected data allows the extraction of good quality models, with a small mean error, for the most used device's operating range[5].

[5]In this vehicle the Seabotix HPDC1502 thrusters are limited to 7.5 A as suggested by manufacturer's recommendations. This aspect bounds the maximum throttle command to about 85% of its input range. Short bursts above 85% are possible but are not used for general navigation in Nessie AUV.

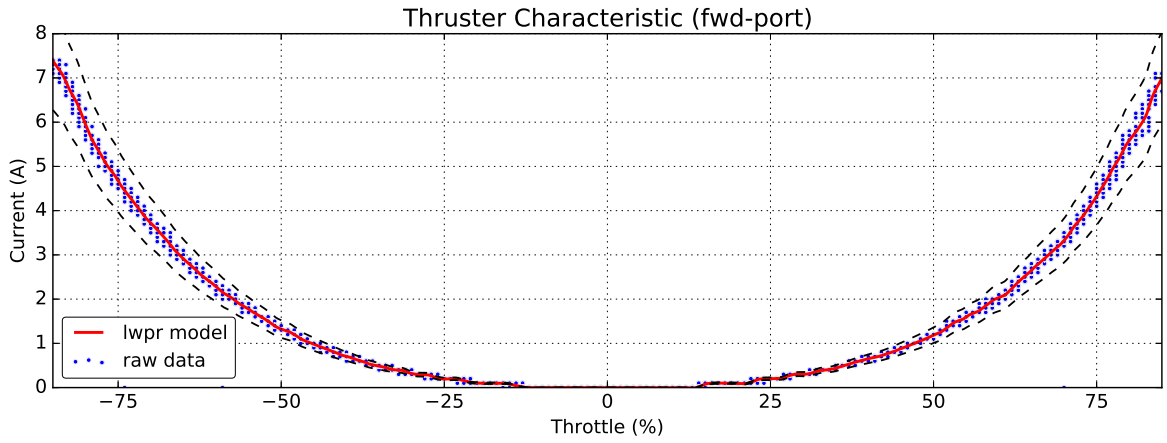


Figure 6.13: Seabotix HPDC1502 current characteristic ($i \rightarrow u$) for the port-side forward actuator of Nessie AUV. Behaviour is non-linear with most of the power being developed at high throttle requests ($u \geq 60\%$).

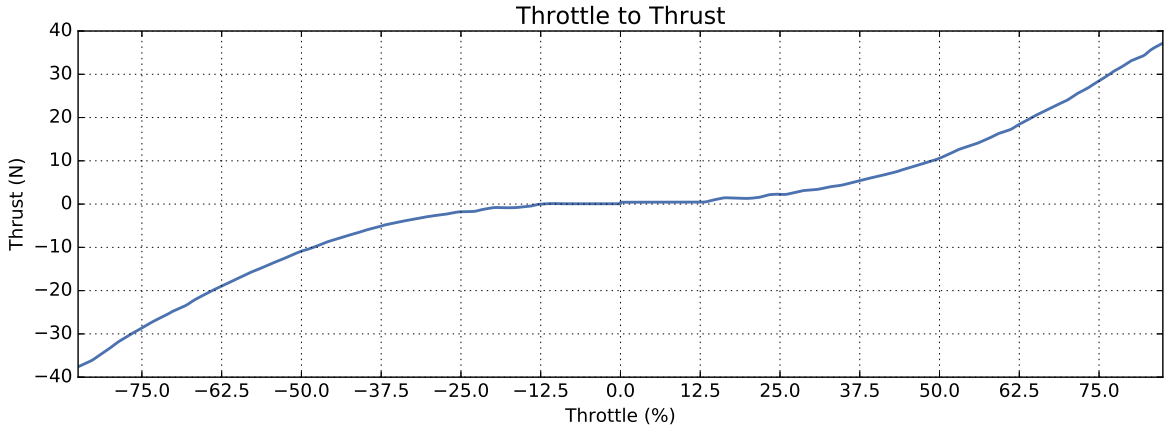


Figure 6.14: Seabotix HPDC1502 throttle to thrust characteristic derived from the LWPR model.

6.4.2 Force Linearization

After identifying models for the six actuators available on this vehicle a *throttle-to-thrust* characteristic is calculated using the LWPR models and the *current-to-thrust* relationship provided by the actuator's manufacturer. Knowledge of this derived characteristic, shown in Figure 6.14, allows a force linearization term to be calculated. This is further employed in the vehicle's control subsystem, briefly outlined in Figure 6.15, to improve the mapping between actuator's force requests and output throttle commands. This takes into account the real behaviour of each actuator.

As shown in the Figure 6.15 schema, a PILIM controller [112] calculates the generalized thrust vector τ that needs to be applied on the vehicle. Later a mapping procedure (known also as *thrust allocation*) distributes the generalized vector τ to individual actuator's requests f still in the force domain. Only after this stage the derived linearization term is employed, converting the forces f into throttle commands u adjusted for each thrusters. Such a strategy is employed because real components deviate from their ideal characteristic once they are used in the field. Experimental evidence shows, in fact, that

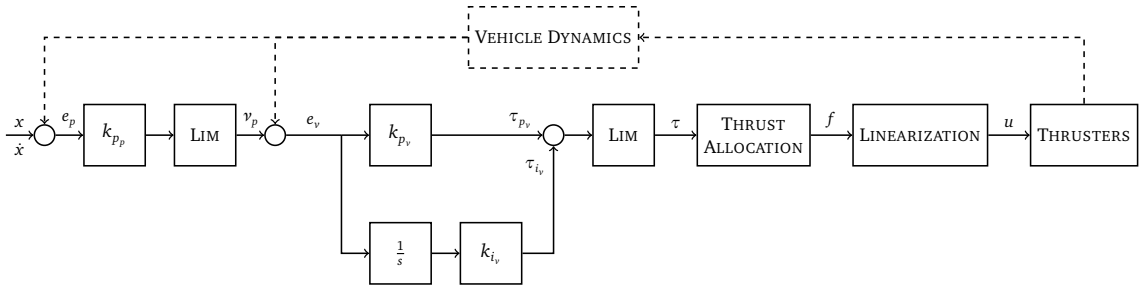


Figure 6.15: Overview of Nessie AUV's control schema implemented using a PILIM controller and linearization terms derived from LWPR models. Coefficients are empirically tuned using experiments conducted in a controlled environment like an indoor tank.

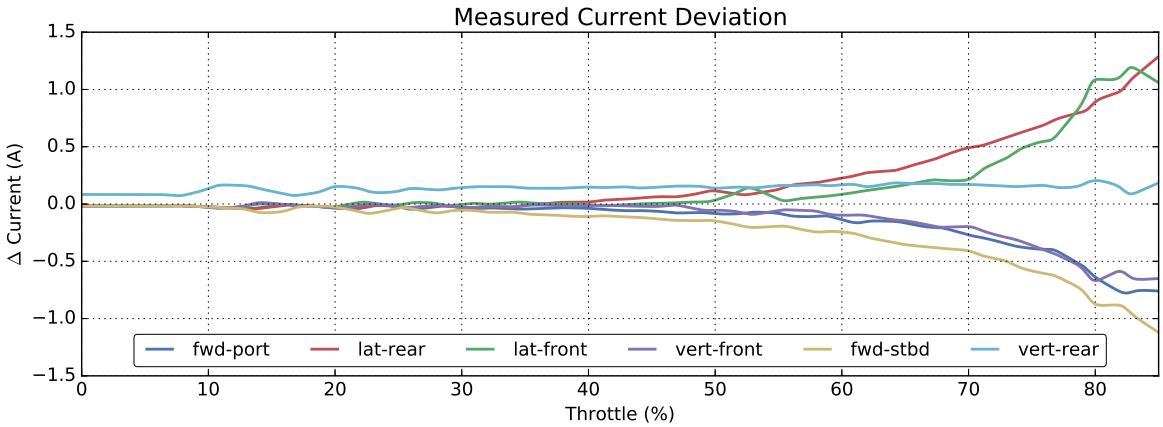


Figure 6.16: Measured current deviation for Nessie AUV's thrusters with respect to a new Seabotix HPDC1502 thruster used as a reference in controlled environment experiments.

Nessie AUV's actuators, almost 4 years old, behave differently with respect to their brand new counterpart taken, for instance, from the vehicle's spares pool. Figure 6.16 highlights this behaviour by showing the effective difference in current draw among vehicle's actuators while running the thruster testing procedure in a controlled environment. As shown in this figure deviations become more evident at high throttle requests where actuators are working close to their recommended current limits of about 7.5 A.

6.4.3 Thrust Allocation

As shown in Figure 6.15, a *thrust allocation* procedure is employed in the control schema to distribute body-fixed frame forces τ into individual actuator's forces. A well-known approach is the use of a *thruster configuration matrix* B derived from geometrical information about relative positions of actuators and the vehicle's centre of mass [77]. This matrix is generally defined, in the case of non-rotatable fixed actuators, as:

$$B_{(6\times N)} = \begin{bmatrix} e_{x_1} & \dots & e_{x_N} \\ e_{y_1} & \dots & e_{y_N} \\ e_{z_1} & \dots & e_{z_N} \\ (r_1 \times e_1)_x & \dots & (r_N \times e_N)_x \\ (r_1 \times e_1)_y & \dots & (r_N \times e_N)_y \\ (r_1 \times e_1)_z & \dots & (r_N \times e_N)_z \end{bmatrix} \tag{6.12}$$

where $e_i = [e_{x_i}\ e_{y_i}\ e_{z_i}]$ and $r_i = [r_{x_i}\ r_{y_i}\ r_{z_i}]$ are, respectively, the orientations and position of vehicle's actuators with respect its centre of mass $c = [x_c\ y_c\ z_c]$. Once defined the B matrix identifies the relationship between actuator's forces $f_{(1\times N)} = [f_0\ \dots\ f_N]$ and the generalized thrust vector τ as:

$$f = B^{-1}\tau \tag{6.13}$$

where B^{-1} represent the inverse of matrix, calculated, for instance, using the Moore-Penrose pseudoinverse for a generic *m-by-n* configuration matrix [115]. For the Nessie AUV this B matrix is defined considering the six actuators as:

$$B = \begin{bmatrix} 1.0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 1.0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.625 & -0.435 \\ 0.185 & -0.185 & -0.730 & 0.530 & 0 & 0 \end{bmatrix} \tag{6.14}$$

In this case redundancy is used for controlling the *yaw* degree of freedom. In this configuration *forward* and *lateral* thrusters are used together to improve the direction keeping capabilities of the vehicle used, for instance, when collecting data using the on-board imaging sensors. Such operations are conducted at low speed or when the vehicle is holding its position in the mission's environment. For general navigation the allocation is reconfigured on-the-fly preventing the use of *lateral* thrusters for yaw control and relying only on the differential actuation of the *forward* ones.

6.5 Motion Model

Another relevant aspect of the Nessie AUV platform is the availability of a motion model that has been characterised during previous experimental campaigns. This type of model is usually derived from motion's equations [69] and its coefficients are obtained through dedicated identification procedures. For a generic underwater vehicle the motion model is often expressed as:

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau \tag{6.15}$$

$$\dot{\eta} = J(\eta)\nu \tag{6.16}$$

$$M = M_{RB} + M_A \tag{6.17}$$

$$C = C_{RB} + C_A \tag{6.18}$$

where M is the inertia matrix of the underwater vehicle, including rigid-body M_{RB} and added mass M_A effects, C is the matrix of centrifugal and Coriolis terms, again both for rigid-body C_{RB} and added mass C_A effects, D is the matrix of damping terms. The term g represents the vector of restoring forces (gravity and buoyancy). The term ν is the vector of linear and angular velocities in vehicle's body-fixed frame (i.e. $\nu_{(1\times6)} = [u\ v\ w\ p\ q\ r]$). The generalized force vector τ represents, instead, the forces acting on the underwater vehicle with respect to the vehicle's body-fixed frame (i.e. $\tau_{(1\times6)} = [x\ y\ z\ k\ m\ n]$).

Precise knowledge of such a model allows designers to improve the underlying vehicle's control architecture by integrating an inverse dynamic term. On the other hand, the motion model also gives information about drag forces that affect the vehicle's performance during navigation. In this vehicle those are modelled using a damping matrix D formulated with linear (X_u, Y_v, ...) and quadratic (x_{uu}, y_{vv}, ...) drag coefficients:

$$D = -\begin{bmatrix} X_u + x_{uu}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v + y_{vv}|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_w + z_{ww}|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & K_p + k_{pp}|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & M_q + m_{qq}|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r + n_{rr}|r| \end{bmatrix} \tag{6.19}$$

Linear terms are calculated using geometrical approximations. Quadratic ones, on the other hand, are estimated experimentally and reported in Table 6.8. The last set of coefficients represent a major contribution to the drag effects acting during vehicle's navigation. Such a modelling approach, widely used in literature, represents a rough approximation [69], [134] of real hydrodynamic damping terms that affect AUVs. More accurate methodologies are also found in research, some of those involve the use of machine learning approaches to improve the quality of motion models and state estimation techniques. An example is the work developed by Fagogenis *et al.* [110] where a correction

Parameter	Value
x_{uu}	31.808
y_{vv}	222.896
z_{ww}	263.422
m_{qq}	40.526
n_{rr}	40.526

Table 6.8: Quadratic drag coefficients for Nessie AUV.

term, representing non-linear behaviours and derived with a LWPR model, is added to the motion model found in classical approaches.

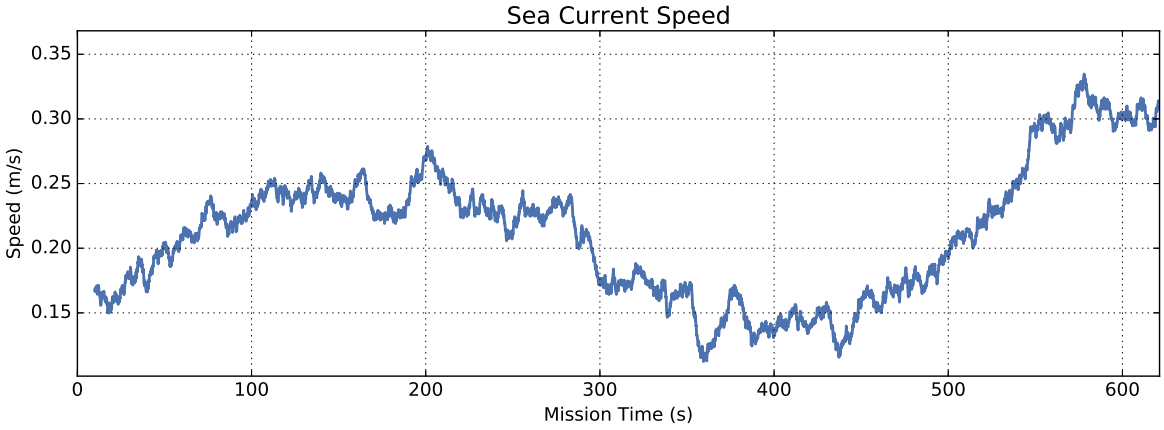


Figure 6.17: Example of simulated sea current with an upper speed limit of about 0.35 m/s.

6.5.1 Sea Current Model

Together with the motion model of the vehicle this work employs a sea current model [69] that is used for simulation purposes. The model is defined as first-order Gauss-Markov (GM) process:

$$\dot{v}_c(t) + \mu_{v_0} v_c(t) = N(0, \sigma^2_{v_c}) \quad \text{with} \quad \mu_{v_0} \geq 0 \tag{6.20}$$

where v_c is the sea current's speed, μ_{v_0} and $\sigma^2_{v_c}$ are parameters that adjust respectively the dependency from the previous state and the variance of a Gaussian sequence. Those characterise the behaviour of simulated sea current. Figure 6.17 shows an example for a moderate sea current generated with the GM model and used for simulation experiments. A similar model is used for sea current's direction:

$$\dot{\psi}_c(t) + \mu_{\psi_0} \psi_c(t) = N(0, \sigma^2_{\psi_c}) \quad \text{with} \quad \mu_{\psi_0} \geq 0 \tag{6.21}$$

where ψ_c is direction, μ_{ψ_0} and $\sigma^2_{\psi_c}$ define the mean and variance of the Gaussian sequence. Altogether random variables $v_c(t)$ and $\psi_c(t)$ describe the behaviour of external disturbances acting on the vehicle during simulated experiments. Parameters $(\mu_{v_0}, \sigma^2_{v_c})$ and $(\mu_{\psi_0}, \sigma^2_{\psi_c})$ are thus chosen to introduce slow variations of speed and direction in the

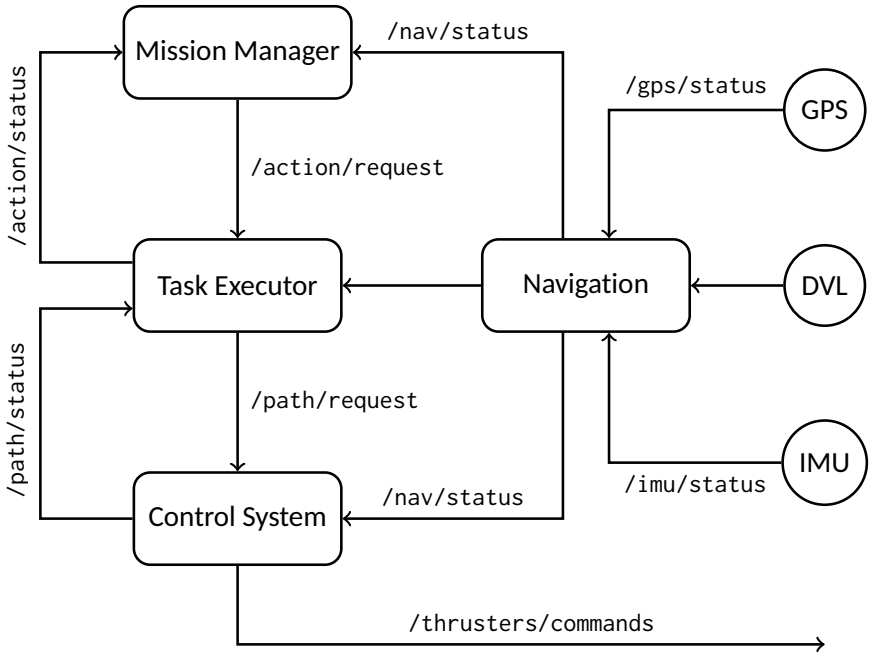


Figure 6.18: Overview of Nessie AUV's software design. Highlighted are the main ROS *nodes* and data *topics* that control the vehicle's functionality for a generic mission.

simulated environments when evaluating the vehicle's performance in the presence of external disturbances.

6.6 Software Architecture

One further relevant aspect for the Nessie AUV platform is the availability of a software architecture build on top of the ROS framework [135]. This is characterised by a few principal elements or *nodes* that deal with vehicle's navigation, control and interfacing with hardware components (e.g. low level drivers). Communication among nodes happens with the exchange of messages on dedicated data communication channels or *topics*. Most of the relevant information is related to navigation, propulsion subsystem and energy monitoring topics that provide base measurements for the proposed architecture at an approximately constant rate of 10 Hz.

An overview of the software nodes composing the Nessie AUV's architecture is shown Figure 6.18. The Navigation node collects measurements from on-board sensors (e.g. DVL, GPS, IMU) and it estimates the vehicle's navigation state sharing it with other relevant nodes using the `/nav/status` topic. Those are the Mission Manager, Task Executor and Control System nodes. The higher level node is tasked with the supervision of mission execution, starting and controlling the development of actions (or tasks) needed to reach the current mission's goal. Information is exchanged with the task executor using the `/action/request` and `/action/status` topics. At intermediate level, this second node monitors the execution of a single task, calculating trajectories, managing the

navigation and controlling the data acquisition processes. Exchange with the control system happens using the `/path/request` and `/path/status` topics. The third node, instead, takes care of platform's control, generating actuator commands (advertised on the `/thrusters/commands` topic and later used by actuators driver) and producing the necessary feedback for the upper nodes.

Two main abstractions are used in such a design: the concept of *action* (or task) and of *path*. The former is used to describe an atomic part of a longer mission and it is characterised by few values (e.g. numerical id, action name, list of parameters) that universally identify a specific task that the vehicle is requested to execute. Action feedback, instead, contains information about outcome (e.g. success or failure) and duration for a given action. Such a formulation is the result of experience collected during the PANDORA project, leading to further studies [136], [137] in interfacing higher level planning systems with the ROS framework as in case of the ROSPlan [137] system. The *path* concept, on the other hand, is used to represent the vehicle's navigation trajectories by specifying a set of intermediate waypoints, an interpolation mode (e.g. linear, quadratic, piecewise) and a locomotion strategy (e.g. smooth, stop and hover, etc.) that fully characterise the final AUV's behaviour. In the case of the Nessie AUV such an abstraction is needed to take into account the different navigation capabilities derived by the specific actuator's configuration. Even for this concept a feedback message is provided, including information about path completion, average speed and duration used to better analyse the actual platform's behaviour.

With all these aspects in mind, the energy-aware architecture is implemented as a collection of additional software nodes that consume and share information with the existing vehicle's architecture and as a small Hardware Abstraction Layer (HAL) that interfaces with the energy monitoring components. The choice of using the ROS framework simplifies the integration with modern vehicle's designs and, at the same time, it allows the distribution of heavy computations, for instance large regression analyses, among different on-board computers such as the ones available on the Nessie AUV.

In this section two relevant implementations are introduced to describe the software elements integrated in the vehicle's architecture for conducting field experiments with the Nessie AUV's platform. Those are related, respectively, with the fault mitigation and the performance estimation frameworks described in previous chapters. The first implementation deals with fault injection procedures, used for simulating the presence of faults in real hardware. The second, instead, deals with monitoring of the vehicle's navigation path monitoring. This is used to extract performance metrics while navigating in the field and calculating runtime estimations used for the improving the navigation.

6.6.1 Fault Injection

A relevant implementation for the Nessie AUV is a dedicated fault injection layer that allows the introduction of actuator failures to the vehicle without reconfiguring low level drivers,

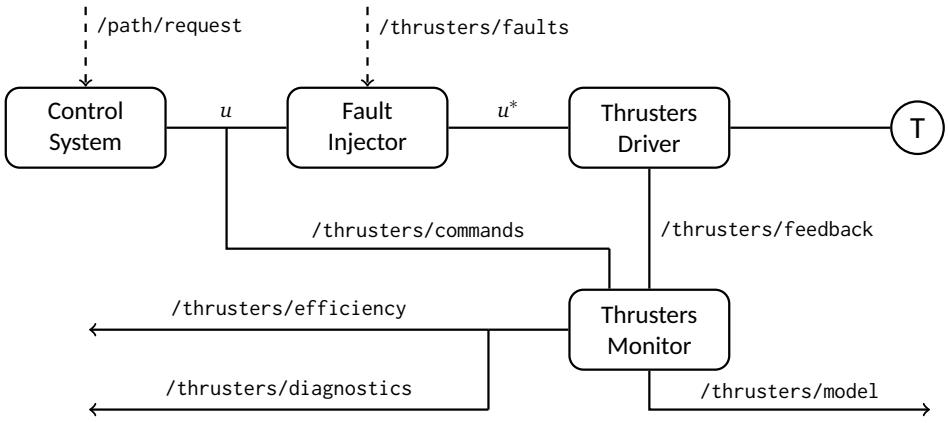


Figure 6.19: Software schema for fault injection in the thruster subsystem. Highlighted are the main ROS *nodes* and data *topics* that control the propulsion subsystem's functionality at runtime.

the control subsystem or, more in general, without changing the platform's behaviour. Introduction of failures is controlled at runtime by remote operators during field testing and it prevents the AUV from operating correctly if no further action is taken. Figure 6.19 shows an overview of the software *nodes* used in the proposed system. The central node is the Fault Injector module. This receives its configuration from the `/thrusters/faults` topic and it manipulates the actuator's commands on the `/thrusters/commands` one by adjusting the equivalent thruster characteristics as detailed in section 4.3. Another node is the Thrusters Monitor. This receives the control input, generated by Control System upon requests from the autonomy software on the `/pilot/request` topic, and the measured thruster's response from the `/thrusters/feedback` topic. At the same it calculates internally the diagnostic metric using the schema introduced in section 4.2 sharing it on the `/thrusters/diagnostics` topic for logging purposes. After calculating the metric the actuator's health status is estimated and published together with the analytical thruster models output on the `/thrusters/efficiency` and `/thrusters/model` topics.

6.6.2 Performance Monitor

Another relevant implementation is the navigation's path monitoring system built around the concept of performance estimation. This calculates at runtime the performance's metrics introduced in Chapter 5 and it provides samples for the non-linear regressions used to characterise the expected vehicle's behaviour in presence of disturbances. An overview of the software nodes for this implementation is shown in Figure 6.20.

The central component is the Path Monitor node. This analyses the vehicle's navigation using the concept of path (i.e. a sequence of waypoints through which the locomotion trajectory is calculated) and of trajectory segmentation as introduced in section 5.2. Paths are advertised by the Task Executor node on the `/path/request` topic. This node handles the execution of all tasks that compose a longer mission. Navigation data is

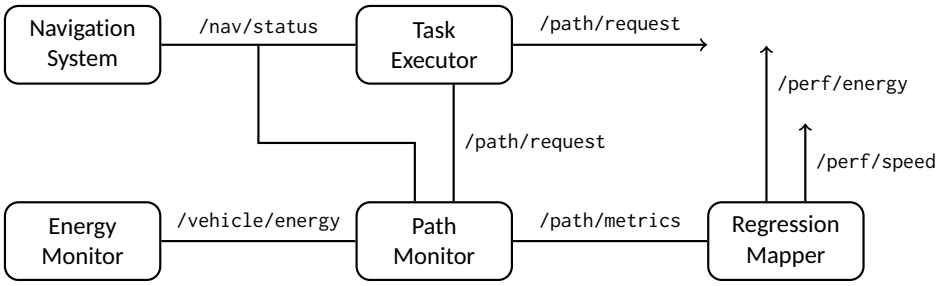


Figure 6.20: Software schema for performance estimation. Highlighted are the main ROS *nodes* and data *topics* that estimate the vehicle's navigation performance at runtime.

provided instead by the Navigation System node on the `/nav/status` topic. Finally, low level measurements are provided by the Energy Monitor node (e.g. acting as HAL for the introduced LEM) using the `/vehicle/energy` topic.

Path messages include a list of waypoints (e.g. geometric points in the 3D world space) and a set of ancillary parameters such as requested navigation speed, interpolation modes, initial and final poses (e.g. geometric points and orientation vectors) that fully characterise the vehicle's behaviour. Using those values the monitor node calculates the performance metrics and it exchanges data with the Regression Mapper node on the `/path/metrics` topics. This last node is responsible for the *data pruning* and *regression* processes, as detailed in section 5.2. Function evaluations happen at a slower rate ($\leq$ 1 Hz) than standard vehicle's data exchange. This is because navigation among waypoints, hence collection of performance samples, happens over a longer time frame, for instance minutes instead of seconds for inspection missions in small areas. After conducting regression analyses the mapper node advertises its estimations on the `/perf/speed` and `/perf/energy` topics, respectively, for the effective *navigation speed* and *energy usage per unit distance* when in presence of external disturbances.

6.7 Summary

In this chapter the experimental platform used in this work has been introduced. An overview of its main features has been presented together with relevant details about the electrical schema, battery and propulsion subsystem. Beside existing components a small low-cost energy monitoring solution has been developed and integrated within the Nessie AUV to augment its capabilities. This required the manufacturing of a standalone board employing integrated Hall effect current sensors. Furthermore, modelling and identification work, done on the vehicle's actuators and on-board batteries has been detailed to better describe the behaviour of those components. Finally, details about the existing software design and the implementation of modules related to the energy-aware architecture has been discussed.

Chapter 7

Experimental Validation

For a successful technology, reality must take precedence over public relations, for nature cannot be fooled.

Richard P. Feynman

After introducing the energy-aware architecture in Chapter 3, the automatic mitigation framework in Chapter 4 and the runtime performance estimation in Chapter 5, experimental validations and operational experience collected with real sea trials are presented. Field operations have been conducted in the context of the EU FP7 PANDORA [1] and EU FP7 ARROWS [2] projects. The mitigation framework is analysed first in section 7.1. Controlled environment experiments and real sea conditions field trials are used to evaluate the effectiveness of the proposed approach when implementing the energy-aware architecture on the hover-capable Nessie AUV. This platform, introduced in the Chapter 6, is employed for inspection missions conducted around human-made structures in shallow water.

After discussing mitigation results the experiments related to runtime performance estimation techniques are presented in section 7.2. Those, beside the Nessie AUV, involve the use of another underwater platform, the IVER3 AUV, a commercial vehicle acquired by the Ocean System Laboratory. This platform is used to conduct archaeological surveys in the context of the EU FP7 ARROWS project. Comparison between the two platforms allows a deeper analysis of the capabilities offered by the proposed runtime estimation procedures. Finally, experiments with route optimisation procedures are addressed in section 7.3. These are conducted with the use of simulated environments where different type of external disturbances are introduced in inspection missions. A simulated Nessie AUV is used for this purpose. The introduced EA-OP and EA-COP algorithms are analysed with respect to an OVRP strategy already in use for the Nessie AUV.

Figure 7.1: Side view of Fort William's mission area. Missions are conducted along both sides of a marine pier with the vehicle going under it when changing inspection side. Combination of high-tides and marine currents on the west side characterise this operating environment.

7.1 Fault Mitigation Experiments

The automatic mitigation framework is evaluated through a series of experiments conducted using the Nessie AUV. First, the operational experience collected during sea trials is reported. In this context the vehicle is deployed in a shallow water environment while in presence of tidal currents. The main goal of such field testing is the performance analysis in presence of external disturbances and longer execution times. Second, a series of detailed tests, including injected failures, are conducted both in a controlled environment and during real sea operations using a hardware-in-the-loop set-up. The vehicle is tasked with an inspection mission, where a set of waypoints are visited to gather sensor readings while respecting user-provided constraints. The controlled environment is used to identify a baseline for the framework's capabilities on-board the experimental platform. Real environment experiments highlight how the effects of a more uncertain scenario impact on the vehicle's survival capabilities and, thus, on the outcome of assigned missions.

7.1.1 Operational Experience

The PANDORA project's sea trials experience gave the opportunity of validating the proposed architecture in the context of a real autonomous mission. The Nessie AUV is assigned several tasks to be executed in a partially known scenario. The experimental site is at The Underwater Centre's facilities located in Fort William, Scotland (N 56°49′25″, W 5°6′30″). There were five days of operations. This environment is characterized by the presence of a marine pier, which can be seen in Figure 7.2a, that extends in the *Loch Linnhe* waters, a sea inlet in the north-western Scotland.

This inlet extends for about 50 km south-west to the *Firth of Lorn* running to the top of *Great Glen Fault*. Despite its relative dimensions this area is characterized by the presence of strong tidal currents with tides low enough to make part of the sea floor around the pier emerge during their lowest period. The operational scenario involves the pier

(a) **(b)**

Figure 7.2: Satellite and detailed view of Fort William's mission area. A pier structure stretches from the shore into *Loch Linnhe* waters. In the satellite view mission (yellow) and restricted (red) areas are marked to highlight the operational constraints.

structure and its surrounding waters giving the vehicle approximately 0.1 km^2 of navigable area. The presence of other activities, mostly related to diving training, ROV testing and small ship traffic, at the operational site required the definition of some restricted zones during the trials where autonomous operations were not allowed. Those can be seen in Figure 7.2a. The mission area is characterized by the presence of high tides which modify the perceived environment disturbances multiple times during a single day. Operations have been carried out at spring tides time (i.e. when the effect of tides is strongest) in order to evaluate the AUV platform when operating in harsh conditions with strong tidal currents but still remaining in shallow waters. Concurrently with the main project's trials some of the proposed architecture components have been evaluated. Dedicated experiments have been conducted during the last 3 days of the original experimental campaign.

Dataset	**Duration** (s)	**Distance** (m)	**Energy** (Wh)	**NRMSE** (%)
day3-1	2040	246	63.74	7.28
day3-2	608	104	23.42	7.49
day3-3	1156	186	28.09	8.99
day3-4	2417	709	143.77	8.44
day4-1	3432	1258	197.46	10.46
day5-1	2440	736	130.03	2.84
day5-2	890	433	47.15	2.77
day5-3	1513	512	67.07	2.91

Table 7.1: Sea trial campaign's results of Nessie AUV. The vehicle conducted several inspection experiments around sunken marine structures in Fort William's waters.

First, an analysis of the proposed *thruster model*, introduced in section 6.4, as an on-board energy estimator has been taken into account. In this case the energy usage of the modelled thrusters E_m has been recorded during field experiments and compared to the real measurements E_r at the end of each mission. Results are shown in Table 7.1

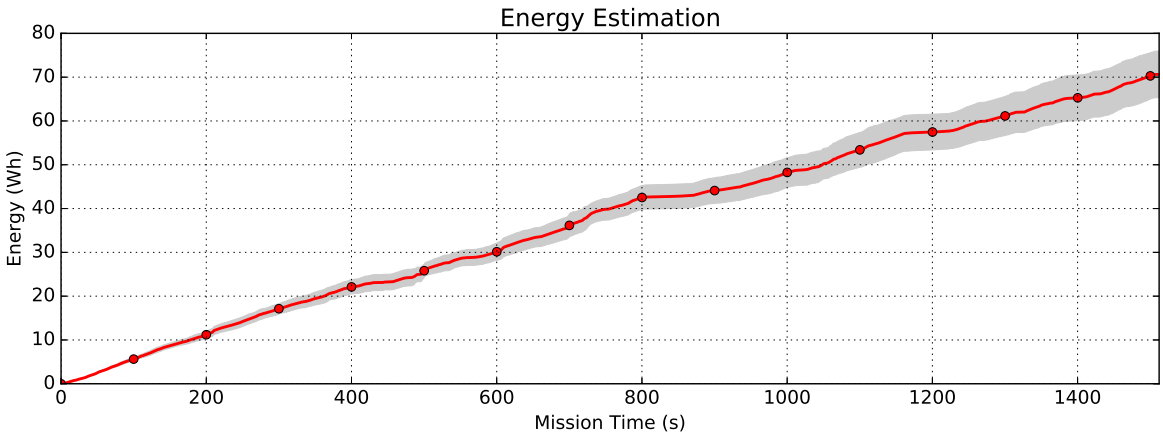


Figure 7.3: Energy usage estimation during a long inspection task (*day4-1*) in real sea conditions while in presence of strong tidal currents. Highlighted are the 95% predictions boundaries from the energy model.

where multiple missions have been conducted on the same day without recovering the vehicle.

Evaluation of model performance is done by calculating the Normalised Mean Root-mean-squared Error (NRMSE) metric for the vehicle operating in real conditions.

$$NRMSE(E_m) = \frac{RMSE(E_m)}{\Delta r_{max}} \tag{7.1}$$

Such a metric is defined as the ratio between Root-mean-squared Error (RMSE) and the maximum value for the short-term energy Δr_{max} as defined in (4.9). The RMSE, instead, is calculated as:

$$RMSE(E_m) = \sqrt{\frac{\sum_k^{N_s}(E_m[k] - E_r[k])^2}{N_s}} \tag{7.2}$$

where E_m values are the model predictions, E_r the real measurements and N_s is the number of samples used to calculate this metric.

Considering the results shown in Table 7.1 and the trend line in Figure 7.3 it can be observed that using the proposed *thruster model* together the low-cost energy monitor, also introduced in section 6.2, as resource estimator allows the energy usage to be tracked with an error of 10% (e.g. in a worst case scenario) without involving more sophisticated navigation models. This performance is achieved when executing missions of up to one hour of duration and more than one kilometre of navigation around marine structures. A relevant case is the *day4-1* mission, later shown in Figure 7.13, where a number of correction manoeuvres (mostly involving vertical navigation due to trimming issues) reduced the final accuracy a bit. Nonetheless, such a preliminary result provides a good confidence level in using the data-driven model in simulated environments for further evaluating the vehicle's expected performance and for improving the planning of outdoor missions.

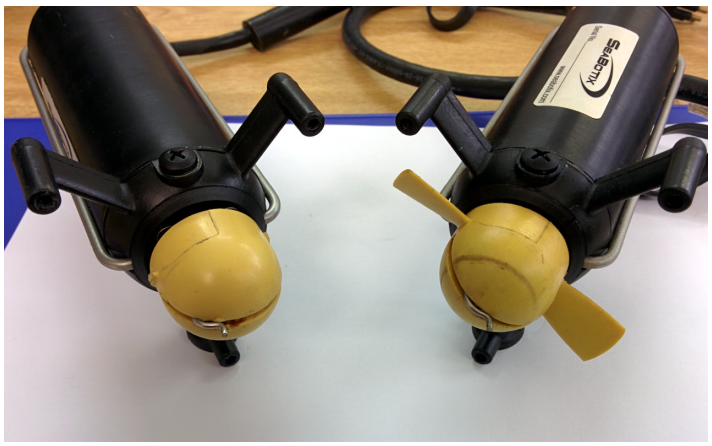


Figure 7.4: Detailed view of Nessie AUV's lateral thrusters. On the left a broken actuator where propeller's blades are missing. Such a condition may have been caused by the ingestion of seaweed inside the thruster's duct leading to the structural failure of the propeller itself.

The overall results, on the other hand, show also that energy usage metrics are good candidates for evaluating the runtime behaviour of the vehicle. They can provide a qualitative indicator of the vehicle's status not only from a diagnostic point of view but also about its performance across a longer time window.

Second, the *fault model* and robustness of the *diagnostic system* have been tested. During the last day of the sea trial campaign Nessie AUV experienced a real failure in one of its thrusters. The fault, discovered at the end of the day, affected the *lateral front* actuator with the loss of the propeller's blades. After initial assessment the fault has been classified as ingestion of sea weed in the thruster's tunnel, causing detachment of blades as shown in Figure 7.4. The *diagnostic system* was able to track the failure for all the remaining part of the daily operations, including during some experiments involving the injection of faults in other actuators. Such an opportunity provided a way of comparing the behaviour of both a real failure and an injected one. For this specific type of fault the damaged component shows a trend similar to an injected fault with a *high degradation* profile ($\eta_T = 0.4$), as detailed in Figure 7.5. Such a behaviour is explained by observing how the current usage pattern for the *broken* thruster deviates from its predicted trend by the *thruster model*, as shown in Figure 7.6. Beside the numerical difference a damaged actuator also presents a highly variable trend possibly caused by the effect of internal speed controller with respect to stall protection procedures.

The failure experienced in the real operations motivates the use of a *tracking algorithm* for the *thruster efficiency* (η_T) parameter within the proposed system. This is because in presence of complex failure scenarios, such as a *broken propeller*, the system is able to rely on the *short-term energy* feature to reduce the estimated efficiency even without introducing a dedicated estimation procedure[1]. In this case the efficiency parameter is adjusted when a mismatch is found between the nominal characteristic and the measured

[1]In an ideal scenario this requires the vehicle to interrupt its current task, execute a specific set of actions and compute an accurate estimation of the *thrust efficiency* (η_T) increasing even more the mission duration. Such a procedure should be trigged, instead, by a planning framework at an higher level.

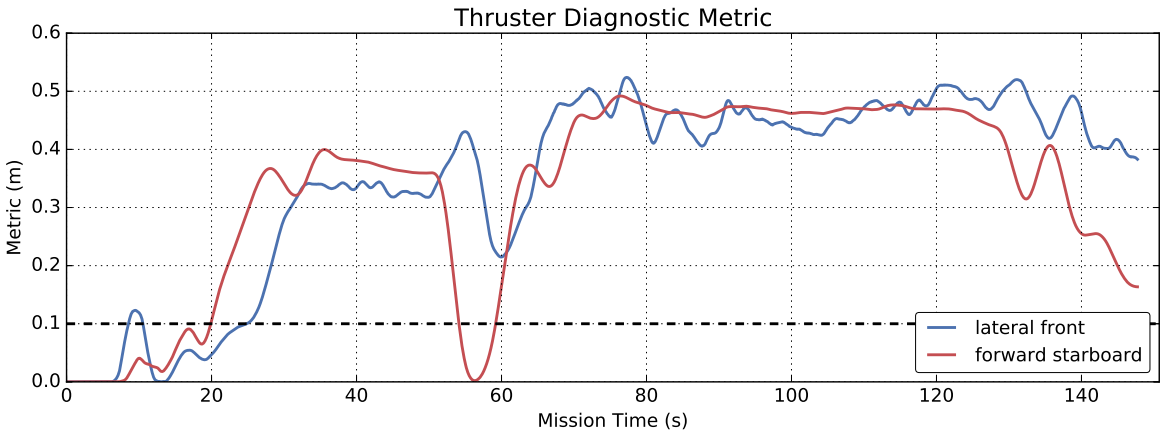


Figure 7.5: Comparison of runtime diagnostic metrics during a navigation task between an injected fault (red) on the *forward starboard* thruster ($\eta_T = 0.4$) and a real failure (blue) in *lateral front* thruster (*broken propeller*).

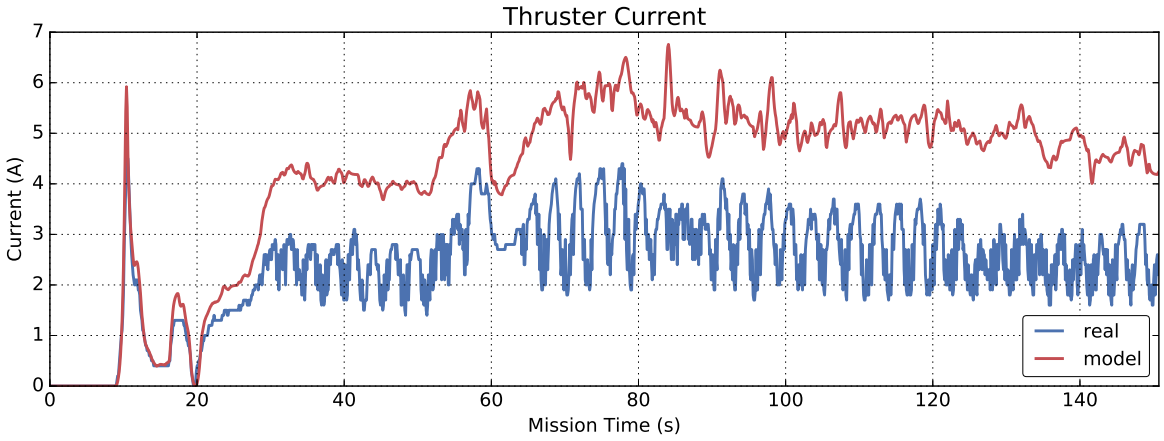


Figure 7.6: Analysis of electrical current's usage pattern for the *lateral front* failed thruster. The plot shows a difference between real measurements (blue) and expected behaviour (red) according to the specific thruster model.

feature or when the lower efficiency threshold is reached removing the actuator from the *thrust allocation* problem. In the latter case the *knowledge base* component is notified about such a removal and its internal belief about the degrees of freedom's availability is updated.

The presence of a real failure is also shown analysing the difference between model estimation and real energy usage for the faulty thruster, as shown in Figure 7.7. In such a case, despite residual uncertainty in the model, the real energy usage falls below the modelled expectation. This is explained by the fact that when a broken actuator is working outside its nominal range this kind of failure is detected even with a qualitative analysis on energy usage trends.

7.1.2 Fault Analysis

Together with the operational experience further testing has been conducted to evaluate the performance of the proposed architecture under the effect of faults. With this in

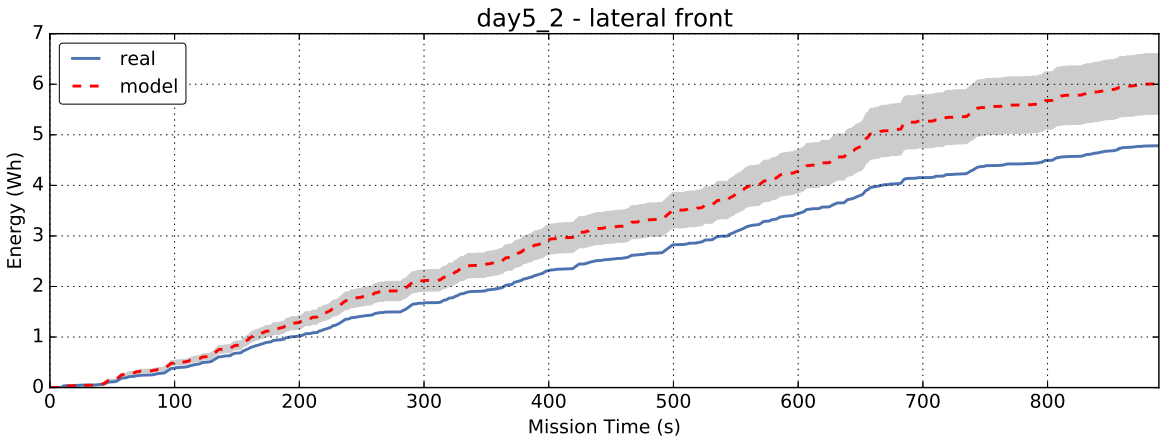


Figure 7.7: Energy usage estimation for a faulty thruster during long inspection task (*day5-2*). In this case the real energy usage falls below what estimated by the model. This confirms that the *lateral front* thruster is not operating correctly.

mind more experiments are conducted in different scenarios: an indoor test tank in steady environmental conditions, and in real sea environment, again, in the context of the PANDORA's project field trials. They benchmark the proposed architecture by injecting different levels of failures. These focus on the *forward* and *lateral* thrusters of

η_T	**Max Thrust** (N)	**Description**
1.0	39.24	*nominal conditions*
0.8	31.39	*low degradation*
0.6	23.54	*medium degradation*
0.4	15.70	*high degradation*
0.2	7.85	*severe degradation*
0.0	0.00	*complete loss*

Table 7.2: Thruster degradation levels used in experiments with Nessie AUV. The thrust values are calculated with respect to their nominal characteristic.

the Nessie AUV. These actuators are used for controlling the *surge*, *sway* and *yaw* degrees of freedom. The vertical plane has not been included in those experiments because of the lack of redundancy in the vehicle's design and the poor operational performance when in presence of a single vertical actuator. Nonetheless, while running such experiments the execution time and the energy usage of each inspection task is measured. This enables the comparison of the effects of the proposed mitigation technique with the normal behaviour of the platform when no failure is present. To better represent the severity of injected faults a simple fault classification, shown in Table 7.2, is introduced. Six different fault levels are thus taken into account.

As described in section 4.3 different degradation modes can be handled with the proposed system. In this first set of experiments the maximum available thrust has been limited according to the injected level of degradation. This assumes a failure mode of type one where a change in the thrust characteristic, similar to the one shown in Figure 4.3 in the case of the *low degradation* fault, is introduced.

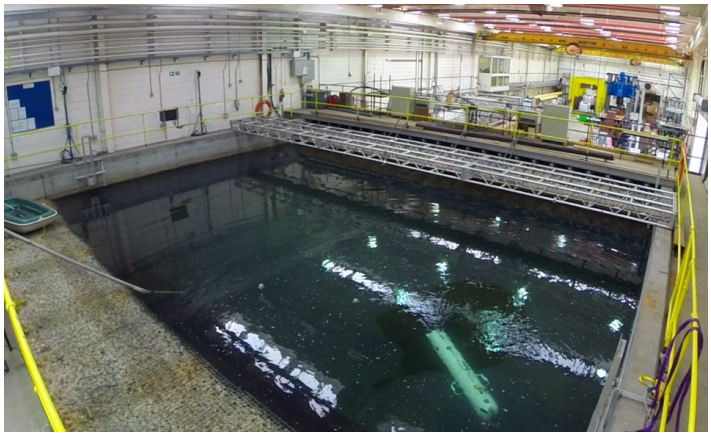
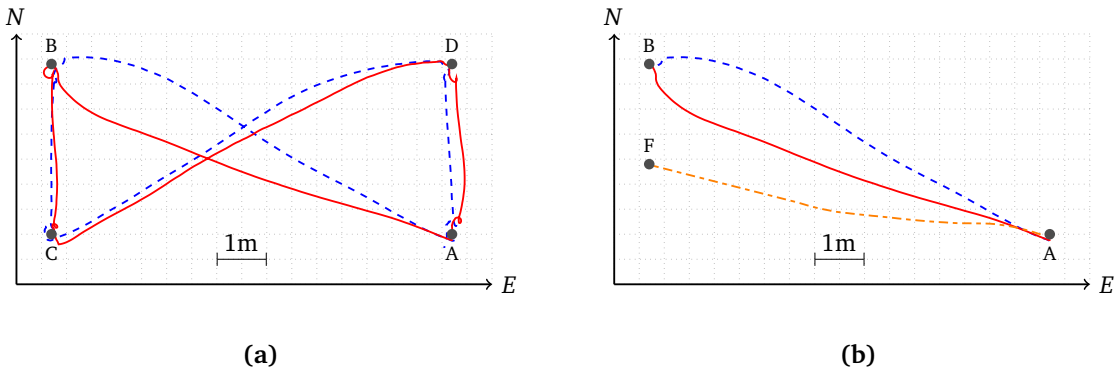


Figure 7.8: Nessie AUV inside an indoor tank before conducting experiments. The vehicle is hovering at fixed depth while executing inspection tasks in presence of injected faults.

7.1.3 Controlled Environment

The first set of experiments takes place in a 12 × 9m indoor tank, shown in Figure 7.8, with steady environmental conditions across different runs. The vehicle is given the task of inspecting the pool's corners looking for the presence of submerged objects. This task is executed several times in presence of the different fault configurations, as described in Table 7.2, introducing, in the first instance, the failure in a *forward* thruster and later in a *lateral* one.

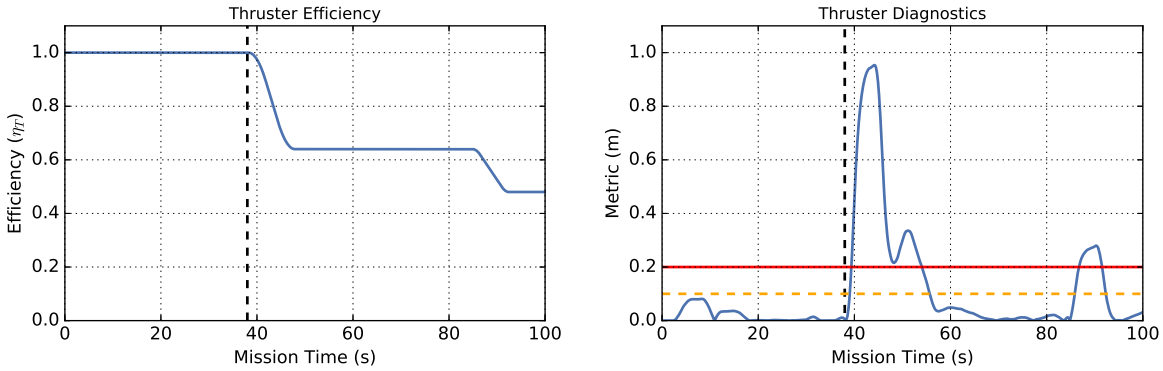
In the case of the Nessie AUV *forward* thrusters are mostly used for the *surge* degree of freedom with a small contribution to the *yaw* only when navigating above manoeuvring speeds while the *lateral* ones are mainly used for the *sway* and *yaw* degrees of freedom especially when navigating at lower speeds, for instance during the hovering phase of an inspection task. Vertical thrusters are not consider in these experiments as the lack of redundancy on the *heave* DOF prevents this vehicle from maintaining efficiently attitude during navigation in the presence of vertical thruster faults.



(a) **(b)**

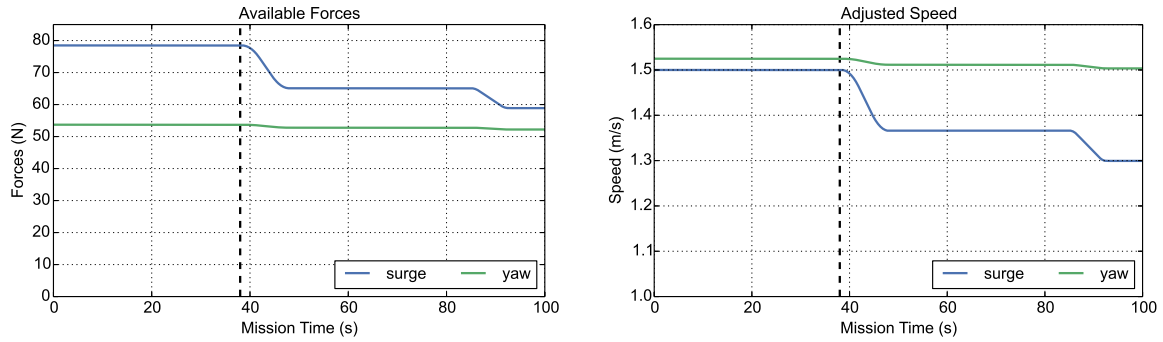
Figure 7.9: Trajectory comparison for controlled environment experiments with *forward thruster degradation* faults ($\eta_T = 0.2$). Dashed blue line shows the navigation path under nominal conditions. Solid red line shows the path in presence of fault condition using the proposed system. Double dashed orange line shows the path taken by the vehicle without mitigation.

With these considerations in mind the effect of thruster degradations is shown, for



(a) **(b)**

Figure 7.10: Experimental results for *high degradation* ($\eta_T = 0.40$) experiment of *port-side forward* thruster. In this context the mitigation framework manages to estimate a thrust efficiency ($\eta_{T_{est}} = 0.46$) close to the injected one.



(a) **(b)**

Figure 7.11: Available thruster's forces and adjusted vehicle's navigation speed during *high degradation* ($\eta_T = 0.40$) fault injection experiment.

a relevant case, in Figure 7.9. In such a case deviations from planned trajectories are explained with the presence of an additional torque generated by the unbalanced thrust allocation among forward actuators. In these charts the vehicle's navigation is reported for both nominal and faulty scenarios to better analyse the effect on the trajectory tracking capabilities of the Nessie AUV. After detecting a mismatch in the faulty actuator along the first trajectory segment $\overline{AB}$ the proposed framework adjusts the underlying control system resuming a standard navigation for subsequent segments $\overline{BC}, \overline{CD}, \overline{DA}$. Navigation, in this type of experiments, is conducted in point-to-point mode with 360-degree rotations around points A, B, C, D needed to acquire more sonar images. The right-hand side of Figure 7.9, on the other hand, shows with more detail the effect of an uncompensated failure when conducting an inspection in the same scenario. Navigation is disrupted and the platform exceeds limits on trajectory tracking tolerances (e.g. more than one metre for lateral separation) resulting in a premature termination of the inspection task.

Figures 7.10 and 7.11 highlight the temporal evolution of calculated metrics showing how the use of a *mitigation framework* allows the vehicle to recover and correct its navigation while still executing its current task. Figure 7.10a shows the estimated thruster

efficiency during the experiment. After injecting the fault at $t = 38.0$ s the *mitigation algorithm* starts adjusting the *weighting coefficients* (w_i) to mitigate the effect of detected degradation. Figure 7.10b shows the calculated diagnostic metric. The two horizontal lines represent the threshold used in the fault detection algorithm. These prevent false alarms from triggering the mitigation algorithm while limiting its responsiveness. Figure 7.11a shows the estimated available forces and Figure 7.11b the speed adjustments enforced at runtime as upper-bound limits for the navigation speeds on selected degrees of freedom.

Numerical results are presented in Table 7.3 and Table 7.4, respectively, for the *forward degradation* and *lateral degradation* scenarios. The tables show how different degradation levels affect the execution of the inspection task. In both cases the vehicle managed to complete the assigned task. Differences, instead, are more evident when in presence of *medium* or *severe* degradations.

η_T	**Duration** (s)	**Energy** (Wh)	**Relative Dur.** (%)	**Relative Ene.** (%)
1.0	126.11	3.60	–	–
0.8	140.99	4.19	11.80	16.30
0.6	148.92	4.62	18.08	28.37
0.4	176.68	5.80	40.09	61.03
0.2	203.77	6.96	61.57	93.11
0.0	204.31	7.02	62.00	94.94

Table 7.3: Experimental results for *port-side forward* thruster degradations. The vehicle completes the inspection tasks with success for all the different levels of degradation.

For the Nessie AUV an increase of 60% and of 100% in the use of *time* and *energy* resources during the execution of a single tasks are considered, respectively, as *warning* and *critical* thresholds. These are used in the *mission execution* module to trigger an adaptation of the current mission. This, for instance, may result in the interruption of current task if the remaining resources are allocated on a more important task (e.g. navigating towards a safe area). At the same time a *replanning* procedure is started taking into account the updated state of the platform.

η_T	**Duration** (s)	**Energy** (Wh)	**Relative Dur.** (%)	**Relative Ene.** (%)
1.0	126.11	3.60	–	–
0.8	137.68	4.16	9.18	15.52
0.6	137.30	4.68	8.88	29.87
0.4	197.36	8.03	56.50	123.14
0.2	203.27	8.49	61.18	135.75
0.0	205.95	8.62	63.31	139.37

Table 7.4: Experimental results for *front lateral* thruster degradations. For $\eta_T > 0.6$ vehicle's efficiency is severely reduced and the use of resources is sensibly increased.

One key aspect that needs to be considered when looking at these results is that the goal of the *fault mitigation algorithm* is to strictly provide an appropriate allocation strategy that enables the vehicle to continue its current task. This, as shown in the tables,

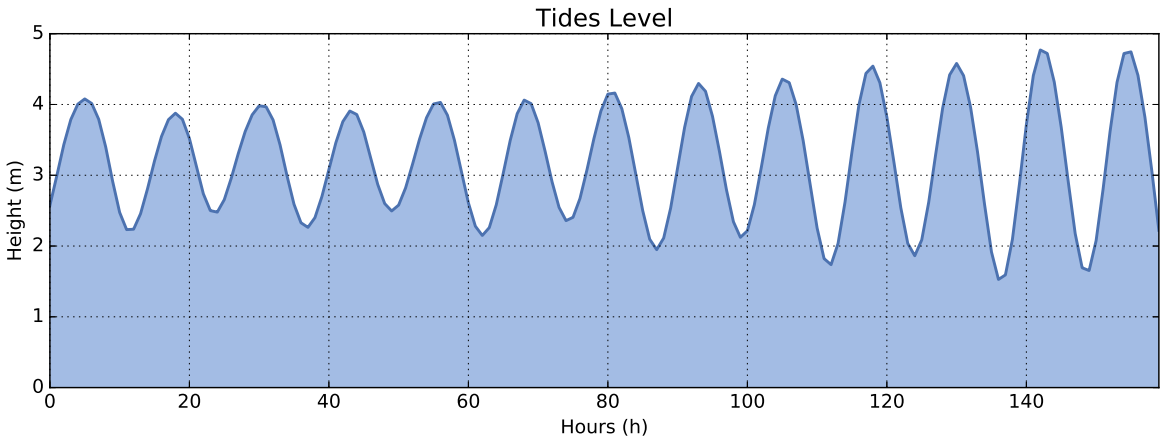


Figure 7.12: Typical weekly tides for the Fort William's mission area. The variation of water level reaches about 3 meters during the highest cycle. Strong currents characterise the shallow water environment where the vehicle is conducting its operations.

is done at the expense of two mission resources, available energy and remaining time. These, while of lesser importance from a pure control point of view, play an important role in supporting the calculations done at the *planning* system level when evaluating the vehicle's effectiveness for the current sequence of tasks. With this in mind an survivability range, defined in terms of an acceptable loss of performance, can be identified for the underwater vehicle used in these tests. The proposed approach allows the platform to survive up to a 60% loss or *high degradation* ($\eta_T = 0.4$) in its *forward* actuators and up to 40% loss or *medium degradation* ($\eta_T = 0.6$) in its *lateral* ones without disrupting the assumptions usually taken at the planning level. This behaviour matches what was initially observed in simulation studies. Furthermore, this suggests that for *severe degradation* faults an additional mitigation strategy is needed to achieve reasonable performance in the context of a long running mission.

7.1.4 Real Sea Environment

Together with the controlled conditions further testing has been conducted with the context of PANDORA project's sea trials. A set of navigation tasks similar to what has been tested in simulated and controlled environments has been assigned to the vehicle operating next to a human-made structure in shallow waters, as shown in Figure 7.2a. In this case additional restrictions have been applied to the mission area and to the vehicle's trajectory in order to control the effects of tidal currents and allow safe operations. The results of these fault analysis experiments are shown in Table 7.5. These show a similar trend with what has been experienced in the controlled environment. Also for this case, the mitigation system has been able to provide assistance in case of *medium-high degradations*. In the case of more *severe* failures the vehicle was not able to keep up with the rising tidal currents, shown in Figure 7.12. After the injection of a *complete loss* failure, the experiments have been suspended due to the vehicle drifting considerably far from the assigned trajectory.

η_T	**Duration** (s)	**Energy** (Wh)	**Relative Dur.** (%)	**Relative Ene.** (%)
1.0	80.96	4.29	–	–
0.8	85.52	4.62	5.63	7.72
0.6	88.01	5.81	8.71	35.54
0.4	134.41	7.32	66.02	70.81
0.2	139.24	8.82	71.99	105.72
0.0	–	–	–	–

Table 7.5: Experimental results for navigation tasks in presence of tidal currents. The case for $\eta_T = 0.0$ resulted in a mission abort to due violation of trajectory tracking constraints.

On the other hand, the real scenario allowed it to conduct a long-range inspection, shown in Figure 7.13, when in presence of injected failures. These experiments highlight the differences between controlled environments and sea operations, where the presence of currents, wind and waves can affect by great amount the vehicle's performance if severe failures are experienced.

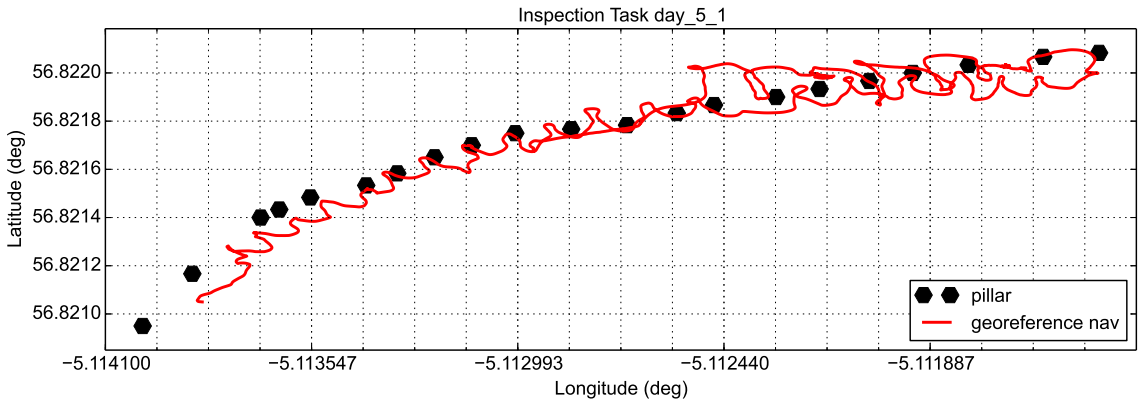


Figure 7.13: Uncompensated navigation data for a long running inspection task. The vehicle uses its GPS unit to acquire the initial position before diving and starting the inspection task. The injection of *low* and *medium* faults does not prevent the vehicle to continue its current task even though its navigation accuracy is affected (as shown in top right part of this chart).

7.1.5 Discussion

The advantage of the proposed approach lies in the use of energy consumption measurements to assess the status of the vehicle and to drive an automatic fault mitigation system. Energy consumption measurements are often already available in existing vehicle's designs for monitoring purposes (i.e. battery management systems, etc.). The choice of analysing such a quantity at runtime helps to reduce complexity of the system design as multiple features can be derived from an homogeneous set of measurements. This allows the proposed architecture to be implemented on several autonomous underwater vehicles without requiring big modifications rather using a small addition of some sensors if an energy metering device is not present.

The experimental results provide an insight on the achievable performance with the proposed architecture. They show that the use of a data-driven approach for extracting the thruster model that allows quick characterisation of the behaviour of such components. This is done with a good accuracy and without the need for dedicated test equipment other than the vehicle itself. Furthermore, the use of the extracted model as an energy usage estimator has been validated in real sea conditions. This offers satisfactory estimation confidence for tasks lasting up to one hour even in presence of tidal currents. These affect the system with strong external disturbances that can reduce the accuracy of a short-term fault detection in presence of *low degradation* faults. Such a behaviour is explained with the use of adjusted detection thresholds that reject an increased disturbance's level without affect the system's capabilities of detecting more severe faults. On the other hand by looking at the long-term energy usage analysis the system is capable of qualitatively detecting deviations from the expected behaviour suggesting the presence of a possible failure even under the effect of strong external disturbances.

The effectiveness of the proposed energy-based diagnostic system is also analysed by comparing its behaviour under the effect of synthetic and real failures. The presence of a broken propeller is shown as a substantial change in the energy usage pattern of the damaged actuator. This, despite the presence of residual uncertainty and external disturbances, is detected and propagated to the output diagnostic metric in a similar way to the synthetic faults. This validates the proposed approach for handling different types of failures under a unified representation for the state of faulty actuators.

The overall performance of the proposed automatic fault mitigation framework has then been analysed. The role of such a system is to provide an automatic response to an unexpected event allowing the autonomous system to continue with its mission. The results, shown in the previous sections, while specific to the underwater platform used during the experiments suggest that by employing this framework the vehicle's survivability is improved. Moreover, the experimental campaign suggests a procedure to evaluate the operational limits of a similar framework. This analyses the changes in energy usage and execution time while in presence of incremental levels of degradation failures. Such an approach can be used with a generic fault mitigation system and therefore characterise its effectiveness when considering a long-term autonomy mission conducted in the presence of unknown environments.

7.2 Runtime Performance Experiments

In this section, instead, experiments with the *runtime performance estimation* framework are presented. These show the use of proposed system on two different autonomous underwater vehicles operating in real conditions. The first is IVER3 AUV [32], a commercial platform used for oceanographic surveys. This vehicle took part in the experimental campaign for the EU FP7 ARROWS project [2]. Sea trials involved archaeological sites

Figure 7.14: IVER3 AUV before deployment during the field trials of the EU FP7 ARROWS project in Trapani, Italy.

in Trapani, Italy and Tallinn, Estonia. The IVER3 AUV is a lightweight torpedo-shaped autonomous vehicle equipped with a side-scan sonar for sea bottom inspection, as shown in Figure 7.14. It has a single large tail thruster together with four control surfaces to maintain its attitude during navigation. It has a 560 W h LiPo battery pack for 8 to 14 hours operations while operating at 2.5 knots cruise speed. The second is Nessie AUV, a prototype vehicle used for inspection surveys in shallow waters, introduced in Chapter 6. This platform was involved in the field trials of the EU FP7 PANDORA project [1] in Fort William, Scotland, UK.

7.2.1 Field Missions

In this analysis two real missions are considered. One is conducted with the IVER3 AUV executing a survey task in an unexplored area. Another conducted is with the Nessie AUV as an initial field navigation test before executing a longer term experiment. In the IVER3 case an ARROWS's project mission is considered. This requires a survey vehicle to conduct an initial sampling of the environment with the goal of detecting underwater objects. After discovering artefacts a different type of vehicle, such as the hover-capable Nessie AUV, may be dispatched towards object locations for further inspections. An example of the IVER3 survey is shown in Figure 7.15, where archaeological artefacts and other human-made objects are detected using side-scan sonar images.

The first mission is shown with more detail in Figure 7.16a. This employs a *lawnmower* navigation pattern to quickly survey a large archaeological site. Figure 7.16b shows the computed values of the performance metric $\varepsilon_{nav}(\psi)$ while the vehicle navigates along its planned trajectory. The sequence of peaks and troughs is explained with the presence of an east-bound sea current that affects the AUV's navigation. Upstream legs require more effort to overcome environmental effects while downstream ones are more efficient.

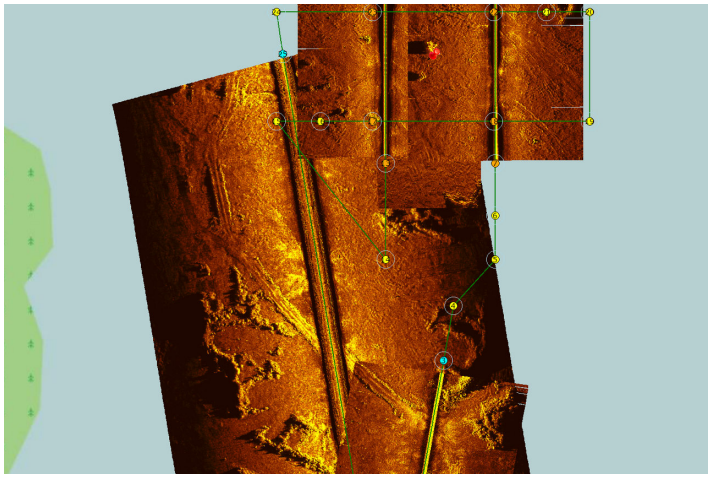


Figure 7.15: Side-scan images acquired by IVER3 AUV during the field trials of the EU FP7 ARROWS project in Rummu, Estonia.

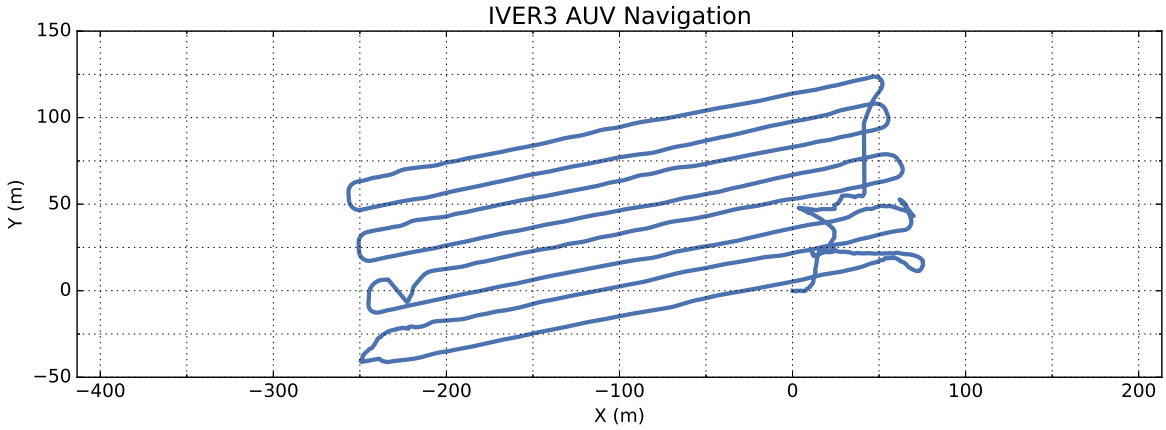
Figure 7.16c shows the distribution of $\varepsilon_{nav}(\psi)$ values for downstream legs with a Gaussian fit overlaid on raw samples. This highlights the effect of uncertainty and residual noise on the calculated metric, spreading measurements away from a mean value and following an approximate Gaussian trend as introduced in the previous sections.

Similar results are obtained with the Nessie AUV both in presence of regular and complex trajectories. A detailed example is shown in Figure 7.17a. The vehicle executes a two leg navigation in coastal waters while in the presence of strong tidal currents. This task is part of a PANDORA's project field mission that employs this AUV for inspection of human-made structures. In this case two different navigation costs are detected as shown in Figure 7.17b. Navigating downstream requires a ε_{nav} of about 220 J/m while the upstream counterpart about 325 J/m. Figure 7.17c shows the distribution of $\varepsilon_{nav}(\psi)$ values for an upstream leg. The overlaid fitting line shows a partial Gaussian trend. This non-ideal behaviour is explained with the presence of higher noise in the energy measurements collected on-board of Nessie AUV. Despite this behaviour the proposed framework provides a satisfactory capability for characterising the operating environment when used in both vehicles.

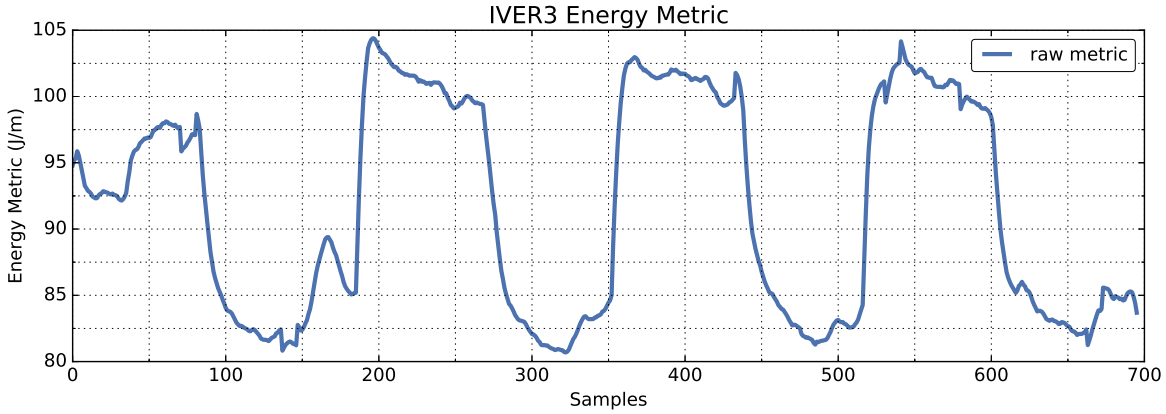
7.2.2 Navigation Patterns

Early experimental results suggest that navigation patterns including few principal directions (e.g. less than eight) are not well suited for obtaining a complete estimation of the quantities that depends on the vehicle's heading. For this reason a few other sampling strategies have been tested in the field to quickly obtain performance estimates. Two examples are shown in Figure 7.18 along with their effects on regression's results.

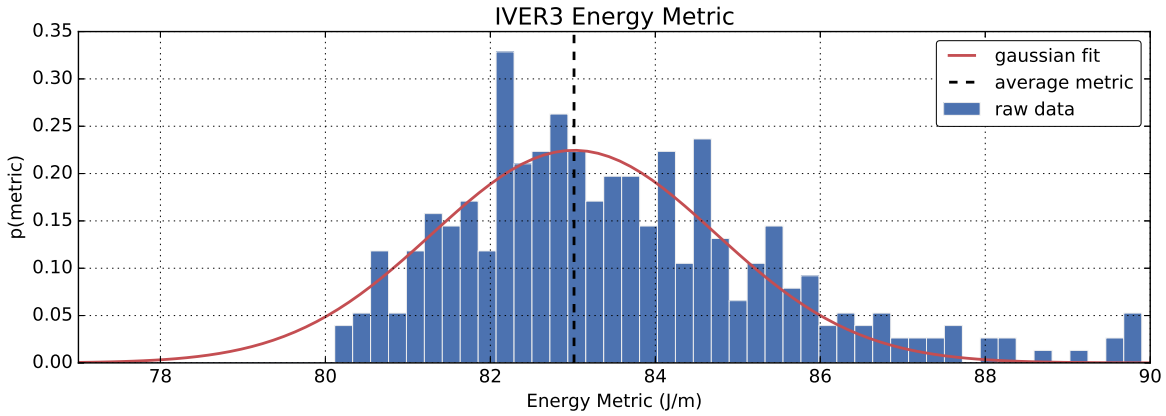
The first is a *radial* pattern executed by the IVER3 vehicle. This trajectory is commonly used with this type of platform when performing a detailed (on the spot) sonar acquisition or for on-board sensor calibration. Figure 7.18a and Figure 7.18c shows the resulting *energy cost for unit distance* estimation for a 10 legs radial pattern. The polar chart shows



(a)

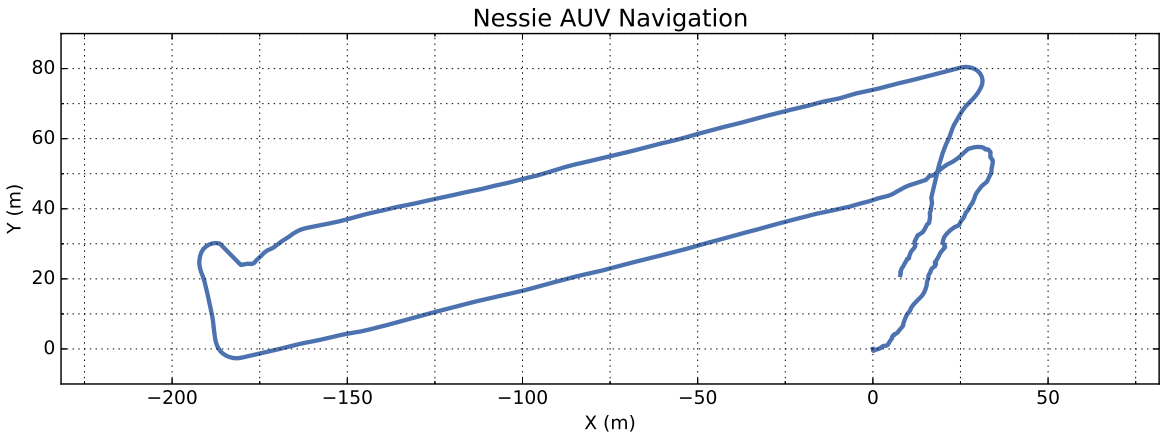


(b)

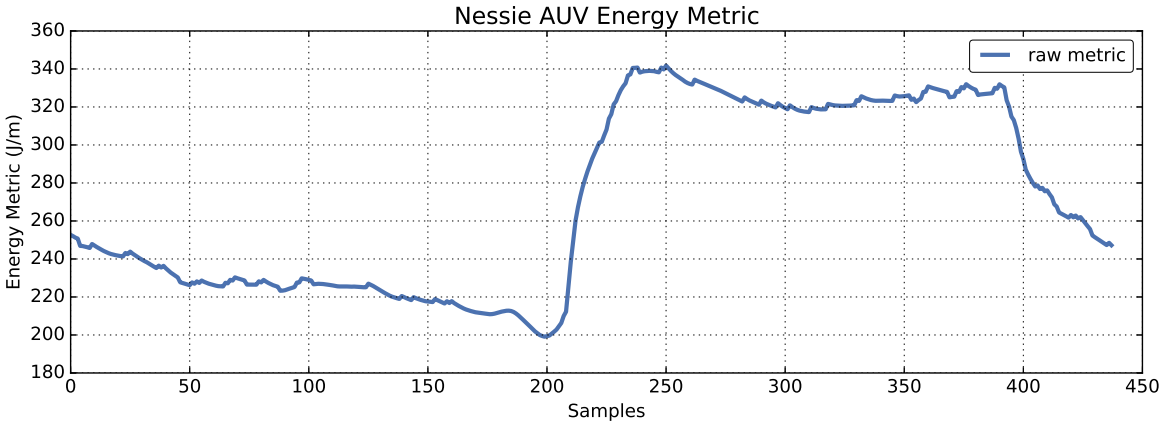


(c)

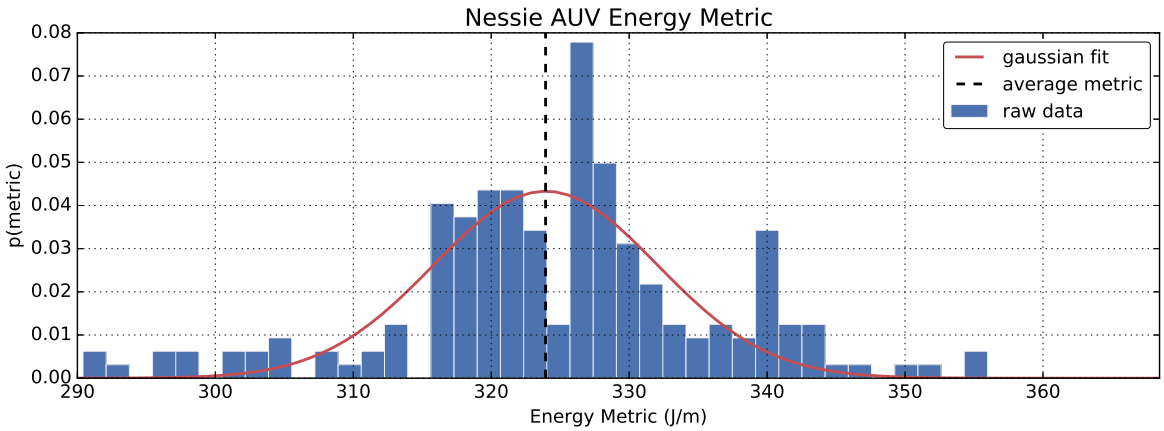
Figure 7.16: IVER3 AUV performance analysis in presence of an east-bound sea current. Chart (a) shows the path followed by the vehicle. Chart (b) shows the raw value for the instantaneous $\varepsilon_{nav}(\psi)$ performance metric over the course of navigation. Sequence of peaks and throats is given by change of heading after each leg. Chart (c) shows the distribution of samples for east-bound legs.



(a)



(b)



(c)

Figure 7.17: Nessie AUV performance analysis in presence of tidal currents. Chart (a) shows the vehicle's forward speed along a two-leg navigation pattern. Chart (b) shows the instantaneous $\varepsilon_{nav}(\psi)$ value over the course of navigation and highlights the difference between legs. Chart (c) shows the samples distribution for the higher cost leg.

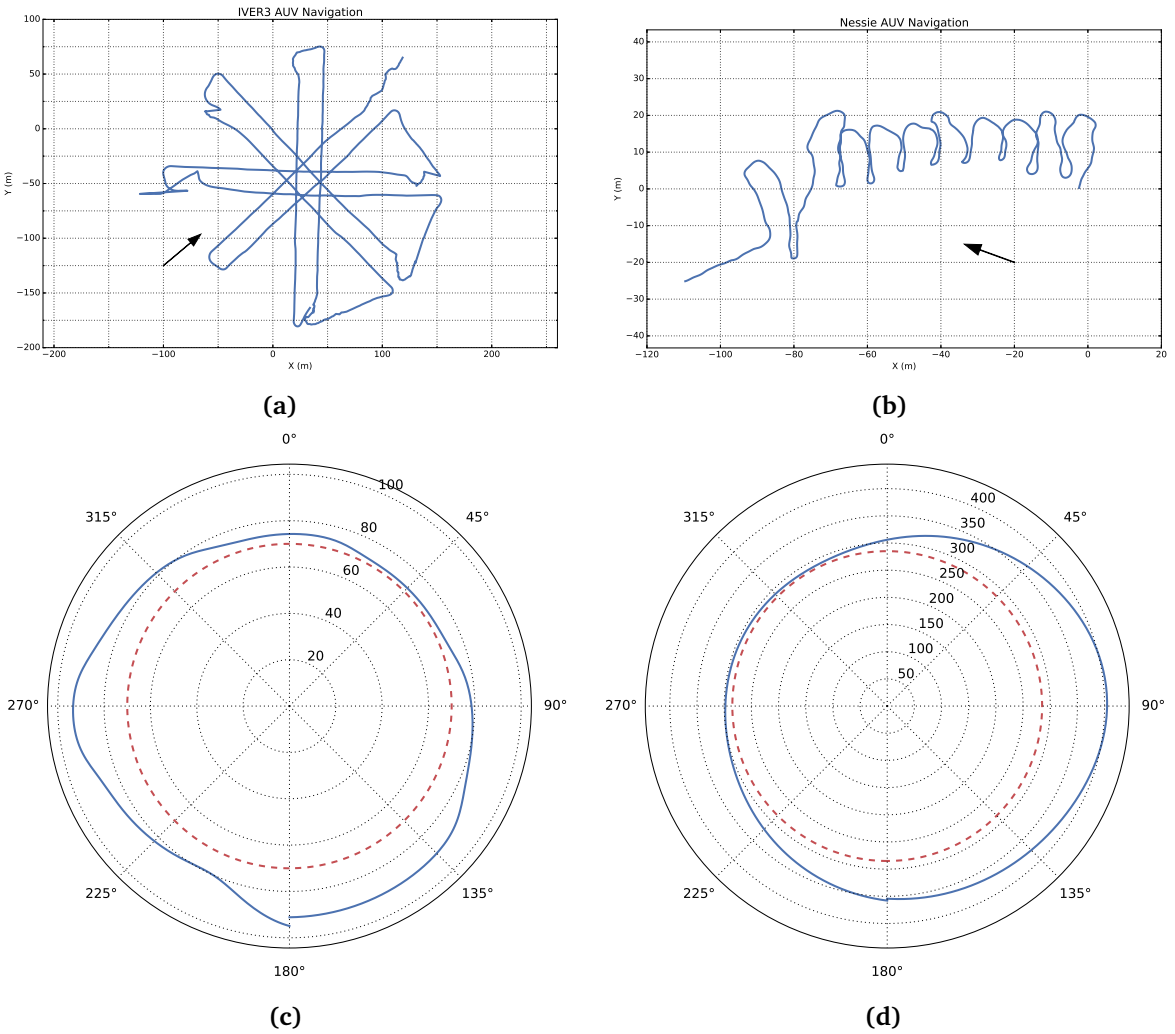


Figure 7.18: Analysis of field performance estimation experiments. Charts (a) and (b) show an example of inspection missions for the IVER3 vehicle and the Nessie AUV operating in real environments. In both the forecasted sea current is shown by an arrow. Polar charts (c) and (d) show the estimated *energy cost for unit of distance* $\varepsilon_{nav}(\psi)$ (J/m) for each vehicle. Dashed lines represents the initial belief at planning stage while solid lines show the estimated cost using runtime measurements during navigation.

how the initial belief (i.e. a uniform navigation cost) is modified by the effect of external disturbances. In the specific case an increase of 20% in energy usage is detected for directions close to the forecasted current. In this example downstream legs require about 80 J/m while upstream ones about 100 J/m. The second is a *smooth* pattern conducted with the Nessie AUV. Figure 7.18b shows a more complex trajectory followed during an inspection around human-made structures. The environment is characterized by the presence of moderate tidal currents. The resulting estimation is shown in Figure 7.18d. This vehicle navigates at lower speed and operates several manoeuvres while in hovering mode. During this experiment energy measurements are heavily filtered by conditions (5.23), (5.24), (5.25) resulting in a longer sampling time (needed to collect enough samples) to build a detailed estimation of the environment.

7.2.3 Discussion

A relevant aspect of IVER3 AUV and Nessie AUV results is the smoothness of the estimated function $\varepsilon_{nav}(\psi)$. This is explained by several factors. One is the domain partition capability of the underlying regression algorithm that allows the modelling of changes in the non-linear function with an increased number of kernels in the underlying model. Another is the availability of several measurements along the function domain, this allows for a better representation of the full heading range with smoother results. An example of this effect is shown in Figure 5.2 where clusters of samples, related to multiple measurements during a single trajectory leg, allows the characterisation of the underlying function along its $[-\pi, \pi]$ domain. In the case of IVER3 AUV, on the other hand, fewer navigation directions have been explored. This is given by the use of regular patterns (e.g. star, cloverleaf, lawnmower, etc.) when conducting field missions. Under this condition the proposed procedure generates a less smooth estimation of the $\varepsilon_{nav}(\psi)$ function. Despite this limitation the resulting estimation is still considered satisfactory because it can further improved, with respect to other headings, by conducting a small training pattern before starting new missions.

Another relevant aspect is the presence of noise in the measurements. This is better shown for the Nessie AUV case where a reduced sensor accuracy, with respect to the other vehicle, affects the estimation. This is shown, for instance, in Figure 7.18d where the absolute direction (i.e. identified with the absolute maximum of the $\varepsilon_{nav}(\psi)$ function) is estimated with some error. More generally, when taking into account the presence of noise, an increased sampling time is needed to provide a complete estimation of the external environment.

On the other hand, quantitative differences for the $\varepsilon_{nav}(\psi)$ functions of the two vehicles are explained by taking into account their specific designs. The Nessie AUV features multiple actuators allocated for forward and lateral navigation, that, together with the use of hover capabilities, is the source for an increased energy consumption. The IVER3 AUV platform, instead, features a single thruster with actuated fins that control its navigation. When executing comparable navigation tasks the IVER3 platform results in a less energy usage. Despite these differences the proposed framework produces satisfactory results both with the commercial vehicle, sampling data at lower rates ($\approx$ 1 Hz), and with a more complex prototype, sampling at faster rates ($\approx$ 10 Hz), without introducing any significant change in their existing software architectures. This aspect suggests how the use of runtime energy measurements is applicable to a broad range of underwater vehicles with different level of autonomy and with an accuracy proportional to the quality of on-board sensors.

7.3 Route Optimization Experiments

After discussing the runtime estimations, simulation experiments are presented in the context of route optimization procedures. These analyse the performance of energy-aware route planning algorithms, introduced in Chapter 5, when employing the previous energy estimations as metrics to adjust the vehicle's mission while operating in presence of unknown disturbances. The algorithms, EA-OP and EA-COP, are evaluated with respect to a standard OVRP formulation that does not incorporate any energy-derived metric or constraint. The MILP/MIQP solver used for evaluating routing problems is Gurobi [138], a commercial solution optimised for fast calculation often used in real vehicles. All computations are performed on a computer with an Intel Celeron J1900 CPU and 8GB of RAM. This configuration is similar to the payload computers integrated in IVER3 AUV and Nessie AUV. In the following experiments a torpedo-shaped underwater vehicle is taken in account. Its behaviour is described using an AUV motion model formulated according to [69]. Model parameters, taken from Nessie AUV, are derived from early identifications and geometrical calculations, as detailed already in section 6.5.

Simulated experiments are conducted for a typical inspection scenario that PANDORA's or ARROWS's vehicles used in their real life applications. These missions require the vehicle to visit a number of inspection points (IPs) where a potential object of interest can be located. Limits on energy usage and execution time are also defined to emulate operational constraints often experienced in real deployments. Points are randomly distributed in a known area following a grid pattern. These are, for instance, identified during a previous survey (e.g. performing a fast and less accurate scan of the environment) or generated using a randomised inspection algorithm [139] usually employed when small *a priori* knowledge is available.

The goal of these experiments is to evaluate the improvements that energy-aware algorithms can introduce in the execution of missions. First, a brief introduction of initial training procedures is presented. These aims at collecting preliminary environmental data for the runtime estimation framework, improving the internal belief of routing procedures and characterising the initial performance metrics. After dealing with those aspects, a comparison between OVRP and EA-OP is introduced. A pair of simulated inspections are conducted in presence of consistent environmental conditions (e.g. point location, state of currents) and of resource scarcity using the two algorithms. In this context it is assumed that the energy budget given to the vehicle is not enough to complete the inspection of all IPs. Results are calculated measuring the mission *outcome*, the number of successfully inspected points while respecting the requested constraints, while taking into account different configurations for the external disturbances. Finally, a comparison between EA-OP and EA-COP is discussed. This, following a similar approach to the previous simulation, evaluates the two algorithms taking into account another the *utility* metric also known as information gain. This measures the amount of potential knowledge collected

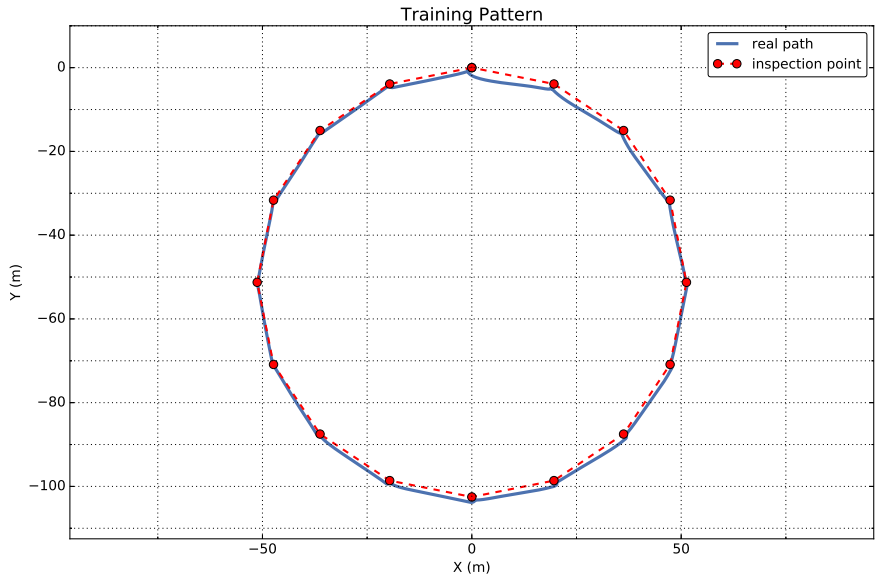


Figure 7.19: Circular training pattern. This is derived from a 16 edges regular polygon and allows the collection of initial measurements for the runtime performance estimation algorithms described in the previous sections. This charts shows the vehicle's behaviour in presence of southbound sea current.

during survey taking into account the sensor model and the spatial relationships between neighbouring inspection points. Both algorithms include energy-aware implementations therefore the EA-COP is employed when clusters of points are expected in the mission scenario.

7.3.1 Training Strategy

Before conducting each experiment the initial performance metrics are derived from the simulated environment using a training procedure. This is devised to be conducted while carrying out initial navigation tests and to assure that transitory effects of the underlying on-line algorithms are not taken into account in the results. This training procedure requires the vehicle to follow a regular pattern after being deployed in the designated mission area. The pattern is generated from a regular convex polygon of $n_b \geq 8$ edges. This assures the collection of measurements along at least n_b principal directions. An example result is shown in Figure 7.19 and 7.20, where a 16 edges polygonal pattern is used in presence of a variable current. This is generated using the model (6.20) with direction $\psi_c = 0$ and an upper speed limit of 0.40 m/s. As shown in the navigation chart strong sea currents affect also the accuracy trajectory following procedures. This is more evident when the vehicle is navigating orthogonally to the current direction. For the rest of this section a similar training procedure is assumed for the simulated vehicle. Environmental knowledge is cleared between experiments to make sure estimation algorithms do not include any residual sample for previous simulations.

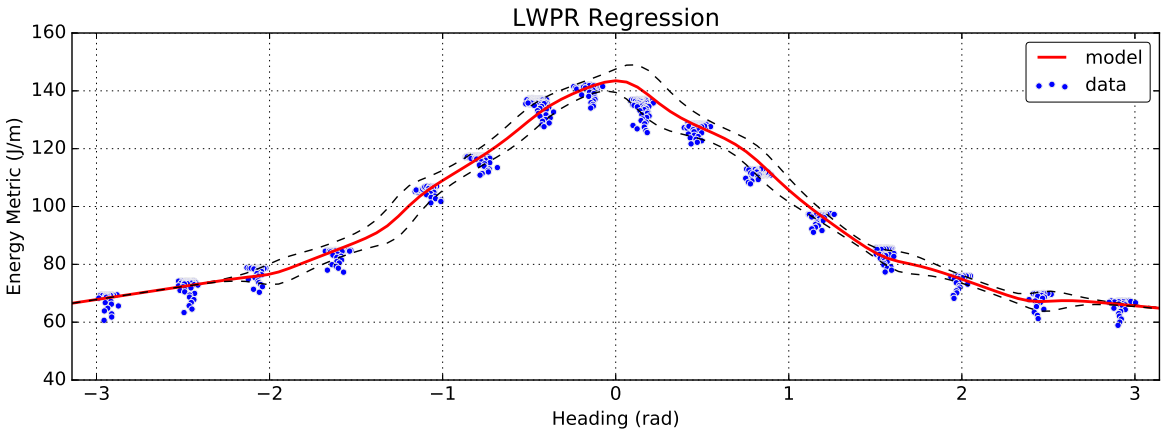


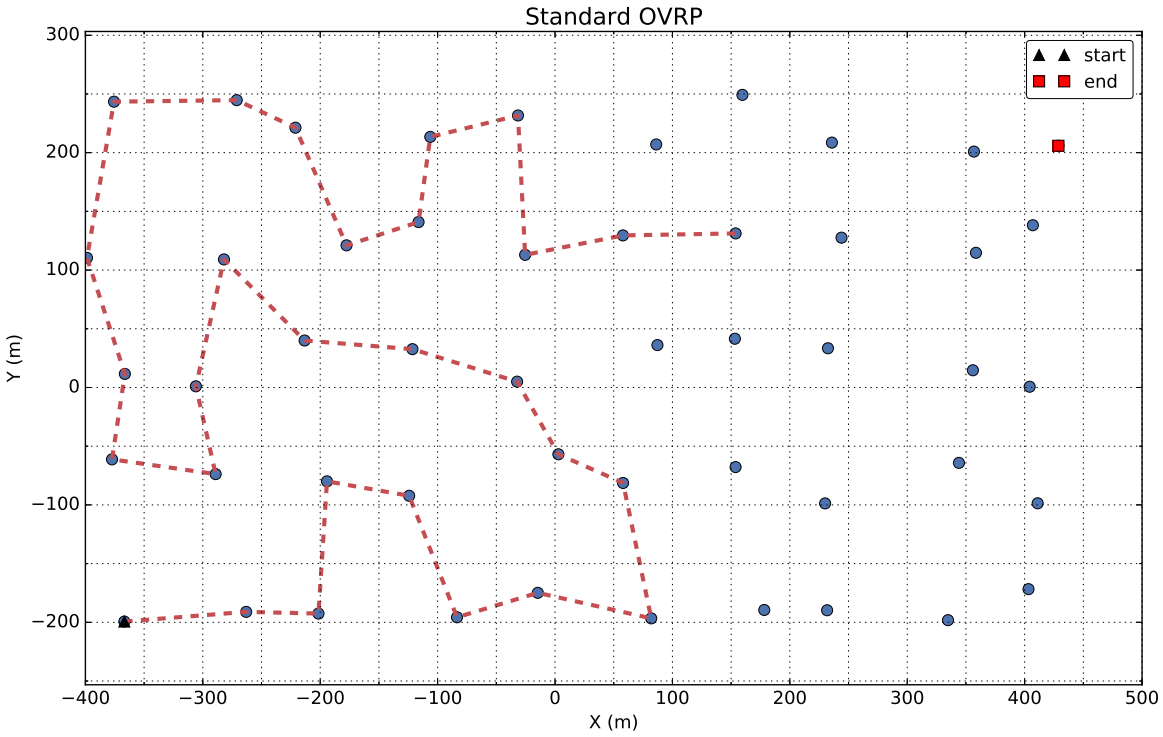
Figure 7.20: Results of the regular polygon training procedure in presence of a varying sea current. Chart shows the estimated performance model (solid line) using the LWPR regression in presence of 16 different clusters of measurements. Dashed lines show the prediction intervals of the resulting model.

7.3.2 Simulated Missions

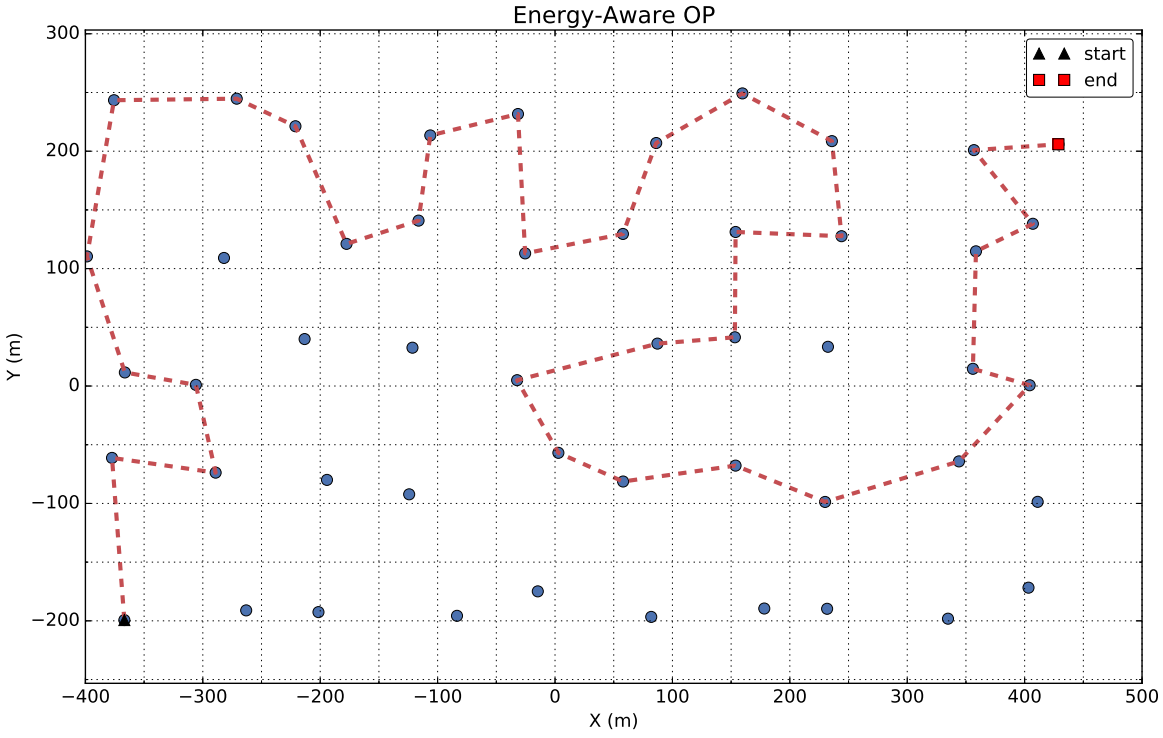
The simulated inspection scenario assumes an underwater vehicle is given a set of randomized inspection points to be visited in an optimal way within specific energy and time constraints. These are chosen to analyse the proposed framework in the presence of resource scarcity, for instance, allocating less energy resources than the amount needed to cover all the inspection points. The environment is characterized by the presence of an unknown varying current with a given upper-bound $v_{c_{max}}$. Each inspection is repeated twice, first using a standard OVRP approach and second using the EA-OP formulation introduced in section 5.3.1.

An example scenario is shown in Figure 7.21, where a 5×10 inspection grid is surveyed in presence of a *moderate* current with an upper-bound of 0.60 m/s. In this experiment the OVRP inspection depletes the allocated energy resources before reaching the designated ending point. This results in the activation of a contingency plan while still inside the mission area. On the other hand, the EA-OP inspection completes in the expected recovery zone (e.g. the top-right corner's point of the map) while discarding intermediate inspections and selecting more favourable navigation legs during navigation. A comparative outcome, in terms of energy usage, is shown in Figure 7.22. In this chart the simulated energy usage is analysed for the two strategies. In the EA-OP case a better use of on-board constrained resources allows the vehicle to visit more points than the classic OVRP strategy while still employing the same amount of allocated energy. This effect is explained by the selection of appropriate navigation legs that are more favourable from a locomotion point of view according the estimated performance metrics.

In order to avoid biased results, given, for instance, by a specific configuration, multiple simulations are conducted varying parameters like the sea current direction ψ_c, the upper-bound of simulated current $v_{c_{max}}$ and the assigned energy budget e_{max}. For the ψ_c term 20 different directions are considered, segmenting the direction domain $[-\pi, \pi]$ in steps of



(a)



(b)

Figure 7.21: Comparison of route planning algorithms for an inspection mission with limited energy resources. Chart (a) shows the route calculated with the standard OVRP formulation. Chart (b) shows instead the proposed EA-OP algorithm in presence of the same simulated environment. The EA-OP visits more IPs improving thus the vehicle's effectiveness in this scenario.

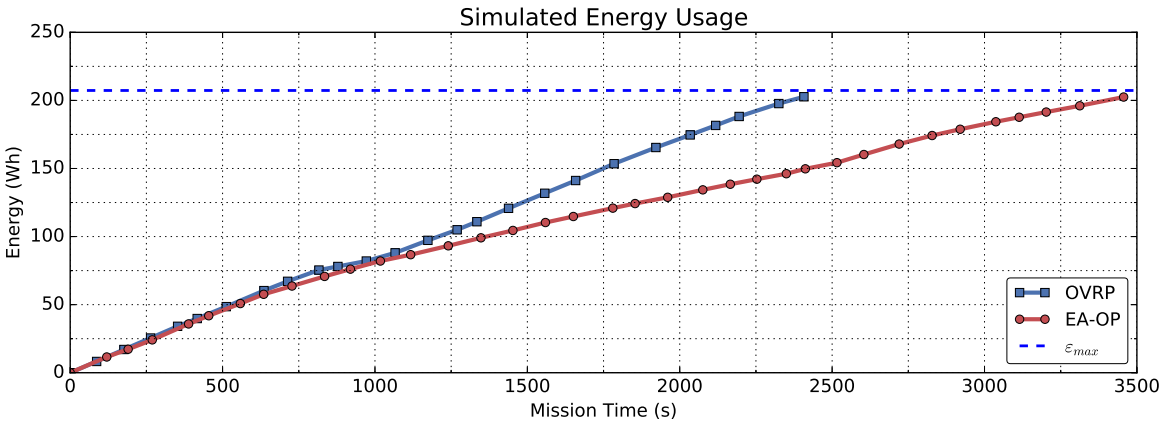


Figure 7.22: Energy usage comparison of simulated inspection missions. Squares represent the OVRP algorithm, circles the EA-OP and diamonds the EA-COP one. The dashed line represent the energy threshold e_{max} calculated for the simulated mission scenario.

18 degrees each. For the $v_{c_{max}}$, instead, 5 configurations are taken into account, varying the maximum allowed current speed from 0.00 m/s to 0.80 m/s, simulating thus each iteration a stronger external disturbance. Regarding the energy budget 4 profiles are considered in the $[100\%, 40\%]$ range. These, beside an initial baseline case, model scenarios where a vehicle starts the inspection mission with less resources available that the one needed to conduct a complete inspection of the mission area. This budget is derived with respect to the OVRP solution as:

$$e_{max} = e_{OVRP} \cdot \gamma_b \tag{7.3}$$

where e_{max} represent the allocated mission energy, γ_b is a scaling coefficient and e_{OVRP} is the estimated energy requirement calculated using the reference OVRP algorithm for the current environmental configuration (i.e. disturbance speed and direction). Finally, 10 variations of the inspection points positions are also taken into account. These all generate a randomized 5×10 inspection grid like the one shown in Figure 7.21. In total 4000 experiments are conducted using the combination of above parameters.

Results are presented in Table 7.6, where, for each sea current $v_{c_{max}}$ and energy budget configuration γ_b, and the mission improvement Δ_{EAOP} are reported along with the average number of inspected points $\bar{n}_{ovrp}$ and $\bar{n}_{eaop}$, respectively, for the two strategies. Individual results are averaged for the different sea current directions ψ_c and grid configurations generated at runtime. From a general point of view, averaged results show how the proposed EA-OP formulation allows for a better mission execution than what was achievable with a standard OVRP approach. Such a behaviour highlights how the introduction of an energy-aware model improves the mission efficiency proportionally with the intensity of external disturbances. Furthermore, it is shown also that the EA-OP strategy maximises the inspection activities when operating with scarce resources, visiting on average more locations than the reference approach while still employing the same energy resources.

After analysing the improvements in terms of mission's *outcome*, another set of experiments is conducted considering the concept of *utility* or information gain. This metric

$\mathbf{v_{c_{max}}}$ (m/s)	$\gamma_\mathbf{b}$	$\mathbf{n_{OVRP}}$	$\mathbf{n_{EAOP}}$	Δ_{EAOP} (%)
0.00	0.4	14.67	16.48	12.34
	0.6	22.52	24.57	9.09
	0.8	29.48	31.62	7.27
	1.0	36.90	38.10	3.23
0.20	0.4	14.24	16.05	12.71
	0.6	21.76	23.90	9.85
	0.8	28.48	30.52	7.19
	1.0	35.52	37.19	4.69
0.40	0.4	14.00	16.43	17.35
	0.6	21.33	24.24	13.62
	0.8	28.19	30.95	9.80
	1.0	35.00	37.33	6.67
0.60	0.4	13.38	16.05	19.93
	0.6	20.05	23.95	19.48
	0.8	25.95	30.76	18.53
	1.0	32.05	36.48	13.82
0.80	0.4	12.55	16.00	27.49
	0.6	18.43	23.90	29.72
	0.8	23.33	30.95	32.65
	1.0	28.38	36.29	27.85

Table 7.6: Averaged results for the simulation experiments. Different energy budgets are employed when conducting the inspection tasks. Values less than 40% are not used as they prevent the vehicle to reach the recovery point using both algorithms.

represents the amount of environmental knowledge that is collected during the simulated mission considering the spatial relationships among neighbouring inspection points. Such an approach takes into account a simulated AUV fitted with a side-scan sonar. The sonar has 200 m sensing range and it is modelled after the device used on IVER3 AUV during ARROWS project's trials. Experiments compare the OVRP, EA-OP and EA-COP formulations in terms of utility collected while conducting simulated inspection missions. The utility metric, defined previously in section 5.3.2, is calculated according (5.58) by characterising the exponential fading model with a distance threshold $d_{sens} = 200$m. Having Figure 7.21 in mind, it is shown that few IPs are spaced less than d_{sens} metres apart yet the inclusion of extra constraints in the correlated problem variant aims at improving further the vehicle's effectiveness in the field. The experimental set-up follows what has been described for the previous analysis. Even in this context energy budget limitations are considered and intermediate results are averaged over multiple configurations of the external disturbances ψ_c. Aggregated results are reported in Table 7.7. These show how the use of EA-COP strategy allows the maximisation of the resulting utility (I_{EACOP}) with respect to the ones (I_{OVRP}, I_{EAOP}) obtained employing the other formulations. It is also shown that higher differences in the utility (Δ_{EAOP}, Δ_{EACOP}) are obtained with respect to the classic OVRP approach when introducing lower energy budgets. Such a

$\mathbf{v}_{\mathbf{c}_{\max}}$ (m/s)	$\gamma_{\mathbf{b}}$	I_{OVRP}	I_{EAOP}	Δ_{EAOP} (%)	I_{EACOP}	Δ_{EACOP} (%)
0.00	0.4	14.47	18.22	25.95	19.31	33.48
	0.6	22.56	26.67	18.24	27.70	22.77
	0.8	30.35	33.49	10.32	35.03	15.40
	1.0	36.74	40.32	9.76	41.60	13.24
0.20	0.4	13.99	17.90	27.91	18.84	34.65
	0.6	21.82	26.27	20.38	26.85	23.04
	0.8	29.59	32.39	9.47	34.07	15.12
	1.0	35.50	38.50	8.46	40.91	15.24
0.40	0.4	13.75	18.16	32.05	18.83	36.96
	0.6	21.33	25.89	21.37	26.99	26.53
	0.8	29.08	32.23	10.81	34.06	17.12
	1.0	34.99	38.46	9.92	40.28	15.14
0.60	0.4	13.08	17.73	35.59	18.20	39.14
	0.6	20.04	25.57	27.60	26.49	32.20
	0.8	26.63	31.60	18.67	33.06	24.16
	1.0	32.44	37.07	14.28	38.34	18.18
0.80	0.4	12.11	17.37	43.38	18.07	49.14
	0.6	18.37	25.39	38.22	26.36	43.51
	0.8	23.87	31.49	31.95	32.48	36.06
	1.0	28.65	36.47	27.28	37.47	30.76

Table 7.7: Averaged results for information gathering experiments. These compare EA-OP and EA-COP strategies when considering the amount of environmental knowledge collected during the inspection task.

behaviour is explained considering that the OVRP formulation produces routes that visit a low number of inspection points. In fact, by referring to the expression (5.58), the utility is proportional to the *length* (e.g. number of IPs) of the calculated route and to the presence of neighbouring points around the visited areas. Still referring to the same expression, the collected *reward* (r_i) for each point is assumed equal to 1, making the inspection of each point equally profitable. Detailed results are shown also in Figure 7.23 and Figure 7.24. The former shows how a different route is introduced by the EA-COP and it highlights the IPs covered by sensing areas. The latter, instead, shows the temporal behaviour of simulated mission. In this specific case the EA-COP collects more utility than the EA-OP solution and conducts a shorter mission without reducing the overall outcome.

After discussing the mission's improvements, an analysis of the computation time for the proposed formulations is presented in Table 7.8. In this context different grid sizes are taken into account to evaluate the effort required in solving the formulated problems. These are defined using the same setting as for the previous inspection experiments. Results show how the EA-OP and EA-COP approaches require more time than the standard OVRP to obtain an optimal solution after increasing the dimensionality. This behaviour is explained by the presence of more constraints in the MILP and MIQP formulation for the energy-aware approach.

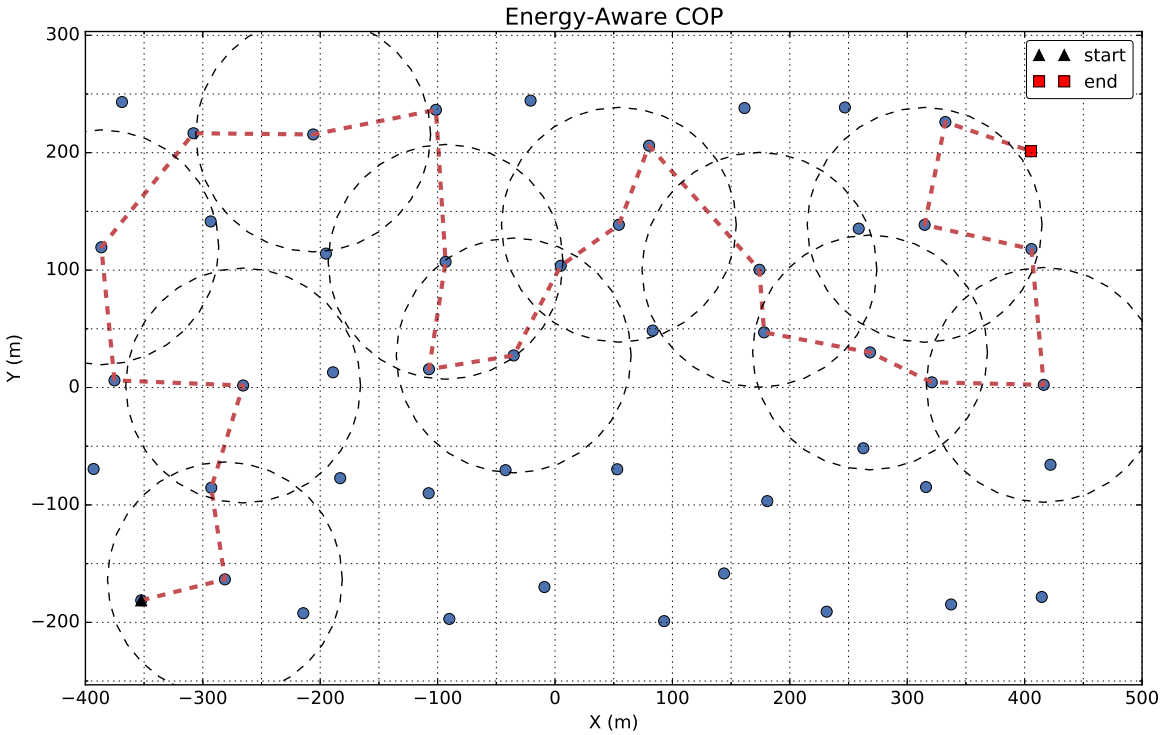


Figure 7.23: EA-COP inspection route. This chart shows the solution calculated with the EA-COP formulation taking into account the spatial relationships among IPs. Circles identify the *useful* sensing zone centred around the visited points. EA-COP solutions gather more environmental information respect to the EA-OP ones.

On the other hand, despite being in presence of NP-hard problems the availability of powerful hardware in modern AUVs makes it suitable to handle moderate sized optimization tasks when operating in the field without relying on external interactions or off-line calculations. Furthermore, it is also shown that the EA-COP requires the maximum effort. This is given by the introduction of quadratic terms in its formulation. Nonetheless, this algorithm provides the best overall results in terms of collected *utility*.

7.3.3 Discussion

This work introduces a runtime energy estimation framework that allows AUVs to derive their navigation performance in presence of external disturbances, such as sea currents. This is done using measurements available on-board and without the support of external

Grid Size	t_{OVRP} (s)	t_{EAOP} (s)	t_{EACOP} (s)
3 × 4	0.04	0.51	0.31
4 × 8	0.12	1.52	1.77
5 × 10	1.94	4.62	5.02
6 × 12	7.18	36.27	47.21

Table 7.8: Averaged execution time for route optimisation experiments using different grid sizes. Solutions are calculated considering a 5% optimality gap (or MIP-gap) and employing a large energy budget. Environmental conditions are kept consistent for all simulated scenarios.

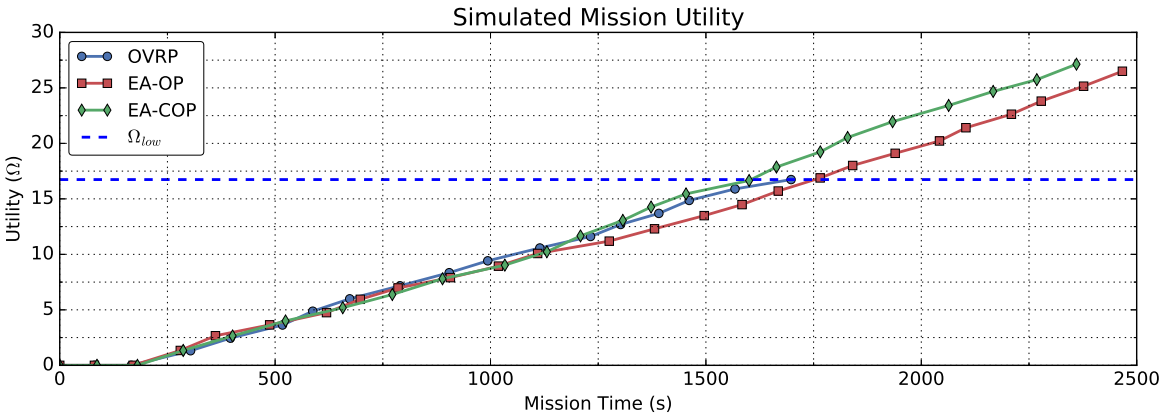


Figure 7.24: Utility comparison of simulated inspection missions. Squares represent the OVRP algorithm, circles the EA-OP and diamonds the EA-COP one. The dashed line represent the lower-bound Ω_{low} for utility collected in the simulated scenario.

sensors. Beside monitoring aspects, the availability of runtime estimations allows the introduction of an energy-aware route optimization procedure that improves the use of on-board resources and the mission's outcome when in presence of resource scarcity. Results show an average improvement from 5% to 20% of the mission's outcome when employing the introduced EA-OP methodology. Experiments are conducted with a simulated vehicle, modelled after the Nessie AUV, operating in the presence of external disturbances and different configurations of sea currents. Simulated scenarios require the vehicle to inspect a unknown area where several inspection points are provided along a randomised grid. Different energy budgets are also taken into account. These simulate vehicle's operations in presence of resource scarcity: when the stored energy does not allow conducting an exhaustive search over the overall mission's area. In those conditions the benefits of the proposed energy-aware strategies are more evident, especially when in presence of moderate disturbances.

After discussing the results of a first comparison, the energy-aware algorithms, introduced in section 5.3, are compared using the concept of mission utility. This measures the environmental information gathered during the simulated inspections taking into account also the spatial relationships among neighbouring inspection points. In this case the introduced EA-COP strategy allows the maximisation of the overall outcome, even if the number of inspections is not the same of the other strategies. Using the EA-COP approach, simulated experiments show an improvement from 15% to 35% with respect to a reference OVRP strategy. The correlated variant also improves the overall outcome with respect to the non-correlated EA-OP approach. Both strategies demonstrate the capability of improving the energy usage, optimising the vehicle's navigation path and enforcing the constraint of terminating the inspections in the user-defined recovery point without depleting the on-board resources.

Beside dealing with mission's results the computational effort for the introduced algorithms has been evaluated on a platform similar to what is available inside the Nessie AUV and IVER3 AUV. The EA-COP, given its quadratic formulation, results as the more

demanding strategy. Nonetheless, for reasonable sized problems (e.g. about 75 inspection points), its application does not require more than one minute of computational time. Given the experimental results the EA-COP and EA-OP represent ideal strategies for adjusting the vehicle's behaviour using the performance estimations conducted with the proposed architecture. Choice of one specific strategy instead of the other should be done according to the available sensors and specific mission requirements.

7.4 Summary

In this chapter the experimental validation of the proposed energy-aware architecture has been presented. This involved the Nessie AUV platform for the analysis of survivability aspects in presence of actuator failures. In this case, the automated fault mitigation framework, introduced in Chapter 4, has been evaluated in both controlled and real sea environments. The operational experience, gathered during the field trials of the PANDORA project, allowed the system to be evaluated in presence of a real failure. Ingestion of seaweed disrupted the correct behaviour of a lateral actuator, leading to the loss of its propeller's blades, and allowing the comparison of the quality of the proposed diagnostic metric with a real event and injected failures. Beside discussion of a real event, the effect of progressively increased degradations has been further analysed with a series of tests involving the forward and lateral actuators of the AUV. This allows comparing the actual platform's behaviour in presence of reduced capabilities and assessing the robustness of mitigation strategies between controlled environment experiments and real life scenarios. In the forward actuator case, a great level of recovery is experienced taking into account the redundancy in the propulsion system. In the lateral one, instead, the effect of external disturbances and unknown conditions reduces the survival capabilities to only medium and low degradation failures. Nonetheless, the proposed framework generally improves the vehicle's effectiveness by offering runtime detection capabilities even when in presence of more critical fault conditions.

After discussing survivability issues the use of a runtime performance estimations framework, introduced in Chapter 5, has been analysed in the context of real sea experiments. In this case, two vehicles, Nessie AUV and IVER3 AUV, are deployed with the proposed framework when conducting PANDORA and ARROWS projects' field experiments. The AUVs were tasked, respectively, with an inspection mission and a more classical survey while estimating at runtime their navigation performance. The operational environments were affected by tidal and sea currents. The introduced system is able to detect differences in their runtime energy consumption according to the relative orientations with respect to external disturbances. Operational results show how the proposed approach allows the characterisation of the vehicle's behaviour in unknown mission's environments when operating on two different platforms.

Finally, other experiments have been discussed in the context of route optimization

problems. These, introduced at the end of Chapter 5, are employed within the proposed architecture to optimize the vehicle's behaviour at runtime when operating in presence of external disturbances. In this case multiple simulation experiments are conducted to evaluate the effect of optimization procedures on the mission outcome. Inspection scenarios, inspired from PANDORA and ARROWS projects' tasks, are simulated when considering different configurations for external disturbances. Algorithms are evaluated with respect to a standard OVRP route optimisation procedure that does not include run-time constraints. Different energy budgets are also introduced in the simulated scenarios. These model the vehicle's capability of operating in presence of resource scarcity.

The proposed EA-OP and EA-COP formulations improve, in most of the scenarios under test, the overall mission's outcome. The first strategy is more suited for achieving the maximum number of inspections. The second, instead, maximises the information gain when introducing a model for the on-board sensor's payload. Both approaches allows the improvement of energy usage, the termination of navigation within user-defined safe areas, without depleting the on-board resources, and, more in general, the increase of vehicle's effectiveness in the field.

Chapter 8

Conclusion and Future Work

The worthwhile problems are the ones you can really solve or help solve, the ones you can really contribute something to. No problem is too small or too trivial if we can really do something about it.

Richard P. Feynman

A novel energy-aware architecture for autonomous underwater vehicles has been built around the concept of runtime energy usage monitoring. This extends existing autonomy software solutions by providing additional capabilities that improve the vehicle's *reliability* and *survivability* when operating away from human supervision. Such capabilities are delivered by fault mitigation and detection modules that assess the availability of mission critical subsystems, such as the battery and propulsion ones, and by a runtime sampling procedure that estimates the effective vehicle's performance when operating in presence of external disturbances, such as sea currents.

A practical implementation of the proposed architecture has been shown for an existing underwater vehicle, the Nessie AUV, used for inspection missions. The results presented in this thesis validate the use of the energy-aware architecture, highlighting the benefits introduced for field operations and showing how vehicle's effectiveness is improved when taking into account failures and harsh environments. In fact, a vehicle using the energy-aware architecture is able to operate in presence of soft-failures or degradations in its actuators without interrupting task execution, to evaluate its navigation performance when operating in disturbed environments and to adjust accordingly its sequence of task for maximising the mission's outcome. Less severe failures are automatically mitigated upon their detection, more serious conditions instead require a proper mission's restructuring done taking into account the vehicle's health status.

The energy-aware architecture addresses the two main questions introduced at the beginning of this work. These aims at evaluating the use of energy consumption as a source of information:

- *Is it possible to assess the internal qualitative state of an underwater vehicle by monitoring the energy usage of its internal components?*
- *Is it possible to estimate the effect of external disturbances on future mission's performance by monitoring the energy consumption of an underwater vehicle?*

In the first case, self-assessment aspects have been investigated for the case of failures in the propulsion subsystem. In the second one, the effect of sea currents have been evaluated together with route optimisation procedures that employ metrics derived at runtime. These procedures aim at improving the vehicle's effectiveness while operating in the field especially when partially known or variable environments are encountered during long-term deployments.

With these objectives in mind the proposed architecture has been practically developed on a hover-capable AUV used for real sea operations. Two main systems have been introduced in its design: a fault mitigation framework and a runtime performance estimation one. These are discussed in Chapter 4 and 5. Frameworks are built around the low-cost energy monitor (LEM): an optional cost-constrained hardware solution designed for underwater vehicles that do not offer detailed energy monitoring capabilities. This has been introduced into the experimental platform and it has been discussed in Chapter 6. Using measurements collected by this component, analytical models of relevant subsystems have been extracted on the experimental platform following a data-driven approach. This strategy relies on few *a priori* parameters and on a training procedure that operators can conduct periodically before deploying the underwater vehicle in the field.

The fault mitigation framework employs the analytical redundancy in a model-based diagnostic subsystem. This allows the detection of anomalies at runtime if changes in the energy usage are discovered comparing the modelled components with sensor measurements. Upon the detection of failures *thrust efficiency* coefficients are re-evaluated according to the last known state of the propulsion subsystem. Estimated values are then used to adjust the lower-level control architecture and its force allocation strategy to provide a first reaction to the detected anomalies. Such an approach allows the vehicle to quickly react to unexpected events without interrupting its current task. In order to prevent the occurrence of thruster saturation an optimisation procedure is further introduced in the control chain. This redistributes the thrust vector among the available actuators, taking into account the *a priori* constraints for these components together with their estimated health status. Beside short-term reactions the diagnostic information, gathered at a lower level, is propagated into a knowledge base. This allows high-level reasoning to be conducted and a better awareness about the vehicle's internal state to be obtained by combining elementary concepts, such as the actuators, with more complex capabilities, such as navigation modes. After reasoning this knowledge is made available to planning modules that can re-adapt the vehicle's mission according to its latest operating status. Results have been presented for experiments conducted in controlled and real sea conditions. In the latter case a real unplanned failure, the loss of a lateral propeller,

validated the diagnostic system while in presence of real disturbances. On the other hand, the use of synthetic faults, injected at runtime, allowed the evaluation of the proposed framework with gradually increasing fault conditions.

The runtime performance estimation framework, on the other hand, employs energy measurements to derive performance metrics using a state-of-the-art non-linear regression algorithm, known as LWPR. This is done without modifying the tasks assigned to vehicles during their sea operations. The choice of this specific algorithm allows the capture of any non-linear behaviour of the external disturbances without requiring a complex or computational-heavy model of the environment. This feature shows its advantages when in presence of residual noise and outliers in the collected samples, for instance, as experienced during real sea operations. Experimental results, collected with the hover-capable AUV and with a commercial underwater vehicle, have shown the framework capability of providing performance estimates when in presence external disturbances. Control of the vehicle against sea currents have been correctly identified as more energy demanding with respect to less disturbed scenarios and the initial vehicle's belief. Furthermore, the computed runtime metrics have been successfully employed to evaluate the feasibility of underwater missions. These have been modelled as a sequence of tasks, each with planned resource usage and execution uncertainty, in the proposed framework. Such a procedure makes use of the energy-aware route optimization problems (EA-OP and EA-COP) also introduced in this work.

The EA-OP is derived from an Orienteering Problem and combines the runtime estimations with additional operational constraints to select an optimal sequence of tasks that maximise the mission's outcome. Results show how this approach improves the vehicle's navigation as more energy-efficient routes are preferred soon after sampling data from the environment. In case of resource scarcity less favourable (and not achievable) intermediate goals are also discarded avoiding resource exhaustion while operating in the field. Simulated missions, conducted in small areas and littoral environments, show overall improvements on the missions' execution proportional to the intensity of external disturbances when comparing the proposed optimisations with a more classic approach, such as an OVRP strategy.

Overall, it has been shown that the introduction of such an architecture provides improvements to the autonomy capabilities of modern underwater platforms. These allow the vehicle to overcome unexpected events, such as degradations of subsystems, to evaluate its effective performance and to adapt its navigation when operating in unknown environments. Such features are supported with the use of metrics derived by the runtime analysis of vehicle's energy consumption.

8.1 Major Findings

In detail the experimental validations and the operational experience collected during sea trials allows the highlighting of the relevant benefits from the proposed architecture.

1. The use of data-driven procedures together with a state-of-the-art non-linear regression algorithm allows system designers to rely on hardware-in-the-loop procedures to train and adjust redundancy models used for diagnostic purposes. The derived models have shown enough accuracy for identifying unexpected behaviours in the propulsion subsystem, when considering the most common types of failures, such as degradations or complete loss of thrust. This feature relieves some of the calibration burden from operators to allow them to focus their attention on more relevant tasks such as a detailed planning of field missions.

2. The use of a mitigation procedure in presence of limited hardware redundancy has been validated on a real AUV. This allowed the vehicle to maintain navigation capabilities in presence of less severe degradation failures. Such a characteristic gives an autonomous platform the opportunity of completing its current task without interrupting a mission upon detection of minor failures. This is relevant for vehicles operating around human-made structures where an emergency stop could position the vehicle in an area difficult to access.

3. The experience with a real fault shows how different failure modes can be covered with the proposed methodology, even if not addressed during the design phases. In fact, providing a reasonable indication of the health status even in presence of unexpected failures allows other software components to become aware that the vehicle's behaviour has changed and full capabilities may be not available as originally planned.

4. The capability of estimating navigation performance while in presence of external disturbances unlocked the possibility to further optimise the vehicle's behaviour. This feature allows a better use of on-board resources while still reusing the infrastructure developed to deliver fault mitigation and diagnostic capabilities.

Given what has been demonstrated in this work the concept of energy-awareness represents a fundamental characteristic for preparing future vehicle designs for the challenges of persistent and long-term autonomy scenarios. Furthermore, while having focused this work around underwater vehicles, the author also believes that such a characteristic is generally relevant for other autonomous robotic platforms, either on the surface or operating above the ground. There is evidence that the combined use of runtime estimations and energy-aware optimizations will enable improved effectiveness of autonomous vehicles operating in partially known environments. Such a combination generally increases the platform's self-awareness when operating in field conditions.

8.2 Future Work

Along with the major findings several aspects have been also identified as possible extensions for the proposed energy-aware architecture. These are related both to the self-observation capabilities of the autonomous platform and to the analysis of external factors affecting the vehicle's effectiveness during field operations. More generally, a deeper integration between those two domains is expected to further improve the vehicle's autonomy capabilities when facing a broad range of unexpected situations.

1. Firstly, an opportunity is seen in the extension of self-assessment capabilities to other relevant internal components of underwater platforms. This, while still relying on a runtime analysis of energy consumption, allows for an even more detailed characterisation of vehicle's behaviour when operating in the field. Energy usage models can, in fact, be derived for remaining sensors or actuators and introduced in the diagnostic modules for improving the built-in fault detection capabilities. These can be further expanded with the inclusion of more sophisticated fault detection algorithms, such as fuzzy decision making methods, hybrid or machine learning approaches, where multiple failures modes can be related to the entire vehicle's hierarchical internal model. Such an extension is envisioned in the fault mitigation framework, improving the methodology that has been introduced in section 4.4 using a more classic approach for a limited range of failures.

2. Secondly, looking at secondary components, such as acoustic modems, imaging sensors and other scientific payloads, that could be further represented in the optimisation procedures introduced in section 5.3. This extension relies on the fact that those types of components could often be switched off selectively if not required for all mission's tasks without disrupting the overall operations. Such an approach, known in the aviation industry as *load shedding*, is expected to introduce an even better use of on-board resources, especially if sensors are used only for short periods of time compared to a long-term deployment. Such an approach could lead to an extension of execution times (e.g. by introducing energy savings) and, at the same time, it could maximise the chances completing a mission if more resources are needed while conducting field operations against strong sea currents.

3. Thirdly, developing the extension of the diagnostic system with prognostic and health-management capabilities. These are already of interest to researchers studying reliability and maintainability aspects, represent a milestone to allow autonomy architecture to become even more self-aware about the platform's health status, predicting, for instance, the remaining useful life (RUL) of a degraded marine thruster or battery system. Such predictions can be further integrated in the proposed architecture and improve the mission planning aspects, for example, preventing

the platform from conducting complex tasks if components may not guarantee safe operation.

4. Fourthly, the introduction of a heuristic in place of the MILP/MIQP solver used for calculating runtime optimisations of the vehicle's behaviour. The heuristic can be introduced using a *genetic algorithm* (GA) formulation for the optimisation problems. These can then further extended to be calculated on a many-core architecture, such as a GP-GPU device, given their intrinsic parallel nature [140]. As shown in section 7.3, the classical solver requires long computation times if the size of the optimisation problem grows above a given dimension. This is related to the amount detail (e.g. number of intermediate navigation points, presence of obstacles, etc.) used in representing the vehicle's mission. On the other hand, relying on the speed-ups offered by the heuristic, optimisations procedures can be executed more frequently while conducting autonomous missions, unlocking even faster reactions to changes of the environments, from one side, and allowing the vehicle to adjust its sequence of tasks even if the mission is characterised by many short elementary actions, such a motion primitives in the case of intervention tasks when close to underwater structures.

8.3 Summary

The work presented in this thesis has both proposed and shown the implementation of a novel energy-aware architecture for autonomous underwater vehicles. This analyses the vehicle's runtime energy consumption and it provides autonomy capabilities that improve *reliability* and *survivability* aspects of unmanned operations, such as robustness to dynamic environmental conditions and resilience in presence of soft-failures or degradations. The results presented have highlighted the performance of this novel approach identifying its contribution to common field operations conducted with inspection-class underwater vehicles. The novel architecture is now in use on the experimental vehicles used for field validation. This work has also identified areas where the energy-aware approach can be extended to gather further benefits. These are, for instance, the inclusion of other relevant subsystem in the energy monitoring procedures, with the aim of improving diagnostic and prognostic capabilities, and the introduction of more powerful optimisation strategies that can support on-the-fly mission evolution as envisioned in *persistent* autonomy scenarios.

Bibliography

[1] D. M. Lane, F. Maurelli, T. Larkworthy, D. Caldwell, J. Salvi, M. Fox, and K. Kyriakopoulos, "PANDORA: Persistent autonomy through learning, adaptation, observation and replanning", *Proceedings of 3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*, pp. 367–372, 2012. doi: 10.3182/20120410-3-PT-4028.00061 (cit. on pp. 1, 4, 28, 38, 83, 105, 118).

[2] B. Allotta, R. Costanzi, A. Ridolfi, C. Colombo, F. Bellavia, M. Fanfani, F. Pazzaglia, O. Salvetti, D. Moroni, M. A. Pascali, *et al.*, "The ARROWS project: adapting and developing robotics technologies for underwater archaeology", *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 194–199, 2015, issn: 2405-8963. doi: 10.1016/j.ifacol.2015.06.032 (cit. on pp. 1, 4, 28, 83, 105, 117).

[3] J. Kalwa, M. Carreiro-Silva, F. Tempera, J. Fontes, R. Santos, M.-C. Fabri, L. Brignone, P. Ridao, A. Birk, T. Glotzbach, M. Caccia, J. Alves, and A. Pascoal, "The MORPH concept and its application in marine research", in *OCEANS - Bergen, 2013 MTS/IEEE*, Jun. 2013, pp. 1–8. doi: 10.1109/OCEANS-Bergen.2013.6607988 (cit. on pp. 1, 4).

[4] N. Miskovic, M. Bibuli, A. Birk, M. Caccia, M. Egi, K. Grammer, A. Marroni, J. Neasham, A. Pascoal, A. Vasilijevic, and Z. Vukic, "Overview of the FP7 project "CADDY – cognitive autonomous diving buddy"", in *OCEANS 2015 - Genova*, May 2015, pp. 1–5. doi: 10.1109/OCEANS-Genova.2015.7271375 (cit. on pp. 1, 4).

[5] M. Seto, *Marine robot autonomy*. Springer, 2012, isbn: 9781461456599. doi: 10.1007/978-1-4614-5659-9 (cit. on p. 1).

[6] F. Zhang, G. Marani, R. Smith, and H. T. Choi, "Future trends in marine robotics [tc spotlight]", *Robotics Automation Magazine, IEEE*, vol. 22, no. 1, pp. 14–122, Mar. 2015, issn: 1070-9932. doi: 10.1109/MRA.2014.2385561 (cit. on p. 1).

[7] C. Insaurralde and D. Lane, "Autonomy-assessment criteria for underwater vehicles", in *Autonomous Underwater Vehicles (AUV), 2012 IEEE/OES*, Sep. 2012, pp. 1–8. doi: 10.1109/AUV.2012.6380746 (cit. on p. 1).

[8] G. Frost and D. Lane, "Evaluation of q-learning for search and inspect missions using underwater vehicles", in *Oceans - St. John's, 2014*, Sep. 2014, pp. 1–6. doi: 10.1109/OCEANS.2014.7003088 (cit. on p. 2).

[9] K. Hamilton, D. M. Lane, K. E. Brown, J. Evans, and N. K. Taylor, "An integrated diagnostic architecture for autonomous underwater vehicles", *Journal of Field Robotics*, vol. 24, no. 6, pp. 497–526, 2007 (cit. on p. 2).

[10] E. Miguelanez, P. Patron, K. Brown, Y. Petillot, and D. Lane, "Semantic knowledge-based framework to improve the situation awareness of autonomous underwater vehicles", *Knowledge and Data Engineering, IEEE Transactions on*, vol. 23, no. 5, pp. 759–773, May 2011, issn: 1041-4347. doi: 10.1109/TKDE.2010.46 (cit. on pp. 2, 7, 23, 30, 39).

[11] G. Papadimitriou, Z. Saigol, and D. Lane, “Enabling fault recovery and adaptation in mine-countermeasures missions using ontologies”, in *OCEANS 2015 - Genova*, May 2015, pp. 1–7. doi: 10.1109/OCEANS-Genova.2015.7271535 (cit. on p. 2).

[12] L. L. Whitcomb, “Underwater robotics: Out of the research laboratory and into the field”, in *Robotics and Automation, 2000. Proceedings. ICRA ’00. IEEE International Conference on*, vol. 1, 2000, pp. 709–716. doi: 10.1109/ROBOT.2000.844135 (cit. on p. 2).

[13] A. Sadrpour, J. Jin, and A. G. Ulsoy, “Mission energy prediction for unmanned ground vehicles using real-time measurements and prior knowledge”, *Journal of Field Robotics*, vol. 30, no. 3, pp. 399–414, 2013, issn: 1556-4967. doi: 10.1002/rob.21453 (cit. on pp. 3, 28–29, 32, 43).

[14] J. R. LeSage and R. G. Longoria, “Characterization of load uncertainty in unstructured terrains and applications to battery remaining run-time prediction”, *Journal of Field Robotics*, vol. 30, no. 3, pp. 472–487, 2013, issn: 1556-4967. doi: 10.1002/rob.21456 (cit. on pp. 3, 28–29).

[15] J. LeSage and R. Longoria, “Mission feasibility assessment for mobile robotic systems operating in stochastic environments”, *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 3, Oct. 2014, issn: 0022-0434. doi: 10.1115/1.4028035 (cit. on pp. 3, 28, 30, 32–33, 43).

[16] R. Eubank, E. Atkins, and D. Macy, “Autonomous guidance and control of the flying fish ocean surveillance platform”, in *AIAA Infotech@ Aerospace Conference*, 2009, pp. 2009–2021. doi: 10.2514/6.2009-2021 (cit. on p. 3).

[17] R. Eubank, “Autonomous flight, fault, and energy management of the flying fish solar-powered seaplane”, PhD thesis, The University of Michigan, 2012 (cit. on pp. 3, 12).

[18] M. Caccia, R. Bono, G. Bruzzone, G. Bruzzone, E. Spirandelli, and G. Veruggio, “Experiences on actuator fault detection, diagnosis and accomodation for rovs”, *International Symposiyum of Unmanned Untethered Sub-mersible Technol*, 2001 (cit. on pp. 3, 21, 23, 37, 51).

[19] R. Dearden and J. Ernits, “Automated fault diagnosis for an autonomous underwater vehicle”, *Oceanic Engineering, IEEE Journal of*, vol. 38, no. 3, pp. 484–499, Jul. 2013, issn: 0364-9059. doi: 10.1109/JOE.2012.2227540 (cit. on pp. 3, 39).

[20] P. Patron, E. Miguelanez, Y. Petillot, D. Lane, and J. Salvi, “Adaptive mission plan diagnosis and repair for fault recovery in autonomous underwater vehicles”, in *OCEANS 2008*, Sep. 2008, pp. 1–9. doi: 10.1109/OCEANS.2008.5151975 (cit. on pp. 7, 30).

[21] C. Eriksen, T. Osse, R. Light, T. Wen, T. Lehman, P. Sabin, J. Ballard, and A. Chiodi, “Seaglider: A long-range autonomous underwater vehicle for oceanographic research”, *Oceanic Engineering, IEEE Journal of*, vol. 26, no. 4, pp. 424–436, Oct. 2001, issn: 0364-9059. doi: 10.1109/48.972073 (cit. on p. 8).

[22] J. Ferguson, “The theseus autonomous underwater vehicle. two successful missions”, in *Underwater Technology, 1998. Proceedings of the 1998 International Symposium on*, Apr. 1998, pp. 109–114. doi: 10.1109/UT.1998.670072 (cit. on p. 8).

[23] M. Caccia, R. Bono, G. Bruzzone, and G. Veruggio, “Unmanned underwater vehicles for scientific applications and robotics research: The romeo project”, *Marine Technology Society Journal*, vol. 34, no. 2, pp. 3–17, 2000. doi: 10.4031/MTSJ.34.2.1 (cit. on p. 8).

[24] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, “Girona 500 auv: From survey to intervention”, *Mechatronics, IEEE/ASME Transactions on*, vol. 17, no. 1, pp. 46–53, Feb. 2012, issn: 1083-4435. doi: 10.1109/TMECH.2011.2174065 (cit. on p. 8).

[25] N. Valeyrie, F. Maurelli, P. Patron, J. Cartwright, B. Davis, and Y. Petillot, “Nessie V Turbo: a new hover and power slide capable torpedo shaped AUV for survey, inspection and intervention”, in *AUVSI North America 2010 Conference*, 2010 (cit. on pp. 8, 83).

[26] M. Sangekar, M. Chitre, and T. Koay, “Hardware architecture for a modular autonomous underwater vehicle STARFISH”, *OCEANS 2008*, pp. 1–8, 2008 (cit. on p. 8).

[27] S. R. Vaswani, C. Raikar, B. Parida, A. Kumar, R. S. Banthiya, and S. Singh, “System Design and Implementation of Autonomous Underwater Vehicle “Matsya””, *AUVSI & ONR’s 15th Robosub Competition Journal Paper*, pp. 1–9, (cit. on p. 8).

[28] P. Walters, N. Fischer, M. Thompson, and E. M. Schwartz, “SubjuGator 2012”, 2012 (cit. on p. 8).

[29] P. Beaudet, M.-A. Duchesne, M. Lachapelle, F. Langlois, and J. St-Jules-Prévost, “Concept and Design of the 2013 SONIA AUV Platform”, pp. 1–10, 2013 (cit. on p. 8).

[30] B. Allen, R. Stokey, T. Austin, N. Forrester, R. Goldsborough, M. Purcell, and C. von Alt, “Remus: A small, low cost auv; system description, field trials and performance results”, in *OCEANS ’97. MTS/IEEE Conference Proceedings*, vol. 2, Oct. 1997, pp. 994–1000. doi: 10.1109/OCEANS.1997.624126 (cit. on p. 8).

[31] R. Stokey, A. Roup, C. von Alt, B. Allen, N. Forrester, T. Austin, R. Goldsborough, M. Purcell, F. Jaffre, G. Packard, and A. Kukulya, “Development of the remus 600 autonomous underwater vehicle”, in *OCEANS, 2005. Proceedings of MTS/IEEE*, vol. 2, Sep. 2005, pp. 1301–1304. doi: 10.1109/OCEANS.2005.1639934 (cit. on p. 8).

[32] J. Crowell, “Design challenges of a next generation small auv”, in *Oceans - San Diego, 2013*, Sep. 2013, pp. 1–5 (cit. on pp. 8, 83, 117).

[33] A. Bradley, M. Feezor, H. Singh, and F. Sorrell, “Power systems for autonomous underwater vehicles”, *Oceanic Engineering, IEEE Journal of*, vol. 26, no. 4, pp. 526–538, Oct. 2001, issn: 0364-9059. doi: 10.1109/48.972089 (cit. on p. 8).

[34] Z. Zeng, L. Lian, K. Sammut, F. He, Y. Tang, and A. Lammas, “A survey on path planning for persistent autonomy of autonomous underwater vehicles”, *Ocean Engineering*, vol. 110, Part A, pp. 303–313, 2015, issn: 0029-8018. doi: 10.1016/j.oceaneng.2015.10.007 (cit. on p. 9).

[35] N. Kularatna, “Rechargeable batteries and their management”, *Instrumentation Measurement Magazine, IEEE*, vol. 14, no. 2, pp. 20–33, Apr. 2011, issn: 1094-6969. doi: 10.1109/MIM.2011.5735252 (cit. on p. 9).

[36] B. Price, J. Richardson, and E. Dietz, “State-of-charge and state-of-health monitoring: Implications for industry, academia, and the consumer”, *2012 IEEE International Conference on Electro/Information Technology*, pp. 1–6, May 2012. doi: 10.1109/EIT.2012.6220713 (cit. on p. 9).

[37] V. H. Johnson, A. A. Pesaran, T. Sack, and S. America, *Temperature-dependent battery models for high-power lithium-ion batteries*. National Renewable Energy Laboratory City of Golden, 2001 (cit. on pp. 9–11, 91).

[38] R. Rao, S. Vrudhula, and D. Rakhmatov, “Battery modeling for energy aware system design”, *Computer*, vol. 36, no. 12, pp. 77–87, Dec. 2003, issn: 0018-9162. doi: 10.1109/MC.2003.1250886 (cit. on pp. 9–10).

[39] M. Chen and G. Rincon-Mora, “Accurate electrical battery model capable of predicting runtime and i-v performance”, *Energy Conversion, IEEE Transactions on*, vol. 21, no. 2, pp. 504–511, Jun. 2006, issn: 0885-8969. doi: 10.1109/TEC.2006.874229 (cit. on pp. 9–10).

[40] K. Goebel, B. Saha, A. Saxena, J. Celaya, and J. Christophersen, “Prognostics in battery health management”, *Instrumentation Measurement Magazine, IEEE*, vol. 11, no. 4, pp. 33–40, Aug. 2008, issn: 1094-6969. doi: 10.1109/MIM.2008.4579269 (cit. on p. 10).

[41] B. Saha, K. Goebel, S. Poll, and J. Christophersen, “Prognostics methods for battery health monitoring using a bayesian framework”, *Instrumentation and Measurement, IEEE Transactions on*, vol. 58, no. 2, pp. 291–296, Feb. 2009, issn: 0018-9456. doi: 10.1109/TIM.2008.2005965 (cit. on p. 10).

[42] A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel, “Metrics for offline evaluation of prognostic performance”, *International Journal of Prognostics and Health Management*, vol. 1, no. 1, pp. 4–23, 2010 (cit. on p. 10).

[43] S. M. Rezvanizaniani, Z. Liu, Y. Chen, and J. Lee, “Review and recent advances in battery health monitoring and prognostics technologies for electric vehicle (ev) safety and mobility”, *Journal of Power Sources*, vol. 256, pp. 110–124, 2014, issn: 0378-7753. doi: 10.1016/j.jpowsour.2014.01.085 (cit. on pp. 11–12).

[44] F. Hafslund, “Design of a low-cost CC-VFC for one-celled Li-Ion batteries”, no. July, 2007. [Online]. Available: http://www.diva-portal.org/smash/record.jsf?pid=diva2:348541 (cit. on pp. 11, 87).

[45] H. He, R. Xiong, X. Zhang, F. Sun, and J. Fan, “State-of-charge estimation of the lithium-ion battery using an adaptive extended kalman filter based on an improved thevenin model”, *Vehicular Technology, IEEE Transactions on*, vol. 60, no. 4, pp. 1461–1469, May 2011, issn: 0018-9545. doi: 10.1109/TVT.2011.2132812 (cit. on pp. 11, 13–14).

[46] S. Ziegler, R. Woodward, H.-C. Iu, and L. Borle, “Current sensing techniques: A review”, *Sensors Journal, IEEE*, vol. 9, no. 4, pp. 354–376, Apr. 2009, issn: 1530-437X. doi: 10.1109/JSEN.2009.2013914 (cit. on p. 14).

[47] C. Xiao, L. Zhao, T. Asada, W. Odendaal, and J. Van Wyk, "An overview of integratable current sensor technologies", in *Industry Applications Conference, 2003. 38th IAS Annual Meeting. Conference Record of the*, vol. 2, 2003, pp. 1251–1258. doi: 10.1109/IAS.2003.1257710 (cit. on pp. 15, 88).

[48] K. Yang and H. Amplifiers, "Advanced Current Sensing Suits High-Rel Systems", *Power Electronics*, no. August, 2005 (cit. on pp. 15, 88).

[49] E. Meissner and G. Richter, "Battery monitoring and electrical energy management: Precondition for future vehicle electric power systems", *Journal of Power Sources*, vol. 116, no. 1 – 2, pp. 79–98, 2003, issn: 0378-7753. doi: 10.1016/S0378-7753(02)00713-9 (cit. on p. 15).

[50] V. De Carolis, D. Lane, and K. Brown, "Low-cost energy measurement and estimation for autonomous underwater vehicles", in *OCEANS 2014 - TAIPEI*, Apr. 2014, pp. 1–5. doi: 10.1109/OCEANS-TAIPEI.2014.6964490 (cit. on pp. 15, 50).

[51] A. J. Sørensen, "A survey of dynamic positioning control systems", *Annual Reviews in Control*, vol. 35, no. 1, pp. 123–136, 2011, issn: 1367-5788. doi: 10.1016/j.arcontrol.2011.03.008 (cit. on p. 15).

[52] A. Healey, S. Rock, S. Cody, D. Miles, and J. Brown, "Toward an improved understanding of thruster dynamics for underwater vehicles", *IEEE Journal of Oceanic Engineering*, pp. 354–361, 1995, issn: 03649059. doi: 10.1109/48.468252 (cit. on p. 15).

[53] L. Whitcomb and D. Yoerger, "Development, comparison, and preliminary experimental validation of nonlinear dynamic thruster models", *Oceanic Engineering, IEEE Journal of*, vol. 24, no. 4, pp. 481–494, Oct. 1999, issn: 0364-9059. doi: 10.1109/48.809270 (cit. on p. 15).

[54] J. Kim, J. Han, W. K. Chung, J. Yuh, and P.-M. Lee, "Accurate and practical thruster modelling for underwater vehicles", in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, Apr. 2005, pp. 175–180. doi: 10.1109/ROBOT.2005.1570115 (cit. on pp. 15, 51).

[55] A. Hanai, S. Choi, G. Marani, and K. Rosa, "Experimental validation of model-based thruster fault detection for underwater vehicles", in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 194–199. doi: 10.1109/ROBOT.2009.5152425 (cit. on pp. 16–17, 20–21).

[56] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part i: Quantitative model-based methods", *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 293–311, 2003, issn: 0098-1354. doi: 10.1016/S0098-1354(02)00160-6 (cit. on p. 18).

[57] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies", *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 313–326, 2003, issn: 0098-1354. doi: 10.1016/S0098-1354(02)00161-8 (cit. on p. 18).

[58] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis: Part iii: Process history based methods", *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 327–346, 2003, issn: 0098-1354. doi: 10.1016/S0098-1354(02)00162-X (cit. on p. 18).

[59] J. Gertler, *Fault detection and diagnosis in engineering systems*. CRC press, 1998 (cit. on p. 18).

[60] R. Isermann, *Fault-diagnosis systems: An introduction from fault detection to fault tolerance*. Springer Berlin Heidelberg, 2005, isbn: 9783540241126 (cit. on pp. 18, 21).

[61] R. Isermann, "Model-based fault-detection and diagnosis – status and applications", *Annual Reviews in Control*, vol. 29, no. 1, pp. 71–85, Jan. 2005, issn: 13675788. doi: 10.1016/j.arcontrol.2004.12.002 (cit. on pp. 19, 55).

[62] I. Hwang, S. Kim, Y. Kim, and C. Seah, "A survey of fault detection, isolation, and reconfiguration methods", *Control Systems Technology, IEEE Transactions on*, vol. 18, no. 3, pp. 636–653, May 2010, issn: 1063-6536. doi: 10.1109/TCST.2009.2026285 (cit. on p. 19).

[63] M. Corradini, A. Monteriu, and G. Orlando, "An actuator failure tolerant control scheme for an underwater remotely operated vehicle", *Control Systems Technology, IEEE Transactions on*, vol. 19, no. 5, pp. 1036–1046, Sep. 2011, issn: 1063-6536. doi: 10.1109/TCST.2010.2060199 (cit. on p. 20).

[64] A. Cristofaro and T. A. Johansen, "Fault tolerant control allocation using unknown input observers", *Automatica*, vol. 50, no. 7, pp. 1891–1897, 2014, issn: 0005-1098. doi: 10.1016/j.automatica.2014.05.007 (cit. on p. 20).

[65] A. Alessandri, M. Caccia, and G. Veruggio, "Fault detection of actuator faults in unmanned underwater vehicles", *Control Engineering Practice*, pp. 357–368, 1999. doi: 10.1016/S0967-0661(98)00169-5 (cit. on p. 20).

[66] G. Antonelli, F. Caccavale, C. Sansone, and L. Villani, "Fault diagnosis for auvs using support vector machines", in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 5, Apr. 2004, pp. 4486–4491. doi: 10.1109/ROBOT.2004.1302424 (cit. on p. 20).

[67] N. Miskovic and M. Barisic, "Fault detection and localization on underwater vehicle propulsion systems using principal component analysis", in *Industrial Electronics, 2005. ISIE 2005. Proceedings of the IEEE International Symposium on*, vol. 4, Jun. 2005, pp. 1721–1728. doi: 10.1109/ISIE.2005.1529192 (cit. on p. 20).

[68] J. Wang, "Fault diagnosis of underwater vehicle with FNN", in *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*, Jul. 2012, pp. 2931–2934. doi: 10.1109/WCICA.2012.6358371 (cit. on p. 20).

[69] T. Fossen, *Guidance and control of ocean vehicles*. Chichester New York: Wiley, 1994, isbn: 978-0471941132 (cit. on pp. 21, 57, 61, 99–100, 124).

[70] L. Tang, B. Zhang, J. DeCastro, and E. Hettler, “An integrated health and contingency management case study on an autonomous ground robot”, in *Control and Automation (ICCA), 2011 9th IEEE International Conference on*, Dec. 2011, pp. 584–589. doi: 10.1109/ICCA.2011.6137995 (cit. on p. 22).

[71] J. Ge, M. Roemer, and G. Vachtsevanos, “An automated contingency management simulation environment for integrated health management and control”, in *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, vol. 6, Mar. 2004, pp. 3725–3732. doi: 10.1109/AERO.2004.1368190 (cit. on p. 22).

[72] K. Hamilton, D. Lane, N. Taylor, and K. Brown, “Fault diagnosis on autonomous robotic vehicles with recovery: An integrated heterogeneous knowledge approach”, in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4, 2001, pp. 3232–3237. doi: 10.1109/ROBOT.2001.933116 (cit. on pp. 22, 39).

[73] J. Ernits, R. Dearden, and M. Pebody, “Automatic fault detection and execution monitoring for auv missions”, in *Autonomous Underwater Vehicles (AUV), 2010 IEEE/OES*, Sep. 2010, pp. 1–10. doi: 10.1109/AUV.2010.5779660 (cit. on pp. 23, 51).

[74] M. Brito and G. Griffiths, “A markov chain state transition approach to establishing critical phases for auv reliability”, *Oceanic Engineering, IEEE Journal of*, vol. 36, no. 1, pp. 139–149, Jan. 2011, issn: 0364-9059. doi: 10.1109/JOE.2010.2083070 (cit. on pp. 23, 27, 51).

[75] K. Yang, J. Yuh, and S. Choi, “Experimental study of fault-tolerant system design for underwater robots”, in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2, May 1998, pp. 1051–1056. doi: 10.1109/ROBOT.1998.677229 (cit. on p. 24).

[76] N. Sarkar, T. Podder, and G. Antonelli, “Fault-accommodating thruster force allocation of an auv considering thruster redundancy and saturation”, *IEEE Transactions on Robotics and Automation*, pp. 223–233, Apr. 2002, issn: 1042-296X. doi: 10.1109/TRA.2002.999650 (cit. on pp. 24–25, 58).

[77] E. Omerdic and G. Roberts, “Thruster fault diagnosis and accommodation for open-frame underwater vehicles”, *Control Engineering Practice*, vol. 12, no. 12, pp. 1575–1598, 2004, issn: 0967-0661. doi: 10.1016/j.conengprac.2003.12.014 (cit. on pp. 24–25, 57, 59, 98).

[78] A. Hanai, G. Marani, and S. K. Choi, “Automatic fault-accommodating thrust redistribution for a redundant auv”, *5th JSME/RMD International Conference on Advanced Mechatronics*, 2010 (cit. on pp. 25–26, 57).

[79] G. Karras, C. Bechlioulis, H. Abdella, T. Larkworthy, K. Kyriakopoulos, and D. Lane, “A robust sonar servo control scheme for wall-following using an autonomous underwater vehicle”, in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov. 2013, pp. 3893–3898. doi: 10.1109/IROS.2013.6696913 (cit. on p. 26).

[80] G. Karras, C. Bechlioulis, S. Nagappa, N. Palomeras, K. Kyriakopoulos, and M. Carreras, "Motion control for autonomous underwater vehicles: A robust model-free approach", in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 6529–6534. doi: 10.1109/ICRA.2014.6907822 (cit. on p. 26).

[81] J. Willcox, J. Bellingham, Y. Zhang, and A. Baggeroer, "Performance metrics for oceanographic surveys with autonomous underwater vehicles", *Oceanic Engineering, IEEE Journal of*, vol. 26, no. 4, pp. 711–725, Oct. 2001, issn: 0364-9059. doi: 10.1109/48.972114 (cit. on pp. 26, 34, 43, 64, 69).

[82] M. Cashmore, M. Fox, T. Larkworthy, D. Long, and D. Magazzeni, "Auv mission control via temporal planning", in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 6535–6541. doi: 10.1109/ICRA.2014.6907823 (cit. on pp. 26, 31, 64).

[83] L. Wang, K. Tan, and C. Chew, *Evolutionary robotics: From algorithms to implementations*, ser. World Scientific series in robotics and intelligent systems. World Scientific Publishing, 2006, isbn: 9789812568700 (cit. on p. 26).

[84] S. Ahmadzadeh, M. Leonetti, A. Carrera, M. Carreras, P. Kormushev, and D. Caldwell, "Online discovery of auv control policies to overcome thruster failures", in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 6522–6528. doi: 10.1109/ICRA.2014.6907821 (cit. on pp. 26, 52).

[85] S. Ahmadzadeh, P. Kormushev, and D. Caldwell, "Multi-objective reinforcement learning for auv thruster failure recovery", in *Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), 2014 IEEE Symposium on*, Dec. 2014, pp. 1–8. doi: 10.1109/ADPRL.2014.7010621 (cit. on pp. 26, 52).

[86] S. M. Veres, L. Molnar, N. K. Lincoln, and C. P. Morice, "Autonomous vehicle control systems - a review of decision making", *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 2, pp. 155–195, 2011. doi: 10.1177/2041304110394727 (cit. on p. 27).

[87] R. Remenyte-Prescott, J. Andrews, and P. Chung, "An efficient phased mission reliability analysis for autonomous vehicles", *Reliability Engineering & System Safety*, vol. 95, no. 3, pp. 226–235, 2010, issn: 0951-8320. doi: 10.1016/j.ress.2009.10.002 (cit. on p. 27).

[88] M. Seto, "On-line learning with evolutionary algorithms towards adaptation of underwater vehicle missions to dynamic ocean environments", in *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, vol. 1, Dec. 2011, pp. 235–240. doi: 10.1109/ICMLA.2011.110 (cit. on pp. 28, 30, 38, 43, 45, 64).

[89] B. Saha, E. Koshimoto, C. Quach, E. Hogge, T. Strom, B. Hill, S. Vazquez, and K. Goebel, "Battery health management system for electric uavs", in *Aerospace Conference, 2011 IEEE*, Mar. 2011, pp. 1–9. doi: 10.1109/AERO.2011.5747587 (cit. on p. 28).

[90] B. Saha, C. Quach, and K. Goebel, "Optimizing battery life for electric uavs using a bayesian framework", in *Aerospace Conference, 2012 IEEE*, Mar. 2012, pp. 1–7. doi: 10.1109/AERO.2012.6187365 (cit. on p. 28).

[91] N. Ure, G. Chowdhary, T. Toksoz, J. How, M. Vavrina, and J. Vian, "An automated battery management system to enable persistent missions with multiple aerial vehicles", *Mechatronics, IEEE/ASME Transactions on*, vol. 20, no. 1, pp. 275–286, Feb. 2015, issn: 1083-4435. doi: 10.1109/TMECH.2013.2294805 (cit. on p. 28).

[92] C. Harris and R. Dearden, "Contingency planning for long-duration auv missions", in *Autonomous Underwater Vehicles (AUV), 2012 IEEE/OES*, Sep. 2012, pp. 1–6. doi: 10.1109/AUV.2012.6380747 (cit. on p. 31).

[93] N. Palomeras, A. Carrera, N. Hurts, G. C. Karras, C. P. Bechlioulis, M. Cashmore, D. Magazzeni, D. Long, M. Fox, K. J. Kyriakopoulos, P. Kormushev, J. Salvi, and M. Carreras, "Toward persistent autonomous intervention in a subsea panel", *Autonomous Robots*, pp. 1–28, 2015, issn: 1573-7527. doi: 10.1007/s10514-015-9511-7 (cit. on p. 31).

[94] M. Ghallab, D. Nau, and P. Traverso, *Automated planning: Theory & practice*. Elsevier, 2004 (cit. on p. 31).

[95] M. Cashmore, M. Fox, T. Larkworthy, D. Long, and D. Magazzeni, "Planning inspection tasks for auvs", in *2013 OCEANS - San Diego*, Sep. 2013, pp. 1–8 (cit. on p. 31).

[96] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005 (cit. on p. 31).

[97] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 1998 (cit. on p. 31).

[98] A. Alvarez, A. Caiti, and R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean", *Oceanic Engineering, IEEE Journal of*, vol. 29, no. 2, pp. 418–429, Apr. 2004, issn: 0364-9059. doi: 10.1109/JOE.2004.827837 (cit. on pp. 31, 43).

[99] B. Garau, A. Alvarez, and G. Oliver, "Path planning of autonomous underwater vehicles in current fields with complex spatial variability: An a* approach", in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, Apr. 2005, pp. 194–198. doi: 10.1109/ROBOT.2005.1570118 (cit. on p. 31).

[100] D. Kruger, R. Stolkin, A. Blum, and J. Briganti, "Optimal auv path planning for extended missions in complex, fast-flowing estuarine environments", in *Robotics and Automation, 2007 IEEE International Conference on*, Apr. 2007, pp. 4265–4270. doi: 10.1109/ROBOT.2007.364135 (cit. on p. 31).

[101] N. K. Yilmaz, C. Evangelinos, P. F. Lermusiaux, and N. M. Patrikalakis, "Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming", *Oceanic Engineering, IEEE Journal of*, vol. 33, no. 4, pp. 522–537, 2008 (cit. on p. 31).

[102] P. A. Forero, S. K. Lapic, C. Wakayama, and M. Zorzi, "Rollout algorithms for data storage and energy-aware data retrieval using autonomous underwater vehicles", in *Proceedings of the International Conference on Underwater Networks & Systems*, ACM, 2014, p. 22 (cit. on p. 31).

[103] S. Au, C. Papadimitriou, and J. Beck, "Reliability of uncertain dynamical systems with multiple design points", *Structural Safety*, vol. 21, no. 2, pp. 113–133, 1999, issn: 0167-4730. doi: 10.1016/S0167-4730(99)00009-0 (cit. on p. 31).

[104] A. Birolini, *Reliability engineering*. Springer, 2007, vol. 5, isbn: 978-3-642-39534-5. doi: 10.1007/978-3-642-39535-2 (cit. on p. 31).

[105] S. Kadry and A. El Hami, *Numerical methods for reliability and safety assessment*. Springer, 2015, isbn: 978-3-319-07166-4. doi: 10.1007/978-3-319-07167-1 (cit. on p. 31).

[106] M. Ceraolo and G. Pede, "Techniques for estimating the residual range of an electric vehicle", *IEEE Transactions on Vehicular Technology*, vol. 50, no. 1, pp. 109–115, 2001, issn: 00189545. doi: 10.1109/25.917893 (cit. on p. 34).

[107] A. Sadrpour, J. Jin, and A. Ulsoy, "The role of operator style on mission energy requirements for tele-operated unmanned ground vehicles", in *American Control Conference (ACC), 2014*, Jun. 2014, pp. 1553–1559. doi: 10.1109/ACC.2014.6859089 (cit. on p. 34).

[108] R. Parasuraman, K. Kershaw, P. Pagala, and M. Ferre, "Model based on-line energy prediction system for semi-autonomous mobile robots", in *Intelligent Systems, Modelling and Simulation (ISMS), 2014 5th International Conference on*, Jan. 2014, pp. 411–416. doi: 10.1109/ISMS.2014.76 (cit. on p. 34).

[109] S. Vijayakumar, A. D'Souza, and S. Schaal, "LWPR: A scalable method for incremental online learning in high dimensions", 2005 (cit. on pp. 40, 45, 71).

[110] G. Fagogenis, D. Flynn, and D. Lane, "Improving underwater vehicle navigation state estimation using locally weighted projection regression", in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 6549–6554. doi: 10.1109/ICRA.2014.6907825 (cit. on pp. 45, 99).

[111] G. Fagogenis, V. De Carolis, and D. Lane, "Online fault detection and model adaptation for underwater vehicles", in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, May 2016 (cit. on p. 45).

[112] C. Barbalata, V. D. Carolis, M. W. Dunnigan, Y. Petillot, and D. Lane, "An adaptive controller for autonomous underwater vehicles", in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sep. 2015, pp. 1658–1663. doi: 10.1109/IROS.2015.7353590 (cit. on pp. 50, 96).

[113] G. Antonelli, *Underwater robots motion and force control of vehicle-manipulator systems*. Springer, 2006, isbn: 978-3-540-31752-4 (cit. on p. 51).

[114] L. F. Mendonça, J. Sousa, and J. S. da Costa, "An architecture for fault detection and isolation based on fuzzy methods", *Expert systems with applications*, vol. 36, no. 2, pp. 1092–1104, 2009. doi: 10.1016/j.eswa.2007.11.009 (cit. on p. 55).

[115] G. Strang, *Linear algebra and its applications*. Thomson, Brooks/Cole, 2006, isbn: 9780030105678 (cit. on pp. 58, 98).

[116] T. Johansen, T. Fossen, and S. Berge, "Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming", *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 1, pp. 211–216, Jan. 2004, issn: 1063-6536. doi: 10.1109/TCST.2003.821952 (cit. on p. 59).

[117] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization", *Journal of Machine Learning Research*, 2016 (cit. on p. 59).

[118] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems", in *European Control Conference (ECC)*, 2013, pp. 3071–3076 (cit. on p. 59).

[119] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem: A survey", *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, 2011. doi: 10.1016/j.ejor.2010.03.045 (cit. on pp. 64, 75).

[120] G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms", *European Journal of Operational Research*, vol. 59, no. 2, pp. 231–247, 1992. doi: 10.1016/0377-2217(92)90138-Y (cit. on p. 64).

[121] F. Li, B. Golden, and E. Wasil, "The open vehicle routing problem: Algorithms, large-scale test problems, and computational results", *Computers & Operations Research*, vol. 34, no. 10, pp. 2918–2930, 2007. doi: 10.1016/j.cor.2005.11.018 (cit. on pp. 64, 76).

[122] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to informative path planning for persistent monitoring tasks", in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Sep. 2014, pp. 342–349. doi: 10.1109/IROS.2014.6942582 (cit. on p. 64).

[123] Z. Rymansaib, P. Iravani, and M. N. Sahinkaya, "Exponential trajectory generation for point to point motions", *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics: Mechatronics for Human Wellbeing, AIM 2013*, no. 2, pp. 906–911, 2013, issn: 2159-6247. doi: 10.1109/AIM.2013.6584209 (cit. on p. 72).

[124] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots", *Journal of Artificial Intelligence Research*, pp. 707–755, 2009 (cit. on p. 75).

[125] J. Binney, A. Krause, and G. S. Sukhatme, "Informative path planning for an autonomous underwater vehicle", in *Robotics and automation (icra), 2010 IEEE international conference on*, IEEE, 2010, pp. 4791–4796 (cit. on p. 75).

[126] L. Heng, A. Gotovos, A. Krause, and M. Pollefeys, "Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments", in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, IEEE, 2015, pp. 1071–1078 (cit. on p. 75).

[127] S. Frolov, B. Garau, and J. Bellingham, "Can we do better than the grid survey: Optimal synoptic surveys in presence of variable uncertainty and decorrelation scales", *Journal of Geophysical Research: Oceans*, vol. 119, no. 8, pp. 5071–5090, 2014 (cit. on p. 75).

[128] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic uav and ugv system for precision agriculture", in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, IEEE, 2013, pp. 5321–5326 (cit. on p. 75).

[129] F. Maurelli, T. Larkworthy, D. Lane, G. Karras, C. Bechlioulis, and K. Kyriakopoulos, "Pose-based and velocity-based approaches to autonomous inspection of subsea structures", in *Proceedings of IEEE-MTS Oceans'13, San Diego, USA*, 2013 (cit. on p. 82).

[130] P. J. Sanz, P. Ridao, G. Oliver, G. Casalino, C. Insaurralde, C. Silvestre, C. Melchiorri, and A. Turetta, "Trident: Recent improvements about autonomous underwater intervention missions", in *Navigation, guidance and control of underwater vehicles*, vol. 3, 2012, pp. 355–360. doi: 10.3182/20120410-3-PT-4028.00059 (cit. on p. 83).

[131] G. Ferri, F. Ferreira, and V. Djapic, "Boosting the talent of new generations of marine engineers through robotics competitions in realistic environments: The sauc-e and eurathlon experience", in *OCEANS 2015 - Genova*, May 2015, pp. 1–6. doi: 10.1109/OCEANS-Genova.2015.7271509 (cit. on p. 83).

[132] H. K. Abdella, D. M. Lane, and F. Maurelli, "Sonar based mapping using phd filter", in *Oceans - St. John's, 2014*, Sep. 2014, pp. 1–7. doi: 10.1109/OCEANS.2014.7003083 (cit. on p. 85).

[133] S. Björklund, "A survey and comparison of time-delay estimation methods in linear systems", Linköpings universitet, 2003 (cit. on p. 94).

[134] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011, isbn: 9781119991496 (cit. on p. 99).

[135] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: An open-source robot operating system", in *ICRA workshop on open source software*, vol. 3, 2009 (cit. on p. 101).

[136] M. Cashmore, M. Fox, D. Long, D. Magazzeni, and B. Ridder, "Artificial intelligence planning for auv mission control", *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 262–267, 2015, 4th IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles, issn: 2405-8963. doi: 10.1016/j.ifacol.2015.06.043 (cit. on p. 102).

[137] M. Cashmore, M. Fox, D. Long, D. Magazzeni, B. Ridder, A. Carrera, N. Palomeras, N. Hurtos, and M. Carreras, "Rosplan: Planning in the robot operating system", in *Twenty-Fifth International Conference on Automated Planning and Scheduling*, 2015 (cit. on pp. 42, 102).

[138] I. Gurobi Optimization, *Gurobi optimizer reference manual*, 2015. [Online]. Available: http://www.gurobi.com (cit. on p. 124).

[139] T. Danner and L. E. Kavraki, "Randomized planning for short inspection paths", in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 2, Apr. 2000, pp. 971–976. doi: 10.1109/ROBOT.2000.844726 (cit. on p. 124).

[140] K. Wang and Z. Shen, "A gpu-based parallel genetic algorithm for generating daily activity plans", *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1474–1480, Sep. 2012, issn: 1524-9050. doi: 10.1109/TITS.2012.2205147 (cit. on p. 140).