Learning, Localization, and Control of Hydrobatic Micro Underwater Robots for Autonomous Field Exploration in Confined Environments

Vom Promotionsausschuss der
Technischen Universität Hamburg
zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

von
Daniel-André Dücker

aus
Kiel

2022

1. Gutachter: Prof. Dr.-Ing. Prof. E.h. Edwin Kreuzer
2. Gutachter: Prof. Dr. rer. nat. Jianwei Zhang
3. Gutachter: Prof. Dr.-Ing. Moustafa Abdel-Maksoud

Tag der mündlichen Prüfung: 11. Juli 2022

MuM Notes in Mechanics and Dynamics
Editor: Prof. Dr.-Ing. Robert Seifried
Hamburg University of Technology
Institute of Mechanics and Ocean Engineering (MuM)
www.tuhh.de/mum

Volume 6
Daniel-André Dücker
"Learning, Localization, and Control of Hydrobatic Micro Underwater Robots for Autonomous Field Exploration in Confined Environments"
Hamburg, 2022

DOI: https://doi.org/10.15480/882.4602
ORCID: https://orcid.org/0000-0001-7256-6984

Abstract

Exploration and monitoring of hazardous fields in confined marine environments are among of the most promising tasks to be performed by fleets of low-cost micro autonomous underwater vehicles (μAUVs). The confined and complex nature of these scenarios demands for micro robots with agile maneuvering capabilities. In contrast to vehicles in other domains, underwater robots are forced to perform all computations onboard as no powerful communication links are available underwater.

A detailed problem analysis revealed that, despite recent progress, computationally efficient solutions for guidance, navigation, and control are largely unsolved for agile micro underwater robots. This dissertation contributes to these three research areas.

Firstly, we present the hydrobatic μAUV platform HippoCampus along with a powerful control system that allows agile maneuvering in confined spaces. In this context, potential synergies between the field of acrobatic aerial drones and hydrobatic underwater robots are explored. As a result, dynamic similarities between both platform types are exploited which allows benefiting from existing aerial control approaches. Furthermore, an experimental case study is conducted to demonstrate the maneuvering performance of the HippoCampus platform and its control system.

Secondly, agile maneuvering requires robust and accurate underwater self-localization methods. Today's absence of these systems constitutes a severe bottleneck for the future development of μAUVs. This work proposes a modular embedded self-localization system that uses visual, electromagnetic, and acoustic ranging. Its performance is studied in various field trials ranging from controlled test basins to marinas.

Finally, we propose a lightweight framework for autonomous field exploration with μAUV fleets. The framework combines a deep reinforcement learning exploratory planner with a stochastic field representation. This combination allows to directly incorporate the field uncertainty into the planning process. The reduced computational complexity allows for decentralized design. The performance of the exploration framework is studied for single and multi-agent settings in various configurations. As a result, the proposed approach shows superior performance against state-of-the-art benchmark algorithms.

Acknowledgement

This dissertation summarizes the main parts of my research conducted during my time at the Institute of Mechanics and Ocean Engineering at Hamburg University of Technology. I would like to deeply thank everybody who provided support and encouragement throughout this time.

I am grateful to my advisor Prof. Edwin Kreuzer who offered me the opportunity to explore the exciting field of mobile underwater robotics. He provided guidance and support throughout all stages of this dissertation and I am thankful for the freedom to follow my own ideas. Moreover, I would like to thank Prof. Jianwei Zhang and Prof. Moustafa Abdel-Maksoud for acting on my dissertation committee and Prof. Stefan Krüger for serving as the committee chair. Furthermore, I would like to thank Prof. Robert Seifried for the trust he put into me and the freedom he offered me to find and follow my own path into robotics research and teaching.

This Ph. D. project would have been neither started nor finished without the very flexible administrative and financial support from many sides including TUHH, DFG, and DAAD which is highly appreciated.

Being part of the Institute of Mechanics and Ocean Engineering was a pleasure. I am particularly grateful to Eugen Solowjow for opening the world of robotics research to me and motivating me to pursue a Ph. D. at MuM. He provided initial guidance while getting on track and the lesson learned that Ph. D.-life is a marathon rather than a long sprint. I thank Marc-André Pick for his continuing support in making things work, for engaging discussions about various views on research and life, and for his concept “if you can show it with an elephant, show it with an elephant”. Niklas Meyer was the perfect ‘office mate’ in 0517 and his creative contributions to underwater localization will always put a smile on my face. I want to thank Svenja Drücker for the common passion and time invested in designing new classes and summer school courses, and for questioning life in academia.

Social and research life at MuM was always enjoyable thanks to Timo, Leo, Ali, Johanna, Malte, Tobi, Alex, Helge, Merlin as well as Margret and Bärbel. Furthermore, I want to thank Riza, Wolfgang, and Norbert for their technical expertise and support when starting from some kind of "Bastelei" towards something that almost deserves the name "prototype". It was a pleasure to work on common ideas and projects leading 'into the wild' with Fabian Steinmetz. I want to thank Stefan Schulz for the common fun-time exploring international conferences and for his support and supply in 3D-printing. I also thank my former theses supervisors Axel Hackbarth, Max Schmidt, and Julian Theis whose advice and research thoughts guided me way further than the original theses.

I am deeply grateful to get to know and supervise such excellent students. Representative for more than 35 students, I am especially thankful to Nathalie Bauschmann for her persistent support and commitment to research and teaching. It was a great pleasure creating new courses and making new 'fun-projects' fly together with her, Thies Lennart Alff, and Tim Hansen. Furthermore, I want to thank the student "start-crew" of Team HippoCampus among Viktor Rausch, Tobias Johannink, Andreas René Geist, and Gerrit Brinkmann.

I am deeply grateful to my friends who supported and encouraged me throughout this journey. I would like to thank Christoph and Katrin for their support and patience and for always offering a home-like place with amazing dinner. In the same way, I could always count on Maik's support and spontaneous late-night office visits. Sharing teaching and research anecdotes was fun with Christine and Daniel who never gave up pulling me into sports. Not to forget, Steffen, Hilli, and Anna who provided a never-ending pool for reflection and critical self-questioning. Moreover, I want to thank Lasse and Christoph S. for being anchors to the ground and for endless discussions on important and less important topics of life.

Last but definitely not least, I am extremely thankful for the immense encouragement, support, and patience of Lena and my family, especially when deadlines, traveling, and unplanned events encountered common time. I am deeply grateful to my mother Karin for her lifelong support. Finally, I want to thank my sister Dr. Fenja Leena Dücker, who finally graduated first, for sharing the tough journey towards the Ph. D. degree.

Hamburg, July 2022
Daniel-André Dücker

To my family and friends.

Contents

List of Figures

List of Tables

List of Important Abbreviations and Symbols

Abbreviations

ACK acknowledgement

AUV autonomous underwater vehicle

A3C asynchronous advantage actor critic

BO Bayesian optimization

CB center of buoyancy

CG center of gravity

CFD computational fluid dynamic

CNN convolutional neural network

CTD conductivity, temperature, and depth

DRL deep reinforcement learning

Dec-POMDP decentralized partially observable Markov decision process

DQN deep Q-network

DDQN double deep Q-learning

DOF degree of freedom

DVB-T Digital Video Broadcasting-Terrestrial

EKF extended Kalman filter

FFT fast Fourier Transformation

FCU flight computing unit

GP Gaussian process

GRF Gaussian random field

GMRF Gaussian Markov random field

GNSS global navigation satellite system

GPS global positining system

GNC guidance, navigation, and control

I^{2}C Inter-Integrated Circuit

IQR inter-quartile range

IPP informative path planning

IMU inertial measurement unit

LOS line of sight

MDP Markov decision process

MCTS Monte-Carlo Tree Search

μAUV micro autonomous underwater vehicle

MEMS micro-electro-mechanical systems

PI path integral

PDE partial differential equation

POMDP partially observable Markov decision process

PSD power spectrum density

RL reinforcement learning

RMSE root mean squared error

ROV remotely operated vehicle

ROS Robot Operating System

RRT rapidly exploring random tree

RSM range sensor model

RSS received signal strength

RW random walk

SLAM simultaneous localization and mapping

SBC single board computer

SDR software defined radio

SBL short baseline

TOF time of flight

TWR two-way-ranging

UAV unmanned aerial vehicle

uORB micro Object Request Broker

URSM underwater range sensor model

USBL ultra short baseline

Functions and Scalars

a	action
c	tuning parameter
e	Euler's number
g	gravitational acceleration
h	observation function
k	discrete timestep
m	mass
p	roll velocity
q	pitch velocity
r	yaw velocity, reward term
u	surge velocity

v	heave velocity, hull volume
w	sway velocity
x, y, z	Cartesian coordinates
C	drag coefficient
F	force
J	advance ratio
K	moment
L	loss function
M	moment
N	moment, number of agents
O	origin
P	wave power
Q	Q-function
R	spherical distance, reward
Y	target value
RSS	received signal strength
X, Y, Z	forces
α	attenuation constant, learning rate
γ	discount factor
ϵ	quaternion element
θ	pitch angle
κ	thrust coefficient, hyperparameter
μ	measurement, observation
ν	kinematic velocity coefficient
ρ	fluid density
σ	flat output
τ	hyperparameter
ϕ	roll angle
ψ	yaw angle
Γ	RSM offset factor

Matrices and Vectors

$\boldsymbol{b}$	GMRF canonical mean
$\boldsymbol{e}$	unit vector, control error vector
$\boldsymbol{f}$	force vector, vehicle motion model
$\boldsymbol{g}$	hydrostatic load vector

$\boldsymbol{h}$	observation function
$\boldsymbol{m}$	moment vector
$\boldsymbol{o}$	rendered observation
$\boldsymbol{p}$	vehicle position
$\boldsymbol{q}$	orientation vector
$\boldsymbol{r}$	position vector
$\boldsymbol{s}$	environment state
$\boldsymbol{u}$	vector of control inputs
$\boldsymbol{v}$	translational velocity
$\boldsymbol{w}$	process noise vector
$\boldsymbol{z}$	field measurement
$\boldsymbol{C}$	added Coriolis matrix
$\boldsymbol{D}$	added damping matrix
$\boldsymbol{F}$	linearized vehicle motion model
$\boldsymbol{I}$	identity matrix
$\boldsymbol{J}$	inertia tensor
$\boldsymbol{K}$	Control gain, Kalman gain matrix
$\boldsymbol{Q}$	covariance matrix
$\boldsymbol{R}$	rotational matrix
$\boldsymbol{T}$	transformation matrix, configuration matrix
$\boldsymbol{\alpha}$	angular acceleration vector
$\boldsymbol{\beta}$	regression coefficient vector
$\boldsymbol{\eta}$	inertial state vector
$\boldsymbol{\theta}$	network parameter
$\boldsymbol{\mu}$	field mean, observation vector
$\boldsymbol{\nu}$	velocity vector
$\boldsymbol{\sigma}$	vector of flat outputs
$\boldsymbol{\tau}$	vector of generalized forces and moments
$\boldsymbol{\omega}$	angular velocity vector
$\boldsymbol{\Theta}$	Euler angles
$\boldsymbol{\Lambda}$	GMRF precision matrix
$\boldsymbol{\Sigma}$	state covariance matrix
$\boldsymbol{\Phi}$	mapping matrix

Indices

del	delay
des	desired
ref	reference
Tx	transceiver
Rx	receiver
A	added mass/damping
D	derivative feedback
P	proportional feedback
(-)/(+)	a priori / posteriori
$\sim$	distributed as

Operators and Other Symbols

$\mathcal{A}$	action space
$\mathcal{E}$	environmental belief
$\mathcal{F}$	field object
$\mathcal{N}$	normal distribution
$\mathcal{O}$	observation space
$\mathcal{S}$	environment state space, field grid
$\mathcal{D}$	replay memory
$p(\cdot)$	probability
$\mathbb{E}[\cdot]$	expectation
VAR$[\cdot]$	variance
∂	partial derivative
∇	Nabla-operator
Δ	difference
$()^\top$	transpose
$()^\vee$	*vee*-map operator

Chapter 1 Introduction

Exploration and monitoring of marine environments have a long history and constitute important and active fields of research. By incorporating the data gathered from these missions, scientists from various disciplines obtain fundamental insights into dynamic processes which are present in marine domains. Quantities of interest usually occur in form of spatio-temporal environmental fields. Examples include among others temperature, chemical concentration, radiation, and flow fields. However, data collection in marine and especially submerged settings comes with high effort. Furthermore, many underwater domains are often too dangerous to be monitored by human divers. This particularly covers many complex and confined environments which are potentially hazardous. Their long list includes harbor basins, industry tanks, and extreme environments such as nuclear storage ponds and disaster sides.

An obvious approach is to conduct monitoring with autonomous underwater vehicles (AUVs) which have become a prominent sophisticated tool in oceanic monitoring. Increasing the autonomy level of AUVs is an active field of research and is expected to gain even more attention in the near future. However, due to their size, cost, and limited maneuvering capabilities these vessels cannot be used in confined settings which are characterized by length scales up to tenths of meters with a maximum being about 100 m.

In this sense, aquatic operations in confined environments constitute a challenge on their own. From the robotic platform point of view, highly autonomous small-scale robots are required which perform their exploration and monitoring mission without human supervision. An example of this mission setting is illustrated in Figure 1.1. These small-scale underwater robots constitute a relatively new class of robots and are often referred to as μAUVs if their characteristic length scale is below 50 cm. For obvious reasons, a trade-off arises between autonomous capabilities, vehicle size, and unit cost. On the upside, the development of μAUVs was boosted in recent years due to the miniaturization of electronic components and the success of aerial drones being their counterpart. Note that

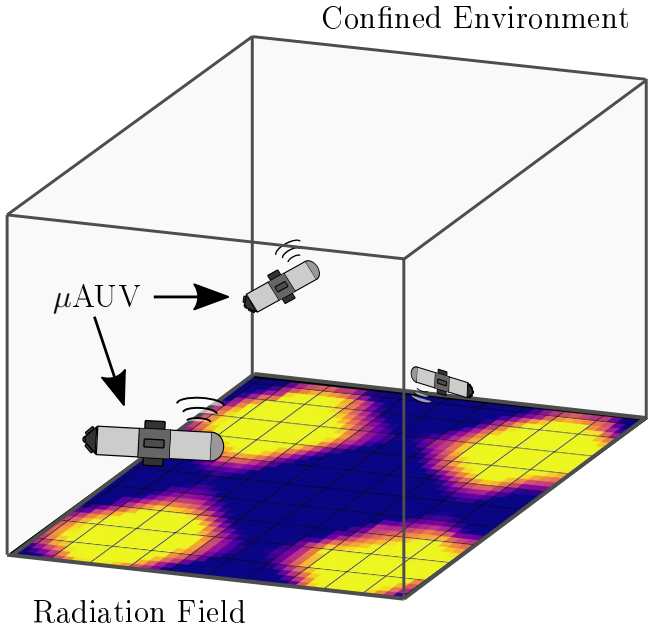


Figure 1.1: A fleet of micro autonomous underwater vehicles (μAUVs) performing an exploratory mission to monitor a radiation field in a confined fluid environment.

while small-scale aerial drones, such as quadrotors, became commercially available to a wide public this is not the case for μAUVs. Furthermore, μAUV designs, similar to their aerial siblings, often allow for agile maneuvering which is an important capability in confined scenarios since these are potentially cluttered.

This aspect in combination with their low-cost price opens a wide range of new application opportunities. Examples include the parallel deployment of multiple vehicles. These eventually heterogeneous fleets are expected to leverage the exploration capabilities of a single vehicle. Given the nonlinear hydrodynamic interaction with their fluid surrounding, agile μAUV control constitutes an exciting topic on its own.

In order to emphasize the similarities to acrobatic maneuvering with aerial drones, the term *hydrobatics* was established in recent series of publications [6, 7, 27, 29]. Hydrobatics refers to agile maneuvering of underactuated underwater robots which reach speeds up to multiple body-lengths/second (BL/s). In this sense, it has become the marine counterpart of aerobatics in aerial robotics.

An important aspect when operating underwater is the strong limitations posed by the surrounding fluid. Standard positioning concepts such as the global navigation satellite system (GNSS) are not available underwater due to the strong attenuation of electromagnetic waves in water. The same yields for communication links. As a consequence, communication bandwidth is very limited, unreliable, and data packages are subject to considerable latencies.

In this context, the small size and, thus, limited computational resources of μAUVs in combination with their confined surroundings pose challenging requirements. This applies in particular to the vehicle's localization system. Given the very limited communication resources, robust self-localization of the μAUV has to be realized on-board. In fact, this constitutes a fundamental requirement for de-facto any kind of autonomous behavior of μAUVs. Despite the recent progress, guidance, navigation, and control (GNC) are largely unsolved for agile μAUVs. In order to overcome this capability gap and to allow for robotic operations in confined and complex underwater environments, robust and accurate self-localization systems which fit μAUVs play a key role [133]. In other words, their absence constitutes a severe bottleneck in micro underwater robotics research.

1.1 Thesis Goal

Closing the loop to the original goal of exploration and monitoring of marine environments, our goal is to effectively conduct exploratory missions with one or a fleet of μAUVs. These tasks are also referred to as the informative path planning (IPP) and adaptive sampling problem. It is notable that in the light of μAUVs being the target platform potential algorithms have to be lightweight enough to run on-board since μAUVs offer only very restricted computational resources. However, controlling and coordinating robot fleets are known to be complex tasks. Human control of such a robot fleet is at best challenging, however, it often ends up being just impossible. From a control perspective, field exploration and monitoring link classic control theory and path planning methods with the area of information-theoretic control. Traditionally, handcrafting control policies for these exploratory missions has been complex and tedious. The recent progress in the field of data-driven methods offers promising techniques from the field of machine learning. However, so far, their applications to environmental field exploration are rare.

This dissertation addresses important aspects within the areas of marine field exploration and monitoring. Based on a detailed problem analysis three key areas are identified which can be currently seen as bottlenecks. This dissertation contributes to these areas, namely, design and control of agile micro underwater robots, self-localization for μAUVs, and computational feasible information gathering control.

1.2 Contribution and Thesis Outline

This dissertation contributes to three areas within the research field of exploration and monitoring of environmental fields with μAUV groups. As a foundation, we conduct a detailed problem analysis in Chapter 2. Based on this problem analysis, we formulate three reference scenarios to cover important aspects of confined environments for further development steps. First, we present in Chapter 3 a fully revised design of the HippoCampus μAUV. The design revision is accompanied by a new embedded control architecture. The combination of both enables agile maneuvering control and extends HippoCampus' autonomous capabilities to a new level which is considerably beyond the state of the art. Second, we address the problem of underwater μAUV self-localization in Chapter 4. For this, we present a modular state estimation framework in combination with a total of three lightweight self-localization methods. All methods are evaluated in experimental studies. Third, we propose a new information-theoretic field exploration methodology in Chapter 5. It is based on deep reinforcement learning and a Gaussian Markov random field belief representation. Each chapter concludes with a summary section pointing out the highlights and contributions of the corresponding chapter.

Hydrobatic Micro Underwater Robot HippoCampus

The HippoCampus X is an underactuated, low-cost, and modular μAUV platform which we present in Chapter 3. It is a full revision of the early prototype from 2015 [46]. With this new design, we explore and exploit the link between small aerobatic aerial drones and agile micro underwater robots [29]. The new HippoCampus version features a powerful modular and open-source state estimation and control architecture that can easily be transferred to other underwater robot platforms. It is specifically designed for autonomous operations in complex environments. Therefore, HippoCampus possesses highly agile maneuvering capabilities. These let the HippoCampus μAUV become one of the founding members of the new class of *hydrobatic* μAUVs [7].

Contents of this chapter have been partly published in:

- **D.-A. Duecker**, C. Horst, and E. Kreuzer. From Aerobatics to Hydrobatics: Agile Trajectory Planning and Tracking for Micro Underwater Robots. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8617-8624, Prague, CZ, 2021.
- **D.-A. Duecker**, N. Bauschmann, T. Hansen, E. Kreuzer, and R. Seifried. HippoCampus X – A Hydrobatic Open-source Micro AUV for Confined Environments. In: *IEEE/OES Autonomous Underwater Vehicles (AUV) Symposium*, pages 1-6, St. Johns, Canada, 2020.

- **D.-A. Duecker**, E. Kreuzer, G. Maerker, and E. Solowjow. Parameter Identification for Micro Underwater Vehicles. In: *Proceedings of Applied Mathematics and Mechanics 2018 (PAMM)*, pages 1-2, Munich, Germany, 2018.

- **D.-A. Duecker**, A. Hackbarth, T. Johannink, E. Kreuzer, and E. Solowjow. Micro Underwater Vehicle Hydrobatics: A Submerged Furuta Pendulum. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7498-7503, Brisbane, Australia, 2018.

Underwater Self-localization in Confined Environments

Robust and accurate self-localization techniques constitute a crucial bottleneck that hinders the development of micro underwater robots. The main challenges for μAUV self-localization systems arise from the requirement of being used in strictly confined workspaces while their algorithms are implemented embedded onboard the robot with limited computational resources. At the same time, the localization system has to be small-size and low-cost. Based on a detailed analysis, Chapter 4 of this dissertation proposes a self-localization framework that specifically addresses the demands of μAUVs. For this purpose, a modular system based on three core technologies is developed, namely, visual ranging using fiducial landmarks, ranging based on the attenuation of electromagnetic carrier signals, and acoustic two-way ranging. All three localization concepts are developed and implemented onboard the HippoCampus platform. The developed system is extensively tested in various real-world experimental settings to cover various deployment scenarios of μAUVs in confined environments. The field trials include a marina, a public swimming pool, and a research basin. Throughout the evaluations, the proposed framework demonstrated high accuracy and robust localization performance and thus overcomes the identified capability bottleneck.

Contents of this chapter have been partly published in:

- **D.-A. Duecker**, N. Bauschmann, T. Hansen, E. Kreuzer, and R. Seifried. Towards Micro Robot Hydrobatics: Vision-based Guidance, Navigation, and Control for Agile Underwater Vehicles. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1819-1826, Las Vegas, NV, USA, 2020. (Winner of the IROS Best Entertainment and Amusement Paper Award)

- S. Watson, **D.-A. Duecker**, and K. Groves. Localization of Unmanned Underwater Vehicles (UUVs) in Complex and Confined Environments: A Review. In *Sensors*, pages 1-35, 2020, 20, 6203.

- **D.-A. Duecker**, T. Hansen, and E. Kreuzer. RGB-D Camera-based Navigation for Autonomous Underwater Inspection using Low-cost Micro AUVs. In: *IEEE/OES Autonomous Underwater Vehicles (AUV) Symposium*, pages 1-7, St. Johns, Canada, 2020.
- **D.-A. Duecker**, F. Steinmetz, E. Kreuzer, and C. Renner. Micro AUV Localization for Agile Navigation with Low-cost Acoustic Modems. In: *IEEE/OES Autonomous Underwater Vehicles (AUV) Symposium*, pages 1-7, St. Johns, Canada, 2020.
- **D.-A. Duecker**, T. Johannink, E. Kreuzer, V. Rausch, and E. Solowjow. An Integrated Approach to Navigation and Control in Micro Underwater Robotics using Radio-Frequency Localization. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6846-6852, Montreal, Canada, 2019.
- **D.-A. Duecker**, A.R. Geist, M. Hengeler, E. Kreuzer, M.-A. Pick, V. Rausch, and E. Solowjow. Embedded Spherical Localization for Micro Underwater Vehicles based on Attenuation of Electro-Magnetic Carrier Signals. In: *Sensors*, pages 959-981, 2017.

Deep Reinforcement Learning - Field Exploration Controller

Autonomous exploration of environmental fields is a challenging task within the field of information-theoretic control. Chapter 5 presents a novel methodology to this problem which combines deep reinforcement learning (specifically double deep Q-learning) with stochastic field belief representation. The proposed approach interprets the stochastic field belief in the local vicinity of the agent as an imaginary image. These images can be effectively processed by a deep Q-network that has convolutional layers as an input to obtain Q-values for action selection. The survey of related work revealed the current approaches mostly base their DRL-IPP on deterministic grid maps rather than stochastic representations which provide a natural interpretation of the field's uncertainty. An important aspect for the feasible implementation on-board computationally restricted micro robots is the constant computationally complexity. This is achieved by means of a sequential inference formulation of the measurement update step.

Contents of this chapter have been partly published in:

- **D.-A. Duecker**, A. R. Geist, E. Kreuzer, and E. Solowjow. Learning environmental field exploration with computationally constrained underwater robots: Gaussian processes meet stochastic optimal control. In: *Sensors*, 19(9), 2094, 2019.
- **D.-A. Duecker**, B. Mersch, R. C. Hochdahl, and E. Kreuzer. Embedded Stochastic Field Exploration with Micro Diving Agents using Bayesian Optimization-Guided Tree-Search and GMRFs. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8649-8656, Prague, CZ, 2021.

Chapter 2 Problem Analysis

This chapter provides a detailed analysis of the manifold aspects of field exploration in confined environments with autonomous underwater robots. As a first step, we study and differentiate deployment scenarios and point out important characteristics of confined environments. This is supplemented by a detailed overview of application scenarios in confined settings. Subsequently, we define three reference scenarios to represent the versatility of confined settings for the following investigations and examinations in this dissertation.

Next, we explore and identify requirements for underwater robots, their localization systems as well as exploration methodologies. These build the basis for our review on related work representing today's state-of-the-art in this field. Finally, we identify remaining research gaps which will be addressed in the remainder of this dissertation.

2.1 Deployment Scenarios

Underwater vehicles offer versatile applications. Traditionally prominent scenarios include oceanographic surveying and military tasks. However, technological progress and cost reductions boosted fast emerging fields such as civil exploration and inspection in complex and hazardous areas including e. g. nuclear storage facilities, flooded mines, and liquid storage tank inspection. In the following, we analyze these scenarios with respect to their deployment scenery and the application areas and their characteristics. Based on this, we define three representative reference scenarios to capture the main characteristics.

2.1.1 Target Environments

In the following, we briefly introduce and contrast the demands of open-water environments in comparison to confined settings.

Open-water Environments The category of open-water scenarios commonly includes oceans, seas, and large lakes. These settings share the following characteristics: These environments possess large length scales ranging from multiple to hundreds of kilometers. As a consequence, the velocity of deployed robotic vehicles is relatively slow compared to the environment's length scale. The same yields for the vehicle's maneuvering capabilities as planned and tracked paths mostly consist of long straight legs. Challenges arise from long mission durations which lie in the range of days, weeks, or even months. Access limitations in open-water environments are, thus, usually on the logistic side rather than posing geometrical design criteria. For instance, when considering a target area of interest, vehicle deployment may require considerable effort e. g. using support vessels. Alternatively, the vehicles may face very long approaches from the dock toward the target mission area. While obstacles may exist, i. e. islands, their spatial density is usually low given the large environment. Thus, their detection and avoidance are usually not time-critical. As a consequence, long update cycles are sufficient and obstacle avoidance can be integrated as an adaptation to the globally planned path. It can even be possible to consider obstacles a-priori as part of the mission planning phase.

Confined Environments Confined marine environments differ from open environments in various ways. First, their length scales are multiple magnitudes smaller. Typical dimensions range from multiple hundred meters for lake and harbor environments down to tenths or only a few meters for industry tanks, storage ponds, and test basins. Especially when focusing on the smaller end of confined environments suitable vehicles require maneuvering capabilities with increasing agility. The importance of agile maneuvering increases in the presence of obstacles which may be known a-priori. As the ratio of vehicle speed to volume-length-scale rises, mission duration times decrease to the range of hours. At the same time, similar missions may be conducted on a periodic basis. This shifts the focus from extremely robust long-lasting designs to the identification of effective deployment processes and convenient access to docking facilities. Concerning planning, control, and localization, accuracy and update rates become more critical than in open settings.

2.1.2 Application Areas and Characteristics

Within the category of confined environments, application areas are manifold and come with their specific characteristics. The following descriptions introduce the most relevant application areas, which are later analyzed based on their characteristics. Table 2.1 provides a concise overview of the application areas and their characteristics. The categories are based on the recent survey article by Watson, Duecker, and Groves [133].

Harbors, Marinas, and Boatyards These naval facilities constitute a combination of static structures and dynamic and vital objects including moving boats and ships. Current tasks with high potential for automation include water quality monitoring to quickly detect environmental threats e. g. oil spills. Moreover, quay walls, pile moorings, and pontoons require periodic inspection [35]. Having divers operating in these environments is often extremely dangerous due to currents and ship propellers.

Modern Nuclear Storage Ponds Wet nuclear storage facilities require continual monitoring. Modern storage ponds are usually indoors and constitute a static well-structured environment with clean water. Moreover, the ponds have good ambient illumination which facilitates visual monitoring and inspection.

Legacy Nuclear Storage Ponds This category compromises early pond designs constructed in the 1950s and 60s. They are beyond their originally scheduled lifespan and require an increased maintenance effort compared to modern facilities. While the total number of legacy ponds is limited, these ponds pose considerable decommissioning challenges [57]. In contrast to their modern counterparts legacy storage ponds are often outdoors and subject to environmental impacts.

Liquid Storage Tanks Liquid storage tanks are widely used in the process engineering industry. These tanks require periodic inspection to ensure the quality of the stored fluid, e. g. water [16]. Moreover, the detection and repair of potential structural defects may otherwise result in catastrophic failures. Maintenance can be conducted in wet (full) or dry (empty) tanks. Wet inspections are performed by human divers which may be dangerous and expensive. Small-scale underwater robots would allow reducing cost, which makes this a promising application field for these vehicles.

Moored Structures Moored structures such as aquaculture fish cages require periodical monitoring and inspection. For instance, fish outbreaks from aquacultures constitute a severe threat to the surrounding environment [59]. Additionally, regulations of these cages pose an increasing demand for continuous water quality monitoring to ensure a minimized natural impact.

Offshore Asset Decommissioning A large number of offshore assets exist which will be subject to decommissioning in the near future. These include assets at known and unknown locations. The first group is mostly associated with the energy sector including the oil and gas industry and is usually well mapped [122]. The latter group includes large amounts of world war munitions. These compromise toxic mustard weapons which have been dumped into coastal areas after 1945. One hot spot area is for instance the Baltic sea. Current estimates are that 1.6 *million* tons of munitions and explosives were dumped considering only North German and Baltic Sea waters [9]. So far, only their rough locations are known. The continuous corroding munitions constitute an increasing time-critical demand for decommissioning. This hazardous process is expected to require strong support from autonomous underwater robots at all stages. This support includes the full task pipeline from exploration and mapping to manipulation and recovery of the assets [11].

Disaster Response Disaster response scenarios summarize a wide group of potential application areas. These settings usually share an unstructured and cluttered environment whose state is mostly unknown and requires efficient exploration strategies. Depending on the scenario, hazardous threats such as radiation prohibit human diver missions. This renders the deployment of autonomous underwater robots an appealing alternative. Examples include the Fukushima Daiichi nuclear power plant in Japan which forms a highly complex and radioactive aquatic environment [87].

Research Facilities Test basins in research facilities constitute well-structured controlled environments. Their size is usually small with volume length scales ranging from meters to tenths of meters. Water quality is high with good visibility conditions. In this sense, test basins are usually used to identify an upper performance bound when testing control and localization algorithms since the controlled nature of the environment allows imposing various simplifying assumptions on the tested systems. Moreover, they provide good access to the volume and usually allow for infrastructure deployment with minimal effort.

Environment Characteristics

Scale Describes the characteristic length scales of the target environment and the available workspace. It directly affects the size of the underwater robot and the required localization accuracy. Dimensions are provided in meters describing either [width $\times$ length $\times$ depth] or [ϕ (diameter) $\times$ depth].

Obstacles Most application scenarios cover facilities containing water. The target volume is usually constrained by a floor, walls, and often a ceiling. In this sense, obstacles are defined as objects which are not part of this bounding infrastructure. If they are free-standing, they will likely be placed on the floor. However, moving boats and ships are considered dynamic obstacles.

Structure We categorize obstacles within the target volume as either structured or unstructured. Structured obstacles refer to objects that have been placed in the environment in an ordered manner. Examples include mooring piles in a marina. In contrast, unstructured refers to unknown or known objects which have been placed in an unordered manner.

Obstacle Type We classified existing obstacles as static or dynamic. Static obstacles are fixed and do not move for the duration of a mission while dynamic obstacles are moving during the mission.

Access Two methods of access will be considered: surface deployment and restricted access. The first is applicable in the absence of a ceiling at the target volume so that the robot can be deployed directly into the water, e. g. from the tank edge. The latter case of restricted access occurs at closed volumes where the robot needs to be deployed through a hatch or similar entry port, e. g. a tank.

Additional Infrastructure Some environments allow for the installation of additional infrastructure. This is often possible in open surroundings where, for example, anchor beacons could be installed around the volume's edges. The concept cannot be applied to closed environments with restricted access.

Line of Sight (LOS) If an environment has obstacles, their disposition may inhibit line of sight from the robot to various points. In this analysis, we consider the line of sight from the robot towards the surface and to the edges of the environment.

Turbidity Turbidity is a measure of a fluid's clarity and, thus, of the visibility conditions inside the volume. It is affected by the presence of suspended particles. With increasing turbidity, more light is scattered by particles. Thus, the visibility decreases. Hence, clear water has a negligible particle concentration and the turbidity is very low.

Ambient Illumination Ambient lighting is provided by external light sources, i. e. not by vehicle-mounted spotlights. Examples include spotlights on the laboratory ceiling or the sun. This may not be the case for some environments where the only light will be generated from onboard the robot.

Salient Features Certain localization technologies require the identification of features in the environment. Detectable features are known as salient features. A long smooth, uniform surface will provide no salient features, however, if you placed defined objects, such as QR codes, on the wall, these could then be detected.

Variance of Environment The variance of the environment considers changes over several robot missions. While some settings do not change over years, some may change over hours or days.

2.1.3 Definition of Reference Scenarios

Based on the previous review, we define three reference scenarios that aim to consider a wide range of requirements:

Scenario I – Hydrobatic Control in Confined Water Basins: Consider the performance examination of the guidance, navigation, and control architecture on-board a highly agile μAUV inside controlled confined volumes such as small freshwater research tanks. Sophisticated planning and control algorithms for μAUVs are required when executing complex fully autonomous missions. Considering planning and control methods, their efficient development is hindered due to the absence of the high-rate and on-board availability of robust, accurate information on the vehicle position and its yaw orientation. Note, this capability gap of evaluating the control performance in hydrobatic maneuvering is specific to the research on μAUVs. In contrast, full-scale AUVs and remotely operated vehicles (ROVs) can generally compensate for this lack by using high-fidelity inertial measurement units [61, 62] or by having access to powerful off-board computational capacities [53, 134]. Neither approach can be adapted to the domain of μAUVs as they are contrary to the goals of low-cost, small-size, and full autonomy. Note, with relaxed demands on robustness and accuracy this scenario also covers modern nuclear storage ponds, see also Table 2.1.

Scenario II – Navigation in strictly Confined Volumes: Consider a fleet of underwater robots autonomously navigating in a strictly confined volume, e. g. during a field exploration mission. In contrast to Scenario I, this scenario considers poor water quality with high turbidity instead of the previous freshwater assumption. As a consequence, vision-based approaches are rendered infeasible. However, the requirements on accuracy and robustness are slightly relaxed since the μAUV operates in a standard operation mode rather than studying the control scheme at its limits.

Table 2.1: Environmental Characteristics of Confined and Hazardous Application Areas. Categories are defined in Subsection 2.1.2.

Area	**Scale (m)**	**Obstacles**	**Structure**	**Obstacle Type**	**Access**	**Add. Infrastr.**	**LOS**	**Turbidity**	**Ambient Illumin.**	**Salient Features**	**Variance of Env.**
Modern Nuclear Storage Ponds	50×100×10 (l×w×d)	Yes	Structured	Both	Surface	Possible	Surface	Very Low	Good	Med	Very Low
Legacy Nuclear Storage Ponds	50×100×10 (l×w×d)	Yes	Unstructured	Both	Surface	No	Surface	Variable	Good	Med	Med
Offshore Asset Decommissioning	100×30 (ϕ×d)	Yes	Unstructured	Static	Surface	Yes	Maybe	Variable	Low	Med	Low
Moored Structures	20×20×10 (l×w×d)	Yes	Structured	Both	Surface	Possible	Both	Variable	Med	Med	Low
Liquid Storage Tanks	20×30 (ϕ×d)	No	Structured	Static	Restricted	No	Both	Low	None	Low	Very Low
Marinas, Harbors, Boatyards	100×100×10 (l×w×d)	Yes	Both	Both	Surface	Possible	Both	Variable	Med	Med	Very High
Disaster Response	50×50×20 (l×w×d)	Yes	Unstructured	Both	Restricted	Possible	Both	High	Variable	Med	Very High
Research Facilities	4×4×2 (l×w×d)	No	Structured	Static	Surface	Yes	Surface	Very Low	Good	Med	V. Low

Scenario III – Navigation in Marinas and Offshore Dumping Sites: For the third scenario, we consider a field exploration and monitoring scenario in a less restricted, confined environment such as a marina or an offshore asset decommissioning site. In these settings, we face relatively large length scales, which require longer ranges of the localization system. Moreover, the requirements on absolute localization accuracy can be relaxed. Although obstacles may be present, these environments are usually not tightly bound by walls. Similar to Scenario II, we cannot assume sufficient visibility conditions as required for visual localization.

2.2 Autonomous Capabilities

In this section, we first introduce the categorization of autonomy levels briefly before we analyze the tasks which are expected to be performed by underwater robots.

2.2.1 Level of Autonomy

Robot autonomy is not a binary decision between fully manual control by a human operator or fully autonomous decision-making by the robot. In fact, the autonomous capabilities of robotic systems range continuously between both extrema. These abilities do not only depend on the robot itself, but also on the environment where it is deployed. For instance, challenging surroundings, which deteriorate the sensing performance, likely reduce the ability of the robot to act fully autonomously. In order to compare the autonomous capabilities of robotic systems, it has been shown to be helpful to define different levels of autonomy [13].

For this work, we follow the convention from the survey by Watson et al. [133] and build on and use the six levels of autonomy proposed for surgical robots by Yang et al. [139] as a foundation. Considering underwater robots, we formulate these adapted level definitions in the following. However, the classification concept applies also to other robotic systems.

Level 0 – No Autonomy The underwater robot is entirely teleoperated by a human.

Level 1 – Robot Assistance The underwater robot provides fundamental assistance functionality which supports the human operator. Examples include vehicle stabilization at pre-set depths and attitudes. The human operator is still in full control of the robot.

Level 2 – Task Autonomy The underwater robot executes motion sequences under the guidance of the operator. A common example is tracking an operator-defined way-point sequence. For this particular task phase, no further input from the operator is required.

Level 3 – Conditional Autonomy The robot generates various task strategies but the decision-making is left to the human operator. For example, in an exploratory mission, the robot proposes several feasible paths to the human expert for final decision.

Level 4 – High Autonomy The robot plans and executes missions based on a set of boundary conditions specified by the operator. The decision-making is performed by the robot while the human operator is in a supervision role.

Level 5 – Full Autonomy The robot operates fully autonomously and does not require any human input. It is deployed into the environment and left with no operator oversight.

2.2.2 Task Analysis

Tasks performed by underwater robots can be generally grouped into two categories: Firstly, *exploration and monitoring* and, secondly, *maintenance and repair*. Depending on the particular setting *inspection* tasks can be assigned to either of these categories. While this dissertation focuses on the first category, we also briefly discuss the second as this allows us to point out and clarify important differences between robots.

Exploration and Monitoring Exploration and monitoring tasks describe missions in which a robot acts as a mobile sensor node. The robot aims to gather information about its environment or specific objects within its surrounding. This is also referred to as the *informative path planning* (IPP) problem. Exploratory missions form a subgroup of IPP in the sense that they usually assume less a-priori information when compared with periodic monitoring tasks. For instance, obstacle positions and walls may be unknown and have to be detected and considered for the rest of the mission. Further examples include monitoring of a temperature distribution or a chemical concentration field within a fluid volume. For most missions, the focus lies on finding a strategy that optimizes the information-gathering process with respect to a predefined cost function. This function usually trades off the information gain against the invested effort such as time spent or the consumed electric power. The information gain per time instance may be leveraged by deploying multiple robots. This robot swarm may be able to gather more information within the same time. In most applications, the gathered information is stored in some

kind of map representation. Such a map allows sharing the collected knowledge with other algorithms, robot agents, or human operators allowing them to base their decision on the gained information. It is usually convenient to express the gathered data with respect to an absolute world-fixed reference frame. Hence, the collecting robot requires access to an absolute localization system. From a dynamic perspective, the exploratory robots move mainly in their forward direction. Thus, a streamlined design is desirable which is optimized for a high and energy-efficient marching speed. However, the length-scales volume dimensions and the existence of potential obstacles pose additional requirements on the vehicle's agility and, thus, the actuator configuration.

Maintenance and Repair In contrast, maintenance and repair tasks often include some kind of interaction with a target object, i.e. manipulation. Therefore, they require prior information on the object and the environment. At the simplest level, the existence of the target object. Maintenance and repair tasks are often performed by human divers which is costly and may be hazardous for the diver depending on the scenario. In order to perform maintenance and repair tasks, the robot must be able to maneuver slow and accurately in the vicinity of the target object. Especially for manipulation tasks and detailed inspection, it may be required to hover precisely at a defined distance in front of the object. From a dynamic perspective, this favors fully actuated robot designs. Moreover, the focus lies rather on an accurate localization *relative* to the object. This is of special importance if the target object is moving as well. In this sense, absolute localization is mostly required to navigate towards the object of interest. Especially for complex manipulation tasks, tethered solutions may be appealing as they allow low-latency remote control by a human operator. However, tethered approaches inherit the risk of cable tangle which may even result in losing the robot.

2.3 Identified Challenges

Based on the previous discussion of deployment scenarios we face a situation of very heterogeneous scenarios. Thus, we look for a modular framework that fits most mission settings directly and can be easily adapted to others. For this purpose, we derive an individual set of key requirements for μAUV design and control, underwater self-localization, and the field exploration methodology.

2.3.1 Requirements on Micro Underwater Robots

As already discussed, the desired robot platform should be easy to adapt to a wide range of scenarios. Therefore, it should be modular with respect to its software architecture and hardware design. This modular design also applies to onboard sensors and the localization system. From a design perspective, the vehicle should be small-scale while possessing agile maneuvering capabilities to enable operation in complex confined environments. With regard to the targeted exploratory missions, the robot needs a powerful propulsion system to achieve high marching speeds and to counter potential currents flows. Regarding the power consumption, it needs to be ensured that its energy lasts for mission duration in the range of hours. Moreover, convenient maintenance constitutes a design criterion on its own, e. g. easy replacement of components. A low-cost design is desirable since the platform should enable multi-robot deployment. Furthermore, deployment in hazardous environments naturally comes with the risk of the vehicle being lost. Finally, the vehicle requires a short-range communication link to other vehicles and the base station.

2.3.2 Requirements on Localization

Accurate and robust robot localization is a key capability for autonomous mobile vehicles. Regarding the localization scheme, we aim for an embedded localization system that can be implemented onboard the μAUV. This implies that its size and computational complexity can be handled by the μAUV-platform. An important requirement is that such a system should not rely on an external link. These are seen as too unreliable to allow for continuous localization. The localization principle is ideally passive, meaning that the μAUV has a passive receiving sensor on-board, e. g. a camera. This facilitates a roll-out to multiple robot settings and reduces the onboard energy consumption. For more details on technical considerations, we refer to Section 4.1.

2.3.3 Requirements on Field Exploration Methods

Due to the limited availability of communication bandwidth, the desired field exploration method should be designed for low computational complexity. In particular, the computational load must not increase with the number of collected measurements. The exploration algorithm should be adaptive to the investigated field. This allows mission goals such as source seeking. Furthermore, the developed method should allow an extension to multi-robot missions. These settings demand for decentralized solutions and efficient communication strategies.

2.4 Survey on Related Works

The following survey aims to provide a broad overview of the relevant state-of-the-art in the research fields of field exploration, localization, and micro underwater robot systems. For the sake of brevity at this point, we put our focus on works with a connection to the marine domain. The in-depth technical discussion of relevant technologies and interdisciplinary methods is left to corresponding main chapters as these provide the reader with a better context and background.

2.4.1 Small-Scale Underwater Robotic Platforms

This dissertation addresses the development of field exploration methods for μAUVs. Thus, real-world deployment and testing is a critical step within the development process of robot algorithms. In particular, it is important to validate theoretical methods and simulations in hardware experiments. While many influences can be considered already in the simulation stage, the transfer to field experiments reveals how valid the previously taken assumptions are. Moreover, hardware implementations allow studying whether sensor noise, model uncertainties, and the bandwidth of the communication link have been sufficiently considered within the theoretic model.

Underwater robot vehicles can be grouped into major categories, AUVs and ROVs. By definition, ROV platforms possess a communication link to some kind of base station from which a human operator sends control commands to the robot. The majority of ROV systems rely on tethered communication. Recently, modems relying on optical communication links have become available, while acoustic telemetry systems are usually unsuitable due to latencies and limited bandwidth. In contrast, AUVs systems operate autonomously by definition, thus without human supervision.

For the following, our survey focuses on the subgroup of μAUVs which is defined by their length-scale being smaller than 50 cm. For the sake of completion, we also provide a coarse overview on platforms with slightly larger length scales. When reviewing the market for micro underwater robots, we observe, that commercial platforms are barely available and mostly limited to ROV platforms. In contrast, today's μAUV systems are mostly available in an academic research context. Known μAUVs are designed as special purpose vehicles optimized towards their dedicated application scenarios. Application examples are manifold. The AVEXIS [43] robot design targets the monitoring in complex and hazardous environments such as nuclear storage ponds. A design challenge arises from its targeted deployment through small inspection wholes. The AUV-X platform [47] is designed for space missions to explore under-ice oceans on Jupiter's moon Europa.

In recent years, the development of μAUV platforms strongly profited from the remarkable progress in the miniaturization of electronic components. This enabled the integration of matured off-the-shelf components such as single-board computers. The performance boost in embedded computing platforms, e. g. the Raspberry Pi family, enables today's μAUV platforms to run comparatively complex software algorithms when compared with small underwater robot systems presented in the 2000s. This allowed for a drastic reduction in size and cost. Furthermore, progress in 3d-printing contributed to the flexibility in design and reduced the vehicle cost.

From a dynamic perspective, most μAUVs are torpedo-shaped. They are actuated by a single propeller thruster at their aft and rudder fins for attitude control. A recent trend towards multi-thruster platforms can be observed. This means that the thrusters act directly on the individual degree of freedoms (DOFs)s which considerably increases maneuverability. Depending on the thruster configuration, it allows the robot to hover at a single spot. Most platforms rely on propeller thrusters. However, various bio-inspired approaches exist, e. g. fish-like robots which are solely propelled by their fins [17, 66].

An interesting recent development is the use of semi-professional commercial hardware components. Prominent examples are ready-to-use robot thruster units and sealed housing tubes which are rated to defined water depths. Both were originally designed by the company BlueRobotics for their remote-controlled BlueRov-platform. An increasing number of groups uses these parts to build their custom robot platforms. Recent vehicle designs such as the LoCO-AUV [34] constitute the next step in this direction. However, currently, these components are too large for most micro vehicles designs. The commercial BlueRov2-thrusters have a diameter of roughly 10 cm. For comparison, the propellers of μAUVs lie in diameter ranges of 2-4 cm. The LoCO-AUV and the MONSUN-platform have length scales slightly above the upper bound of μAUVs. From a design perspective, they are sufficiently robust for harbor and open water applications. The same yields for the L-AUV [18]. However, with regard to maneuvering capabilities, these approaches show limited suitability for confined volumes. Moreover, their robust design constitutes a considerable cost-driver within the system's bill of materials. On the other end, smaller light-weight vehicle designs have become available targeting strictly confined volumes. For instance, the AVEXIS [43], JEFF [82] the AUV-X [47, 48]. However, these vehicles are currently limited with respect to their marching speed and agile maneuvering capabilities. Hackbarth et al. presented in [46] an early prototype of the HippoCampus μAUV-platform which is indeed able of agile maneuvering. The platform is fully revised in [22, 27] towards a hydrobatic field exploration platform. The newest version of the HippoCampus μAUV is used throughout this dissertation.

2.4.2 Localization Approaches for μAUVs

Current research on localization systems for μAUVs can be split into two approaches.

Vehicles that are temporarily tethered can outsource major shares of the localization and control processing to an external computing unit which enables them to run complex algorithms. Recent work on these vehicles has shown great steps forward. Robust model predictive control in the presence of currents was presented in [53] while [3] demonstrated trajectory tracking under model uncertainties. However, due to their tether, these vehicles can be categorized as *semi-autonomous*, as their tether limits their freedom in path and motion planning. This yields especially for obstacle-rich domains.

In contrast, tether-free vehicles are fully autonomous but have to rely on their onboard hardware. Literature [27, 43, 46, 47] shows a clear trade-off between a small vehicle size which offers a wide range of mission scenarios while it limits the capabilities of onboard sensors and computational power. An example of these restrictions is the non-availability of high-fidelity sensors in μAUVs such as ring-laser gyroscopes which are essential for dead reckoning-based navigation as it is widely used in medium and full-size AUVs [61, 62].

In contrast, state estimation systems based on small-size MEMS-sensors are widely used on μAUVs platforms. Examples include the AVEXIS [43], the AUVx [47], and the HippoCampus [27, 46] underwater robots. However, due to the strong drift of dead reckoning when using MEMS-IMU data, μAUV navigation requires additional robust and continuous information on the vehicle's absolute position.

Accurate and robust onboard self-localization of μAUVs is required for fully autonomous operation. Moreover, modular concepts of onboard localization systems allow a quick replacement depending on the scenario in which the robot is deployed.

External motion capture systems have been used to provide an accurate localization and even control commands to the underwater robot [17, 66, 88, 114]. However, they require an underwater communication link to the robot which suffers from high latencies and limited bandwidth and, thus, their usage is restricted to limited scenarios. As external systems do not scale with increasing vehicle fleet size, robust and accurate underwater localization can still be seen as widely unsolved.

Localization systems and their challenges can be grouped following their physical principle into vision, acoustics, and electromagnetic waves [19].

Vision-based approaches use cameras to perceive the robot's environment. They aim to estimate the robot pose relative to detected features [56]. Concerning underwater applications, their usage is mostly limited to short-range detection and clear water scenarios,

as they require good visibility conditions. Artificial landmarks are a widely used concept in localization, as they can be used to enrich featureless environments such as halls and tanks. However, they require the effort of pre-mission deployment. This makes their usage attractive for controlled environments such as research testbeds. Classic approaches are based on illuminated markers and require the detection of a marker pattern to compute the robot's pose. Recently, powerful libraries such as the AprilTag marker system [93, 130] became available and are by now a standard method in robotics for pose estimation. The markers are uniquely distinguishable and come with the advantage of simultaneously providing information on position and orientation relative to the camera [62]. Studies on the robustness of fiducial marker detection in various water conditions were extensively conducted in [21] and demonstrated the general suitability for underwater applications, e. g. for insertion tasks [52]. Moreover, simultaneous localization and mapping (SLAM) techniques based on artificial markers were recently applied in underwater domains and have shown promising results [61, 134]. However, known concepts mainly cover comparatively slow motions as these are dominant in docking and manipulation tasks. Their extension to hydrobatic maneuvering, which is a common requirement in μAUV missions, remains an open field of research.

Self-localization systems based on acoustics are widely used for AUVs in open-sea scenarios. They typically use the signal's time-of-flight or time-difference-of-arrival to compute the vehicle's positions relative to the sound signal emitting beacons. Recent work on small-size acoustic modems made this technology available for μAUVs [51, 117]. While acceptable performance has been demonstrated in harbors and large tanks, missions that require precise positioning such as docking or trajectory tracking in e. g small research tanks remain out of today's capabilities. This is due to the fact that acoustic methods suffer massively from reflections caused by e. g. tank walls, reverberations, and multi-path effects [42].

The use of attenuation of electromagnetic carrier waves for accurate short-range underwater localization was recently proposed in [96, 100]. This concept seems to be promising for confined volumes. The concept was miniaturized in [25] to fit the demands of μAUVs with respect to size and cost. Experiments using electromagnetic localization for μAUV control in a small research tank have been conducted and analyzed in [24]. The tests demonstrated the general feasibility but also revealed the method's sensitivity to electromagnetic wave reflections from the tank walls which limits deployment scenarios.

2.4.3 Field Exploration with Underwater Robots

Methods for spatio-temporal field exploration include two components, namely IPP and environmental field modeling. Field modeling aims for a meaningful spatial representation of the phenomena, e. g. a pollution field, based on measurements from a sensor network. In mobile sensor networks, an IPP algorithm guides the robots to regions of interest to optimize the information gain during field sampling. Environmental exploration constitutes an active and emerging field of research among almost all mobile robotic domains. In fact, autonomous exploration is a key capability for robots operating in partially or completely unstructured environments. Thus, it presents a direct link from classical feedback control to information-theoretic control.

Methods for IPP and field modeling are often combined in a modular manner. Thus, we review both components separately in the following. Note that we bias our survey to methods that have been applied to the marine domain. However, we point out important contributions from other domains as well.

Informative Path Planning Methods

In a generalized formulation, the optimization of information gathering can be framed as a sensor placement problem that is known to be complex to solve. The problem of optimal sensor placement is NP-hard, as Krause et al. discuss in their study on near-optimal sensor placement [67].

For the case of a mobile sensor node, the sensor placement problem can be generalized to IPP. A simple description – non-exclusive – goes as follows, one or multiple robot agents perform local control actions according to an information-theoretic control law to gather information on their environment. This task is also referred to as *active sensing*.

Informative path planning tasks include a wide range of subcategories and their flavors. Prominent examples include extremum seeking [44] (e. g. a pollution source) and coverage control [115].

The effectiveness of these techniques can be considerably leveraged by sharing the task among multiple robot agents. From a control perspective, this includes collective motion control and multi-robot coordination. A prominent and challenging example is sampling data in an oceanic scenario which has been addressed by various publications in the last years. Collective motion stabilization in a flow field is studied in [94, 95] on the example of environmental monitoring.

Multi-robot concepts can be categorized into approaches that possess a central coordinating unit and decentralized techniques that do not. In centralized settings, the coordinating unit – often referred to as base station – is linked to each robot. This base station has usually access to considerable computing resources and can run complex algorithms. It gathers all incoming information and computes control signals for each robot. However, these approaches obviously suffer from an increasing number of robot agents as resources have to be shared among the robots. This yields in particular for the communication bandwidth which constitutes a severe bottleneck in underwater missions. Examples include experimental studies presented in [17] and [88].

Information gathering algorithms can be grouped into four categories, of which all come with their individual strengths and flaws. They are referred to as (i) receding horizon, (ii) myopic/greedy, (iii) dynamic programming, and (iv) sampling-based approaches

Receding horizon methods aim for optimality within their control horizon but do not provide guarantees outside their sight. Greedy approaches optimize for the next best action. They are usually lightweight and easy to implement. However, due to their greedy nature, they are prone to local minima. Examples include the concepts relying on Bayesian optimization (BO). To overcome the myopic nature of BO concepts, Marchant et al. [78] reformulate the exploration problem as a partially observable Markov decision process (POMDP) to profit from the look-ahead planning capabilities. POMDPs usually require the discretization of state and action space in order to be computationally tractable. Tree-based methods such as Monte-Carlo Tree Search (MCTS) [109] allow handling large POMDPs with continuous state and observation spaces. Recently, IPP approaches based on POMDPs and MCTS gained considerable attention. The results are various flavors of these combinations [2, 4, 12, 14]. Morere et al. [85] address the limitation of a discrete state and action space by a continuous belief tree search approach that uses dynamic action sampling. Overall, works using the POMDP formalism gained a considerable boost by the recent progress on efficient POMDP solvers [112, 140]. This renders POMDP approaches a promising direction for future robotic applications. Considering the limited computational resources onboard mobile robots, sampling-based approaches constitute an appealing alternative. Cui et al. [17] propose rapidly exploring random trees (RRTs) for monitoring with a fish-like swarm. Hollinger and Sukhatme propose in [54] a stochastic motion planner for information gathering which builds on RRTs and branch and bound [8]. Moreover, various techniques for rapid information gathering using sampling-based algorithms are studied in [55]. In addition, recent work approached the exploration task by using path integrals (PIs) [26, 68] which directly considers robot dynamics during planning.

Environmental Field Representations

Environmental field representations provide the information basis for IPP algorithms by storing previously collected field observations. They usually form a map of the environment which shall be explored or monitored by the robot.

Field representation can be distinguished between physics-based or non-physics-based models. A strong advantage of physics-based representations is their ability to allow extrapolation to the model vicinity. However, they require solving a set of partial differential equations (PDEs), e. g. Navier-Stokes equation [45]. This is computationally demanding and often challenging to realize on embedded robotic platforms. Furthermore, physics-based models require information boundary conditions which may be hard to obtain in real-world scenarios.

Boosted by the recent progress in data-driven methods, probabilistic field models have become a promising alternative. These field representations are often termed field belief to emphasize their stochastic properties. Within this category, Gaussian process (GP) regression constitutes a prominent field inference method. The concept of GP regression originates from geo-statistics where is it referred to as *Kriging* [69]. Field beliefs based on GPs are usually an appealing choice, as they naturally provide a statistically meaningful uncertainty measure. Furthermore, GPs provide a continuous belief representation. However, their real-world application is often hindered by the so-called big-n problem, which becomes critical in multidimensional dynamics processes. Various methods have been proposed to overcome this limitation. Prominent attempts include reducing the number of dimensions, truncation of observations [138], and simplifying the structure of the GP's covariance matrix [75]. From a practical point of view, GPs are particularly challenging in underwater scenarios. The reason is the limited communication bandwidth which usually prohibits offloading their computation to a powerful computer cluster. In fact, the low communication bandwidth enforces a decentralized belief representation onboard each robot agent. Recently, Lindgren et al. [75] showed that GPs can be approximated by Gaussian Markov random fields (GMRFs) with sufficient accuracy. The GMRF approximates the GP on a predefined lattice of random variables. By exploiting the spatial Markov property, GMRFs allow for a representation of the random field which is considerably lighter in terms of computational complexity. The suitability of GMRFs as belief representation for various configurations of mobile sensor networks is studied by Xu et al. [135, 136, 137]. This was exploited in a series of papers in the context of field modeling [26, 58, 68]. Field belief representations ideally possess a constant computational complexity. For the application of field exploration, this means that the effort for the inference process does not increase with an increasing number of collected measurements. Methods

considering this aspect of efficient field updates are discussed in [136, 137]. Furthermore, the effect of uncertain measurement locations is analyzed in [58]. An example of a mobile robot exploring a 3D corridor environment is presented in [129]. However, the application of GMRFs in IPP settings is still rare.

Deep Reinforcement Learning for Informative Path Planning

Recently, deep learning and specifically deep reinforcement learning (DRL) gained high attention in the field of robotic motion planning. However, so far, there are yet few techniques that apply these concepts to the task of IPP. The application domain of these works mostly focuses on unmanned aerial vehicles (UAVs). Due to the similarities between small-scale underwater robots and aerial drones, we briefly present an excerpt of these recent results. Prominent applications include coverage path planning. An example is the study [118] in which Q-learning is applied to UAV coverage path planning under varying power constraints. A multi UAV information gathering framework is presented in [127]. The authors use the popular asynchronous advantage actor critic (A3C) [84] framework and extend it to multiple robots. Viseras et al. [128] study wildfire monitoring with multiple UAVs controlled by a deep Q-learning algorithm. Rather than considering a static global map, Theile et al. [119] propose a combination of local and global maps as an input to their deep neural network.

2.5 Summary and Identified Research Gaps

This chapter presented a detailed problem analysis on aspects associated with the task of field exploration and monitoring with μAUVs in confined environments. A wide variety of scenarios was studied. Overall, we identified three reference scenarios that will be used for the remainder of this dissertation.

Based on the identified challenges arising from the exploratory task and the environmental conditions a survey of related work was conducted. We identified the following research gaps based on our literature survey.

Considering the requirements on underwater vehicles for field exploration scenarios, we observe that the available μAUV-design are mostly designed towards very specific applications. Moreover, available designs are mainly research prototypes not thus commercially available on market. Regarding our posed requirements, we observe that available designs come with a considerable lack with respect to their agile maneuvering capabilities. Furthermore, their marching speed is comparatively slow which limits their suitability

for field exploration tasks. As a result, a gap exists for small-scale platforms with agile maneuvering capabilities.

Accurate and robust localization systems for underwater robots constitute a severe bottleneck and hinder the future development of μAUVs. In summary, each localization concept comes with its specific advantages and disadvantages. However, when studying control strategies for fully autonomous hydrobatic maneuvering, existing approaches reach their limits. This is in particular the case for small-scale robotic systems. Thus, a bottleneck exists for all reference scenarios even for the potential simple scenario of testing the robot's autonomous capabilities in controlled environments such as research tanks.

Overall, the deployment of sophisticated field exploration algorithms in submerged scenarios is challenging and, thus, rare. Kemna et al. [64] provide a feasibility study for a GP-based adaptive sampling algorithm. Sampling-based methods such as RRTs have been successfully adapted to field exploration scenarios using surface [55] and underwater robots [17]. However, large numbers of samples result in expensive evaluations of the cost function. Thus, their deployment is often infeasible onboard embedded systems. This contradicts the idea of fully autonomous exploration with μAUVs and leaves a research gap for lightweight exploration methods.

Furthermore, the previously discussed works on DRL-based exploration constitute an appealing and new approach to IPP. However, they rely on deterministic field representations rather than a stochastic belief representation. As a consequence, their training process mainly aims to minimize the error between their field representation and the actual gathered measurement. In contrast, most traditional IPP algorithms consider the field belief uncertainty as a quality metric for their control decisions. Thus, the combination of a stochastic field belief representation seems to be a promising research direction.

Chapter 3 HippoCampus: A Hydrobatic μAUV Platform

Hydrobatic robots constitute a recently formed class of underactuated underwater vehicles with agile maneuvering capabilities. In this sense, hydrobatics have become the marine counterpart to aerobatics in aerial robotics [7]. Hydrobatic capabilities are desirable for robots which aim to be deployed in confined and complex environments such as nuclear storage ponds, industry tanks, marinas, and harbors.

This chapter introduces the HippoCampus X μAUV-platform which constitutes the first *micro* robot within the new class of hydrobatic underwater vehicles [22, 27]. HippoCampus X is the matured and thoroughly revised version of the early 2015 design concept [46]. See Figure 3.1 for a portray of important design milestones.

The developed system targets and fills the capability gaps identified in Subsection 2.3.1. The HippoCampus X platform possesses – given its size and cost range – unique dynamic and autonomous capabilities including

1. a modular software/hardware architecture for easy adaption to various scenarios,
2. a robust embedded localization and control system for hydrobatic maneuvering, and
3. a streamlined small-scale low-cost design based on off-the-shelf components.

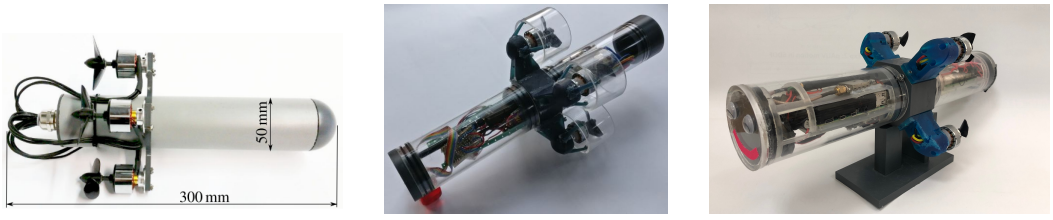


Figure 3.1: Milestones of the HippoCampus platform: concept as of 2015 (*left*) [46], intermediate design as of 2018 (*middle*) [27], and the matured HippoCampus X platform (*right*) [22].

The remainder of this chapter is structured as follows: First, we thoroughly discuss the μAUV system's architecture including the hardware and software considerations in Section 3.1. Second, we derive the vehicle model and its equations of motion including hydrostatic and dynamic effects in Section 3.2. Third, in Section 3.3 we develop a nonlinear geometric control scheme which builds on HippoCampus' dynamic property of being differentially flat. Finally, we present two experimental case studies in Section 3.4 to examine and discuss the capabilities of the developed HippoCampus platform with respect to its agile dynamics and the control performance.

3.1 System Architecture

HippoCampus' system architecture follows a strict modular approach with respect to software and hardware. This simplifies customization and opens the platform's capabilities towards a wide range of mission tasks. The goal is to easily add and exchange components depending on the current mission profile and its individual requirements. This demands for carefully chosen and well defined interfaces between software modules and hardware components.

The following gives a brief overview on the developed overall concept while the subsequent subsections explore and discuss the individual aspects in more detail.

On the hardware-side, the modular design is achieved by exchangeable 3D-printed mounting racks which host all hardware for individual modules, i. e. the visual self-localization module houses all required components for this task. This includes for instance camera, computing unit as well as spot lights if necessary. As a result, these module racks can then be easily exchanged depending on whether visual or acoustic localization is required.

On the software-side, we propose a modular GNC framework. This framework consists of three master modules covering high-level mission planning, vehicle self-localization, and low-level attitude control.

The high-level planning module covers both the target mission as well as transfer phases to which we refer to as *intra*-mission and *inter*-mission phases, respectively. Here, the term *intra* refers to the phase where the vehicle actively targets the mission goal, e. g. field exploration within a fluid volume. Complementary, *inter*-mission phases include all remaining phases starting from the initial vehicle launch, to approaching the mission area until the vehicle has finally reached its recovery docking area. Moreover, if the vehicle is assigned to multiple subsequent mission tasks, transfers are also covered by the *inter*-mission planner. Note that inter-mission phases are of special importance for autonomous

missions in real-world settings. Here, the vehicle usually cannot be launched directly at the target area of interest. Instead, the controlled mission begins at a remote docking area from which the vehicle approaches the target areas. However, even in controlled small-scale environments, such as test tanks in research facilities, planning and control during inter-mission phases remains necessary in order to cover idle-times, e. g. when multiple vehicles shall be deployed.

The vehicle's self-localization module includes various localization concepts. For their detailed discussion we refer the reader to Chapter 4.

From a control perspective, we propose a hierarchical approach consisting of two layers. First, the high-level planning algorithm outputs set-points to drive the vehicle along a given trajectory or path. Second, a low-level attitude controller computes control commands based on the current attitude and the given set-points. A mixer module then maps the control commands onto the actual motor actuators. As discussed above, the high-level planner is responsible for the successful mission execution. As a consequence, the planner's design and complexity is directly mission dependent. Its complexity ranges from simple path following during launching and transfer (inter-mission) to sophisticated environmental exploration tasks (intra-mission). In comparison to low-level control, high-level planning is often computationally expensive and requires a considerable share of the computing resources. On the upside, these high-level tasks usually do not require high update rates as opposed to the low-level controller. This makes it appealing to assign these tasks to individual and specialized computing units. Namely, high-level tasks run on single board computer (SBC) units and low-level time critical tasks are implemented directly on the flight computing unit (FCU), see Subsection 3.1.3 and Figure 3.2. Detailed software considerations are discussed in Subsection 3.1.2.

3.1.1 Structural Design

Missions in confined environments demand for agile maneuvering capabilities. This constitutes a key design criterion for HippoCampus' structural design and propulsion system. Figure 3.3 shows the final design of the HippoCampus μAUV.

Regarding the vehicle dimensions a small design is favorable. It enables the vehicle deployment through small inspection holes while also leveraging the dynamic agility through reduced inertia. The length of the HippoCampus X μAUV is 450 mm with its hull diameter being 70 mm. Its mass is trimmed to 1.5 kg in order to achieve neutral buoyancy. However, from a practical point of view, a slight positive buoyancy can be desirable in practice, as discussed in Subsection 3.2.3. The vehicle hull consists of a base unit and two

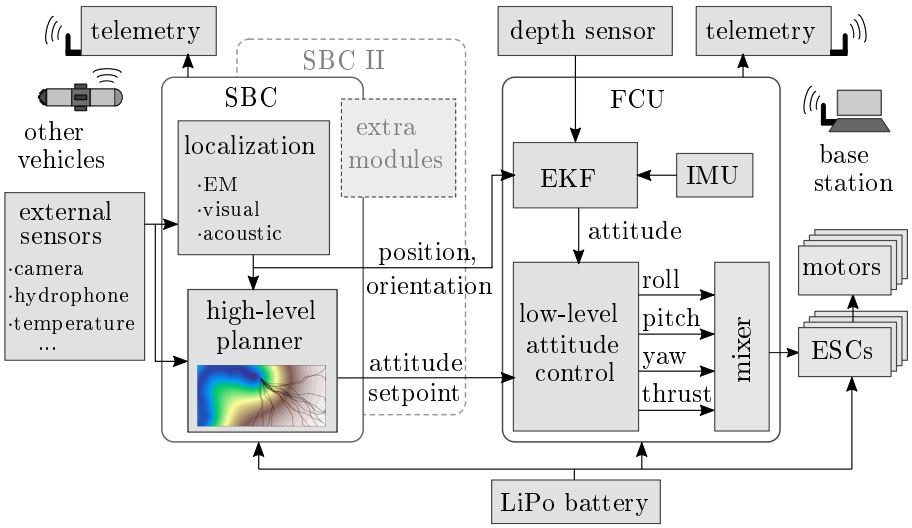


Figure 3.2: System architecture of the HippoCampus μAUV including the master modules of its guidance, navigation, and control framework: Vehicle self-localization is covered in Chapter 4. The low-level control-system is presented in Section 3.3, whereas Chapter 5 proposes a full framework for autonomous field exploration based on informative path planning.

acrylic tubes which house the electronic components. Depending of the individual mission requirements, the tube lengths can be adapted to accommodate additional hardware, e. g. sensors. The sealing between base unit and the acryl tubes is realized with O-rings. The vehicle has been designed to withstand water pressures at depths up to 10 m.

The strong requirements with respect to agility and the desired ability to turn the vehicle on spot without forward motion contradict traditional designs which combine fins and rudders with an aft-propeller. In fact, we favor a propulsion concept which is inspired by the quad-rotor design of aerial drones. This multi-rotor setup allows to control turning rates independent of the vehicles forward motion which is desirable for missions in confined, small volume tanks. The propulsion system consists of four thruster units mounted to the vehicles base unit in an X-shaped configuration, as depicted in Figure 3.4. Each thruster unit consists of a brushless DC motor and a carbon fiber propeller mounted to a 3D-printed gondola. The current vehicle design uses unsealed *Turnigy Aerodrive DST-700kv* motors which connect to the base unit with a simple plug-in connector, depicted in Figure 3.4. This design facilitates maintenance since the thruster unit can be replaced as a whole while the vehicle hull remains closed and waterproofed. Each motor is controlled by an *Afro 20A Race mini Opto* electronic speed controller which is housed within the base unit. In order to decouple the control input from the motors' high power stream, we choose the *opto*-electronic connection between both system. This combination of motor and speed controller achieves a theoretical maximum power of 150 W per thruster unit.

Cooling can become a performance bottleneck during missions with computational heavy tasks. This is especially the case if multiple computing modules such as the *RaspberryPi 4 B* are used. For these scenarios, the regular base unit can be replaced by an aluminum version which supports the heat transfer to the surrounding water environment.

The vehicle's power supply consists of a 3-cell 2,200 mAh LiPo-battery which allows for operational times of up to one hour. The LiPo-battery directly powers the vehicle's 12 V circuit while low-voltage components (5 V), e. g. computing modules, are powered via BEC-voltage converters. Hence, only a single onboard battery is required. Note that the electromagnetic compatibility of the individual electronic components has to be ensured in order to achieve a high reliability of integrated sensors and the overall vehicle.

Within the vehicle hull, 3D-printed racks provide defined mounting positions for all inside components. This conveniently allows a module-wise pre-assembly and replacement, e. g. of mission-specific self-localization modules, presented in Chapter 4. Moreover, it facilitates maintenance and system calibration.

Note that the overall design is aimed to be low-cost which additionally enables low maintenance effort by using off-the-shelf components whenever possible. This is found an important advantage during experimental trials as it minimizes setup times.

3.1.2 Software-Design

The software architecture follows as strictly modular design philosophy. Its vertical axis describes increasing layer of abstraction while additional functionality is covered by modules on the horizontal axis, see Figure 3.5. Note, the requirement of real-time executions varies between individual software processes. With an increasing level of abstraction we observe relaxed requirements of the processes running at high rates with low latency. At the same time high-level tasks are usually more computationally demanding. For in-

Figure 3.3: Final designs of the HippoCampus platform: regular design (*left*) and robust aluminum version for outdoor deployment (*right*).

Figure 3.4: X-shaped thruster configuration (*left*) and plug-in connection of thruster unit with base module (*right*).

stance, inertial sensing data and actuator control signals have to be processed at a high update rate to enable smooth thrust trajectories. In contrast, it may be sufficient to run the high-level mission planning module with a low rate of a few cycles per second. Still, this planning usually includes complex inference processes and thus requires considerable computational resources.

We account for this wide range of requirements by assigning the individual software modules to specialized computing units which address their needs accordingly. Note that all processing has to be performed on-board the vehicle as the limited external communication links do not allow to offload computation tasks. Thus, we assign the processes to either a real-time capable FCU or a powerful SBC.

On the first and lowest layer, the FCU runs the *NuttX* real-time operating systems. NuttX is a deeply embedded lightweight framework with direct interfaces to the hardware components such as the actuator signal generator for the actuators and sensors [79]. NuttX follows general POSIX-standards and provides various applications interfaces (APIs) to Unix environments.

The second layer is organized via the PX4-middleware [79] and its micro Object Request Broker (uORB). The uORB systems allows for asynchronous inter-process communication using a *publish-and-subscribe* structure. Consider, an individual software process which wants to share its information with another process. A publisher advertises this information via a semantic message channel referred to as *topic*, e. g. the `position` topic. Another process *subscribes* to this topic, asking for new information at its own desired rate, or can be woken up if new information is available. Within this layer, applications such as the controller or state estimators run as stand-alone processes. However, they can exchange information at high rates via the uORB system. Moreover, this layer provides off-board

communication via the MAVLink protocol to either a fellow companion-computer or to external platforms e. g. other vehicles or a base station.

As a third-layer, the HippoC-firmware stack provides fundamental extensions and functionality for unmanned underwater vehicles in general and in particular for the HippoCampus μAUV-platform. This includes the low-level geometric attitude controller (Section 3.3) as well as various extensions for state estimation. The general modular design greatly facilitates the transfer of control concepts to other underwater robotic platforms, such as the BlueROV platform. Moreover, the developed HippoC-firmware has become part of the official PX4-framework by contributing underwater self-localization and control capabilities [101].

Tasks which are expected to require large computational resources are assigned to the SBC units. These units run the open-source Robot Operating System (ROS) in combination with Linux Ubuntu. In recent years, ROS has become the standard tool in robotics research as it provides well defined interfaces between software modules [102]. Similar to the PX4-middleware, ROS uses a publish-and-subscribe design with stand-alone software. With regard to the HippoCampus platform, key modules such as the localization system and the high-level field exploration path planner are integrated via the ROS environments. Moreover, ROS comes with a wide range of analysis and debug tools which make its integration appealing. Additionally, ROS allows to conveniently integrate open-source third party software and contribute AUV-specific software to the open-source community.

3.1.3 Computational Resources

The computing units onboard the HippoCampus μAUV can be grouped into two categories. First, the FCU and, second, one or multiple SBCs which run individually assigned processes as discussed in Subsection 3.1.2. Both, the FCU and SBC units exchange data via a high bandwidth serial UART Mavlink/MavROS-interface. Additionally, the SBC units share a Gigabit Ethernet link.

Flight Controller In its current version the HippoCampus hosts a *PX4 PixRacer* FCU which is a light weight flight controller originally designed for racing quadcopters. It is chosen as computing unit for the time critical high-rate state estimation and the low-level controller. Both are implemented within the open-source drone firmware PX4 [79]. The Pixracer board hosts a 180 MHz ARM Cortex M4 CPU and provides 2 MB flash. It features an on-board gyroscope, accelerometer, and magnetometer.

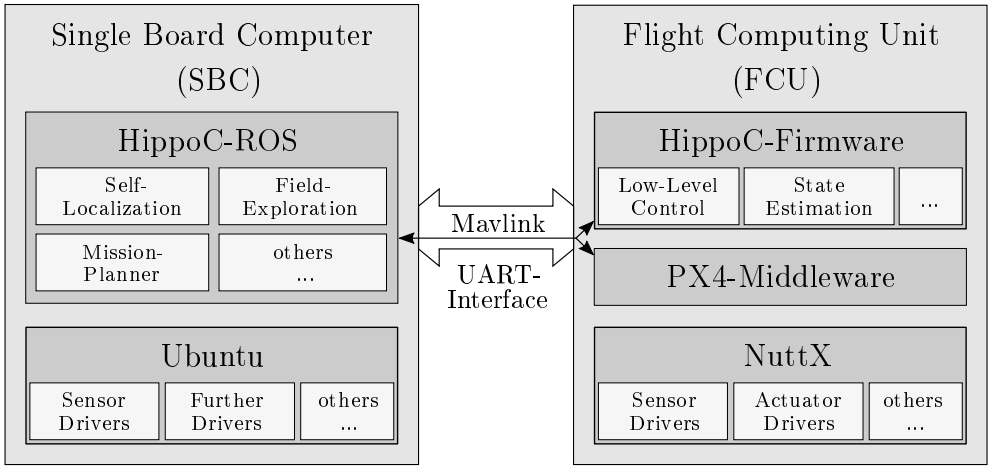


Figure 3.5: Software-Architecture with multiple layers of abstraction facilitating vertical and horizontal extensions.

Companion Computer Depending on the mission and the expected computational requirements up to two *RaspberryPi 4B* SBCs with 8 GB RAM are deployed onboard. Within the HippoCampus software architecture complex algorithms which run at lower rates (<30 Hz) such as the high-level exploration planner or vision processing tasks are implemented on SBC modules. Onboard the SBC, these algorithms are implemented in `Python` and `C++` within the ROS environment which ensures a modular design with well defined interfaces between the algorithms. If high performance computing such as complex image processing and neural network inference is required, the RaspberryPi SBC can be replaced by a *Nvidia Jetson TX2* SBC which features a dedicated graphic processing unit for efficient parallel computing.

3.1.4 Sensors and Electronic Components

The modular concept of the HippoCampus platform aims to be deployed within various scenarios conducting individual mission tasks. This requires mission dependent integration of a wide range of sensory equipment. For this purpose HippoCampus features various standardized interfaces including USB 3.0, SPI, and I2C, which allow for a straight-forward integration of additional hardware. In the following, we provide an overview on available onboard hardware components for sensing and communication.

Environment Perception The HippoCampus Platform is designed for environmental exploration tasks. It features various sensors to perceive its surrounding environment. The basic setup features two mono-cameras mounted in forward- and downward-looking directions. While the downward-looking camera is mainly used for visual localization

and monitoring purposes, the forward-looking camera is required for obstacle tracking and avoidance. With regard to environmental field exploration HippoCampus carries a temperature sensor in its basic configuration. Depending on the task the sensor suite can be supplemented with sensors measuring salinity, chemical concentrations, and radiation fields. However, also complex off-the-shelf sensors such as a RGB-D depth camera can easily be added to the HippoCampus platform. The reader is referred to [28] for an experimental case study on autonomous aquaculture inspection using the HippoCampus μAUV in combination with an *Intel Realsense D435i* RGB-D camera.

Vehicle Attitude Sensing It is a key capability of autonomous vehicles to gain knowledge on their current attitude. Thus, the HippoCampus μAUV is equipped with multiple 6- and 9-axes inertial measurement units (IMUs) which provide high-rate data for vehicle state estimation. These IMUs include accelerometers, gyroscopes, and magnetometers which provide measurements for all three axis, respectively. These sensors are usually made out of micro-electro-mechanical systems (MEMS). Note that MEMS-gyroscope measurements are prone to a strong temperature dependent drift which is estimated over time and hence compensated. Moreover, the vehicle carries a pressure sensor in order to gain information on the vehicles absolute diving depth.

3.1.5 Communication

While powerful communication links are available for terrestrial and aerial robots their absence constitutes a challenging bottleneck for tether-less underwater vehicles. Moreover, with regard to communication we face a fundamental trade-off between high bandwidth and range. In order to cope with this, the HippoCampus μAUV features a total of three different tether-less methods to communicate with its surrounding.

Firstly, a high-speed 2.4/5 GHz WiFi-link provides full access to the HippoCampus software framework, e. g. for flashing the firmware in case the vehicle is surfaced. This WiFi-link is used to synchronize large log-data files which are recorded on SD-cards while the mission is executed.

Secondly, the μAUV carries two telemetry modules operating within the commercial 433 MHz-band at a Baudrate of 57,600 with 100 mW for short range underwater communication. These telemetry modules are used for inter-vehicle communication (*V2V*) and data exchange between the vehicle and its base-station (*V2B*) allowing live-monitoring of the system's status, e. g. plotting the vehicle position. The telemetry link uses the lightweight MAVLINK protocol which is robust and widely used for aerial drone commu-

nication over long distances. It is worth pointing out that electromagnetic communication in general is barely used in underwater robotic application as it suffers from strong signal attenuation in water. This usually limits its reliable communication range to only a few meters. However, this range is often sufficient for most confined scenarios. During testing the telemetry modules achieved a reliable communication range of approximately 5 m in fresh water.

Thirdly, the *ahoi* micro acoustic modem in combination with a hydrophone developed by Renner et al. [103] can be integrated if longer communication ranges are required. The ahoi-modem has been redesigned in order to fit the dimension requirements onboard HippoCampus μAUV. Experimental trials in the Hamburg Finkenwerder marina showed reliable communication with a range of up to 150 m [103]. However, the achieved communication bandwidth is considerably lower in comparison to the telemetry link.

Recently, light-based communication methods based on LED-modems gained attention within the underwater robotics community [111]. While the modem dimensions not yet fit with μAUV requirements, first results showed promising MBit/s bandwidths over ranges of multiple meters. However, these light-based approaches naturally require line-of-sight between receiver and transceiver. Additionally, they strongly suffer from turbidity.

3.2 Modeling

3.2.1 Reference Frames

In order to describe the 6 DOF motion of an underwater vehicle we define two main geographic reference frames.

World-fixed Frame $\mathcal{W}$ Following the standard notation for marine crafts introduced by Fossen [37] it is common to define a *North-East-Down* (NED) coordinate system with its origin $O_{\mathcal{W}}$ relative to the earth's reference ellipsoid (World Geodetic System, 1984). Its axes are defined through the unit vectors $\boldsymbol{x}_{\mathcal{N}}$, $\boldsymbol{y}_{\mathcal{N}}$, and $\boldsymbol{z}_{\mathcal{N}}$ pointing towards true north, east, and in downward direction, respectively. However, given our focus application field of confined environments, it is often convenient to use a more flexible definition of the inertial reference frame to which we refer to as world frame $\mathcal{W}$. We define this world frame by the unit vectors $\boldsymbol{x}_{\mathcal{W}}$, $\boldsymbol{y}_{\mathcal{W}}$, and $\boldsymbol{z}_{\mathcal{W}}$ and the origin $O_{\mathcal{W}}$. This coordinate system is specific to the facility the vehicle is deployed in. Using this facility specific reference frame facilitates vehicle deployment within restricted volumes. For instance in housed industry tanks where robust measurements of the true north/east directions are

not available. Hence, a convenient approach is to define the $x_{\mathcal{W}}$- and $y_{\mathcal{W}}$- axis along the volume dimensions, e. g. along the walls of a rectangular tank while the $z_{\mathcal{W}}$- axis points in downward direction. For the sake of simplicity, we use the world frame throughout the remainder of this dissertation. During outdoor experiments, however, our world frame definition coincides with the NED convention.

Body-fixed Frame $\mathcal{B}$ Moreover, we define a moving *body* frame $\mathcal{B}$ which is fixed to the vehicle of interest, i. e. the HippoCampus μAUV. By definition the body frame's origin $O_{\mathcal{B}}$ coincidences with the vehicle's center of gravity (CG). We choose its axes $x_{\mathcal{B}}$, $y_{\mathcal{B}}$, and $z_{\mathcal{B}}$ such that they align with the vehicles *principal axes of inertia*. For the design of the HippoCampus μAUV we define

- $x_{\mathcal{B}}$: longitudinal axis (directed from aft to fore)
- $y_{\mathcal{B}}$: transversal axis (directed to starboard)
- $z_{\mathcal{B}}$: normal axis (directed from top to bottom)

as depicted in Figure 3.6.

3.2.2 Vehicle Kinematics

In the following, we present the vehicle kinematics. We start with a definition of the vectorial notation which is followed by a description of the required coordinate frame transformations, for instance from the world frame $\mathcal{W}$ to the body frame $\mathcal{B}$.

Vectorial Notation As a convention we use calligraphic superscripts to denote the reference frame in which the quantity is expressed. Moreover, subscripts are used for additional information for instance to denote relative quantities, e. g. ${}^{\mathcal{I}}\boldsymbol{v}_{A/B}$ is the linear velocity vector of a point A with respect to a point B expressed in the coordinates of the reference frame $\mathcal{I}$. Accordingly, ${}^{\mathcal{I}}\boldsymbol{f}_B$ denotes a force acting on point B represented in the reference system $\mathcal{I}$. In terms of coordinate transformations our notation follows

$$ {}^{to}\boldsymbol{v} = {}^{to}\boldsymbol{R}_{from}\,{}^{from}\boldsymbol{v}, $$

to describe a transformation of a vector $\boldsymbol{v}$ *from* the original frame *to* the new frame by means of the transformation matrix ${}^{to}\boldsymbol{R}_{from}$ with

$$ {}^{to}\boldsymbol{R}_{from} = \begin{bmatrix} {}^{to}\boldsymbol{x}_{from} & {}^{to}\boldsymbol{y}_{from} & {}^{to}\boldsymbol{z}_{from} \end{bmatrix}, \tag{3.1} $$

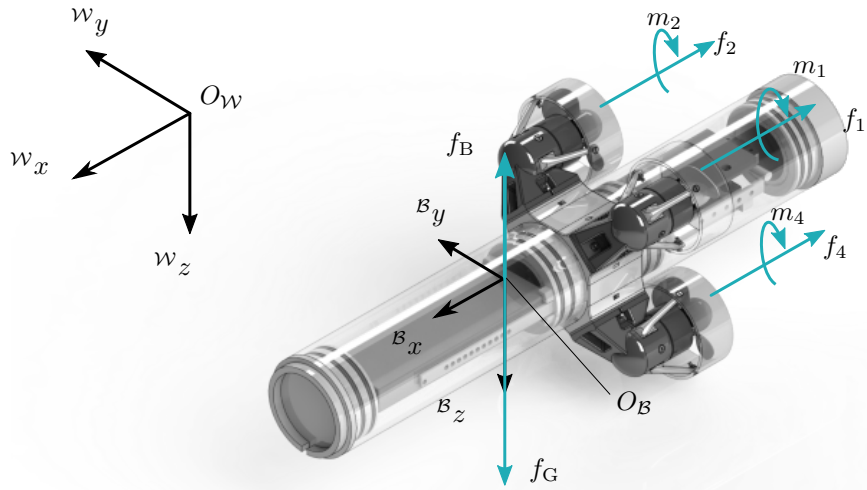


Figure 3.6: Definition of the world $\mathcal{W}$ and body $\mathcal{B}$ reference frames.

where ${}^{to}\boldsymbol{x}_{from}$, ${}^{to}\boldsymbol{y}_{from}$, and ${}^{to}\boldsymbol{z}_{from}$ denote the unit length vectors of the initial frame axes expressed in coordinates of the target frame. Moreover, we define the basis unit vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, and $\boldsymbol{e}_3$ as

$$\boldsymbol{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{3.2}$$

Coordinate Frame Transformations The vehicle's position $\mathrm{O}_\mathcal{B}$ with respect to the world frame origin $\mathrm{O}_\mathcal{W}$ written in world coordinates $\mathcal{W}$ is defined by

$${}^\mathcal{W}\boldsymbol{p}_{\mathrm{O}_\mathcal{B}\mathrm{O}_\mathcal{W}} = \begin{bmatrix} x & y & z \end{bmatrix}^\top \in \mathbb{R}^3, \tag{3.3}$$

where $\mathbb{R}^3$ is the *Euclidean space* of dimension three.

Following standard notation in guidance, navigation, and control applications, we use the z-y-x-convention for the transformation from the $\mathcal{W}$ to $\mathcal{B}$ frame specified by the Euler angles

$$\boldsymbol{\Theta}_{\mathcal{W}\mathcal{B}} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^\top \in \mathcal{S}^3, \tag{3.4}$$

where the set $\mathcal{S}^3$ is a *sphere* implying there are three angles defined on the interval $[0,2\pi]$.

Alternatively, the vehicle orientation can be described by means of the *unit quaternion* or *Euler parameter* representation. This representation allows to avoid singularities which are inherent to the Euler angle representation. However, Euler angles are favorable if

single degrees of freedom shall be analyzed e. g. the pitch motion. Let $\boldsymbol{q}$ be the quaternion describing the vehicle orientation, reading

$$\boldsymbol{q} = \begin{bmatrix} \epsilon_0 & \epsilon_1 & \epsilon_2 & \epsilon_3 \end{bmatrix}^\top \in \mathbb{R}^4, \tag{3.5}$$

where ϵ_0 is the quaternion's real part and $\epsilon_1, \epsilon_2, \epsilon_3$ describe the imaginary part.

The vehicle's body-fixed linear velocities in $x_\mathcal{B}$, $y_\mathcal{B}$, and $z_\mathcal{B}$ direction are also referred to as *surge*, *sway*, and *heave* motion. The linear velocity and the angular velocity vectors are defined with respect to world frame $\mathcal{W}$ and its origin $\mathrm{O}_\mathcal{W}$ and read

$${}^\mathcal{B}\boldsymbol{v}_{\mathrm{O}_\mathcal{B}/\mathrm{O}_\mathcal{W}} = \begin{bmatrix} u & v & w \end{bmatrix}^\top \quad \text{and} \quad {}^\mathcal{B}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}} = \begin{bmatrix} p & q & r \end{bmatrix}^\top, \tag{3.6}$$

respectively. Furthermore, the body-fixed forces and moments are denoted by

$${}^\mathcal{B}\boldsymbol{f}_{\mathrm{O}_\mathcal{B}} = \begin{bmatrix} X & Y & Z \end{bmatrix}^\top \quad \text{and} \quad {}^\mathcal{B}\boldsymbol{m}_{\mathrm{O}_\mathcal{B}} = \begin{bmatrix} K & M & N \end{bmatrix}^\top. \tag{3.7}$$

Hence, the general 6 DOF motion of a submerged vehicle with coordinate origin $\mathrm{O}_\mathcal{B}$ can be described by the following vectors

$$\boldsymbol{\eta} = \begin{bmatrix} {}^\mathcal{W}\boldsymbol{p}_{\mathrm{O}_\mathcal{B}/\mathrm{O}_\mathcal{W}} \\ \boldsymbol{\Theta}_{\mathcal{W}\mathcal{B}} \end{bmatrix}, \quad \boldsymbol{\nu} = \begin{bmatrix} {}^\mathcal{B}\boldsymbol{v}_{\mathrm{O}_\mathcal{B}/\mathrm{O}_\mathcal{W}} \\ {}^\mathcal{B}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}} \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} {}^\mathcal{B}\boldsymbol{f}_{\mathrm{O}_\mathcal{B}} \\ {}^\mathcal{B}\boldsymbol{m}_{\mathrm{O}_\mathcal{B}} \end{bmatrix}, \tag{3.8}$$

where $\boldsymbol{\eta} \in \mathbb{R}^3 \times \mathcal{S}^3$ is the vehicle's absolute position and orientation vector in world coordinates. Moreover, $\boldsymbol{\nu} \in \mathbb{R}^6$ denotes the linear and angular velocities written in body coordinates. Finally, $\boldsymbol{\tau}$ summarizes the forces and moments acting on the vehicle in the body-fixed reference frame. Note that for the sake of readability, we write subscripts in short-form in the following, i. e. $\boldsymbol{f}_\mathcal{B}$ rather than $\boldsymbol{f}_{\mathrm{O}_\mathcal{B}}$. However, we will indeed use their long forms whenever they seem helpful for understanding.

Hence, we can define the rotation matrix

$${}^\mathcal{W}\boldsymbol{R}_\mathcal{B}\left(\boldsymbol{\Theta}_{\mathcal{W}\mathcal{B}}\right) = \begin{bmatrix} \mathrm{c}\psi\mathrm{c}\theta & -\mathrm{s}\psi\mathrm{c}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi & \mathrm{s}\psi\mathrm{s}\phi + \mathrm{c}\psi\mathrm{c}\phi\mathrm{s}\theta \\ \mathrm{s}\psi\mathrm{c}\theta & \mathrm{c}\psi\mathrm{c}\phi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi & -\mathrm{c}\psi\mathrm{s}\phi + \mathrm{s}\theta\mathrm{s}\psi\mathrm{c}\phi \\ -\mathrm{s}\theta & \mathrm{c}\theta\mathrm{s}\phi & \mathrm{c}\theta\mathrm{c}\phi \end{bmatrix}. \tag{3.9}$$

Note, we use the short notation s· and c· representing $\sin(\cdot)$ and $\cos(\cdot)$, respectively. The above rotation is equivalent to the sequence ${}^\mathcal{W}\boldsymbol{R}_\mathcal{B} := \boldsymbol{R}_{z,\psi}\boldsymbol{R}_{y,\theta}\boldsymbol{R}_{x,\phi}$, where $\boldsymbol{R}_{\lambda,\beta}$ denotes the rotation matrix describing a rotation of β around the λ-axis.

The quaternion counterpart reads

$$\boldsymbol{R}_q\left(\boldsymbol{q}\right) = \begin{bmatrix} 1-2\left(\epsilon_2^2+\epsilon_3^2\right) & 2\left(\epsilon_1\epsilon_2-\epsilon_3\epsilon_0\right) & 2\left(\epsilon_1\epsilon_3+\epsilon_2\epsilon_0\right) \\ 2\left(\epsilon_1\epsilon_2+\epsilon_3\epsilon_0\right) & 1-2\left(\epsilon_1^2+\epsilon_3^2\right) & 2\left(\epsilon_2\epsilon_3-\epsilon_1\epsilon_0\right) \\ 2\left(\epsilon_1\epsilon_3-\epsilon_2\epsilon_0\right) & 2\left(\epsilon_2\epsilon_3+\epsilon_1\epsilon_0\right) & 1-2\left(\epsilon_1^2+\epsilon_2^2\right) \end{bmatrix}. \tag{3.10}$$

Moreover, we define the differential equation of the rotation matrix between the body- and world-reference frame as

$$^{\mathcal{W}}\dot{\boldsymbol{R}}_{\mathcal{B}} = {}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}}\,\boldsymbol{S}\left({}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}}\right), \tag{3.11}$$

where

$$\boldsymbol{S}\left({}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}}\right) = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}. \tag{3.12}$$

We can express the body-fixed linear velocity vector ${}^{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}/\mathcal{W}}$ in the world frame $\mathcal{W}$ as

$$^{\mathcal{W}}\dot{\boldsymbol{p}}_{\mathcal{B}/\mathcal{W}} = {}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}}\,{}^{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}/\mathcal{W}}, \tag{3.13}$$

with ${}^{\mathcal{W}}\dot{\boldsymbol{p}}_{\mathcal{B}/\mathcal{W}}$ being the linear velocity vector in world coordinates. Analogously, we obtain the Euler rate vector $\dot{\boldsymbol{\Theta}}_{\mathcal{WB}}$ from the body-fixed angular velocity ${}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}}$ through the transformation matrix $\boldsymbol{T}_\Theta\left(\Theta_{\mathcal{WB}}\right)$. Hence,

$$\dot{\boldsymbol{\Theta}}_{\mathcal{WB}} = \boldsymbol{T}_\Theta\left(\boldsymbol{\Theta}_{\mathcal{WB}}\right){}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}}, \tag{3.14}$$

where $\boldsymbol{T}_\Theta\left(\boldsymbol{\Theta}_{\mathcal{WB}}\right)$ reads

$$\boldsymbol{T}_\Theta\left(\boldsymbol{\Theta}_{\mathcal{WB}}\right) = \begin{bmatrix} 1 & \mathrm{s}\phi\mathrm{t}\theta & \mathrm{c}\phi\mathrm{t}\theta \\ 0 & \mathrm{c}\phi & -\mathrm{s}\phi \\ 0 & \mathrm{s}\phi/\mathrm{t}\theta & \mathrm{c}\phi/\mathrm{c}\theta \end{bmatrix}, \tag{3.15}$$

with s$\cdot$, c$\cdot$, and t$\cdot$ representing the trigonometric functions $\sin\left(\cdot\right)$, $\cos\left(\cdot\right)$, and $\tan\left(\cdot\right)$, respectively.

The alternative quaternion-based representation of the velocity transformation reads

$$\boldsymbol{T}_q(\boldsymbol{q}) = \frac{1}{2}\begin{bmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \epsilon_0 & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \epsilon_0 & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \epsilon_0 \end{bmatrix}, \tag{3.16}$$

which leads to

$$\dot{\boldsymbol{q}} = \boldsymbol{T}_q(\boldsymbol{q})\,{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}}. \tag{3.17}$$

Thus, we can write the full 6 DOF kinematic equations via Euler angles as

$$\begin{bmatrix} {}^{\mathcal{W}}\dot{\boldsymbol{p}}_{\mathcal{B}/\mathcal{W}} \\ \dot{\boldsymbol{\Theta}}_{\mathcal{W}\mathcal{B}} \end{bmatrix} = \underbrace{\begin{bmatrix} {}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}}(\boldsymbol{\Theta}_{\mathcal{W}\mathcal{B}}) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{T}_{\Theta}(\boldsymbol{\Theta}_{\mathcal{W}\mathcal{B}}) \end{bmatrix}}_{:=\boldsymbol{J}_{\Theta}(\boldsymbol{\eta})} \begin{bmatrix} {}^{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}/\mathcal{W}} \\ {}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}} \end{bmatrix}, \tag{3.18}$$

where $\boldsymbol{0}$ denotes a null-matrix of respective dimension. The equivalent short form of Equation (3.18) reads

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_{\Theta}(\boldsymbol{\eta})\,\boldsymbol{\nu}. \tag{3.19}$$

The corresponding quaternion-based representation of the full 6 DOF kinematic equations reads

$$\begin{bmatrix} {}^{\mathcal{W}}\dot{\boldsymbol{p}}_{\mathcal{B}/\mathcal{W}} \\ \dot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_q(\boldsymbol{q}) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{4\times3} & \boldsymbol{T}_q(\boldsymbol{q}) \end{bmatrix} \begin{bmatrix} {}^{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}/\mathcal{W}} \\ {}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}} \end{bmatrix}, \tag{3.20}$$

with $\boldsymbol{0}$ being the null-matrix of respective dimension.

3.2.3 Equations of Motion

In the following, we derive the nonlinear equations of motion for the HippoCampus μAUV based on the *Newton-Euler formalism* and Fossen's nomenclature [37]. The following derivations are displayed with respect to the vehicle's body-fixed reference frame $\mathcal{B}$ if not noted otherwise. Hence, we omit the corresponding frame super- and subscripts for the reader's convenience whenever possible.

The general rigid-body kinetics read

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\,\boldsymbol{\nu} = \boldsymbol{\tau}, \tag{3.21}$$

where $\boldsymbol{M}_{RB}$ is the rigid-body mass matrix, $\boldsymbol{C}_{RB}$ is the rigid-body Coriolis and centripetal matrix due to rotation of the body frame $\mathcal{B}$ with respect to the inertial world frame $\mathcal{W}$. Moreover, $\boldsymbol{\nu}$ denotes the generalized velocity and $\boldsymbol{\tau}$ denotes the vector of the generalized forces and moments acting on the vehicle, as defined in Equation (3.8). As discussed before, the origin of the body-fixed reference frame $\mathcal{B}$ to coincide with the vehicle's center of gravity (CG), as depicted in Figure 3.7. Its axes align with the vehicle's main axes as defined in Subsection 3.2.1. The rigid-body mass matrix reads

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m\boldsymbol{I}_{3\times 3} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J} \end{bmatrix}, \tag{3.22}$$

where m is the vehicle mass and $\boldsymbol{J}$ its inertia matrix with respect to the vehicle center of gravity. The inertia matrix $\boldsymbol{J}$ has the general symmetric form

$$\boldsymbol{J} = \begin{bmatrix} J_x & -J_{xy} & -J_{xz} \\ -J_{xy} & J_y & -J_{yz} \\ -J_{xz} & -J_{yz} & J_z \end{bmatrix}, \tag{3.23}$$

which reduces to diagonal form if the *principal axes of inertia* coincide with the body frame axes implying $\boldsymbol{J} = \operatorname{diag}(J_x, J_y, J_z)$. Furthermore, the rigid-body Coriolis and centripetal matrix reads

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} m\boldsymbol{S}\left(\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}}\right) & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{S}\left(\boldsymbol{J}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}}\right) \end{bmatrix}, \tag{3.24}$$

where $\boldsymbol{S}(\cdot)$ denotes the skew-symmetric matrix representing the vector cross-product. We define the vector cross-product operator $\times$ by

$$\boldsymbol{\lambda} \times \boldsymbol{a} := \boldsymbol{S}(\boldsymbol{\lambda})\,\boldsymbol{a} \tag{3.25}$$

with

$$\boldsymbol{S}(\boldsymbol{\lambda}) = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}. \tag{3.26}$$

If a body is submerged it is subject to hydrostatic and hydrodynamic effects which result from e. g. buoyancy and damping induced by the surrounding water. Moreover, we consider gravitational forces which act on the vehicle. Hence, we extend Equation (3.21) to

$$\boldsymbol{M}_{\mathrm{RB}}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{\mathrm{RB}}(\boldsymbol{\nu})\boldsymbol{\nu} + \underbrace{\boldsymbol{M}_{\mathrm{A}}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{\mathrm{A}}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}_{\mathrm{A}}(\boldsymbol{\nu})\boldsymbol{\nu}}_{\text{hydrodynamic loads}} + \underbrace{\boldsymbol{g}(\boldsymbol{\eta})}_{\text{hydrostatic load}} = \boldsymbol{\tau}. \tag{3.27}$$

The hydrodynamic load is accounted for by the added mass matrix $\boldsymbol{M}_{\mathrm{A}}$, the added Coriolis matrix $\boldsymbol{C}_{\mathrm{A}}$, and the added damping matrix $\boldsymbol{D}_{\mathrm{A}}$. The added mass and the added Coriolis matrix thereby represent effects caused by the motion of the surrounding fluid, whereas the added damping matrix considers hydrodynamic damping. The hydrostatic term $\boldsymbol{g}(\boldsymbol{\eta})$ captures buoyancy and gravity forces and their resulting induced moments acting on the vehicle. Note that this term depends explicitly on the vehicle pose in the world frame $\boldsymbol{\eta}$.

The full equations of motion of the HippoCampus μAUV read

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_{\Theta}\boldsymbol{\nu}, \tag{3.28}$$

$$\hat{\boldsymbol{M}}\dot{\boldsymbol{v}} = \boldsymbol{v} \times \hat{\boldsymbol{M}}\boldsymbol{\omega} - \boldsymbol{D}_{v,\mathrm{A}}\boldsymbol{v} - \boldsymbol{g}(\boldsymbol{\eta}) + u_1\boldsymbol{e}_1, \tag{3.29}$$

$$\hat{\boldsymbol{J}}\dot{\boldsymbol{\omega}} = \boldsymbol{v} \times \hat{\boldsymbol{M}}\boldsymbol{\omega} - \boldsymbol{\omega} \times \hat{\boldsymbol{J}}\boldsymbol{\omega} - \boldsymbol{D}_{\omega,\mathrm{A}}\boldsymbol{\omega} - \boldsymbol{g}(\boldsymbol{\eta}) + \begin{bmatrix} u_2 & u_3 & u_4 \end{bmatrix}^{\top}, \tag{3.30}$$

with

$$\hat{\boldsymbol{M}} = (m\boldsymbol{I} + \boldsymbol{M}_{v,\mathrm{A}}) \quad \text{and} \quad \hat{\boldsymbol{J}} = (\boldsymbol{J} + \boldsymbol{J}_{\mathrm{A}}),$$

where

$$\boldsymbol{M}_{\mathrm{A}} = \mathrm{diag}\,(\boldsymbol{M}_{v,\mathrm{A}}, \boldsymbol{J}_{\mathrm{A}}) \quad \text{and} \quad \boldsymbol{D}_{\mathrm{A}} = \mathrm{diag}\,(\boldsymbol{D}_{v,\mathrm{A}}, \boldsymbol{D}_{\omega,\mathrm{A}}).$$

Hydrostatic Load due to Restoring Forces

In hydrostatic terminology, the term *restoring forces* is widely used in order to describe effects induced by gravitational and buoyancy forces. These restoring forces constitute the hydrostatic pendant to spring forces in mass-damper-spring systems [37]. Their derivation distinguishes between surface and underwater vehicles. For our derivations of the restoring forces acting on the submerged HippoCampus platform we focus on the case of a fully submerged vehicle.

Consider a μAUV with mass m and a displaced fluid volume v. The vehicle is free floating in a water environment with density ρ and subject to a gravitational acceleration g. The

resulting gravity force $W = mg$ and buoyancy force $B = v\rho g$ lie in the vertical plane of the world-frame $\mathcal{W}$ with its $z_{\mathcal{W}}$-axis pointing downwards. The resulting force vectors read

$$ {}^{\mathcal{W}}\boldsymbol{f}_G = \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} \quad \text{and} \quad {}^{\mathcal{W}}\boldsymbol{f}_B = \begin{bmatrix} 0 \\ 0 \\ -B \end{bmatrix}. \tag{3.31} $$

Both force vectors act on the vehicle's CG and the center of buoyancy (CB), respectively. Notice that given the design of the HippoCampus we assume that both points CG and CB are fixed on the vehicle and, thus, fixed with respect to the vehicle body frame $\mathcal{B}$. Hence, their positions are defined by the vectors ${}^{\mathcal{B}}\boldsymbol{r}_G := [x_G\, y_G\, z_G]^\top$ and ${}^{\mathcal{B}}\boldsymbol{r}_B := [x_B\, y_B\, z_B]^\top$, respectively.

We write the hydrostatic load vector consisting of the restoring forces and moments as

$$ \boldsymbol{g}(\boldsymbol{\eta}) = - \begin{bmatrix} {}^{\mathcal{B}}\boldsymbol{R}_{\mathcal{W}}(\boldsymbol{\Theta}_{\mathcal{WB}}) \left({}^{\mathcal{W}}\boldsymbol{f}_G + {}^{\mathcal{W}}\boldsymbol{f}_B \right) \\ {}^{\mathcal{B}}\boldsymbol{r}_G \times {}^{\mathcal{B}}\boldsymbol{R}_{\mathcal{W}}(\boldsymbol{\Theta}_{\mathcal{WB}})\, {}^{\mathcal{W}}\boldsymbol{f}_G + {}^{\mathcal{B}}\boldsymbol{r}_B \times {}^{\mathcal{B}}\boldsymbol{R}_{\mathcal{W}}(\boldsymbol{\Theta}_{\mathcal{WB}})\, {}^{\mathcal{W}}\boldsymbol{f}_B \end{bmatrix}. \tag{3.32} $$

Notice the sign change, as $\boldsymbol{g}(\boldsymbol{\eta})$ is shifted to the left-hand side of Equation (3.27). However, Equation (3.32) can be simplified using our previous definition of the body-frame origin which coincides with CG resulting in ${}^{\mathcal{B}}\boldsymbol{r}_G := [0\,0\,0]^\top$. Moreover, the components of ${}^{\mathcal{B}}\boldsymbol{r}_B := [x_B\, y_B\, z_B]^\top$ are small since HippoCampus is deliberately designed to have CB and CG almost coincide.

For the HippoCampus μAUV, as with most submerged vehicles, a *neutrally buoyant* design is desirable regarding energy efficiency. Generally, neutral buoyancy is achieved if $W = B$. However, from a practical perspective a slight *positive buoyancy*, i. e. $B > W$, is preferable. In this case, the vehicle will automatically surface in emergency situations, e. g. due to power failures. Beside these hardware-side failures, this design choice also facilitates software-side emergency protocols. For instance, the vehicle safely surfaces when the actuators are stopped. In this context, the choice of positive buoyancy can be interpreted as a tuning knob to balance safety surfacing and energy-efficiency, since the vehicle energy consumption increases if $B \gg W$.

Hydrodynamic Loads

Marine vessels are subject to hydrodynamic loads in the presence of motion. This is addressed by the two fields *maneuvering* and *seakeeping* theory. In this context, maneuvering refers to vessel motions in the absence of wave excitation, i. e. calm water conditions or operation below the wave affected zone. As a consequence, motion is only induced by

means of the vessel's actuators. In contrast, seakeeping theory analyzes the vehicle motion assuming the vessel is on constant course with constant speed while being subject to wave excitation.

In the following, we discuss the hydrodynamic loads which are the result of *added mass* and *added damping* effects. Note that this dissertation focuses on application scenarios in confined and submerged environments. Hence, without considerable limitations we can assume the absence of wave excitation and irrotational ocean currents. Following standard literature, such as Fossen [37], our scenario allows to approximate the usually wave frequency dependent added mass terms by constant parameters. This is often referred to as the zero-frequency assumption. As a consequence, potential damping resulting from fluid-memory effects is also neglected. However, the vehicle is subject to viscous damping effects which have to be included within the model.

For the corresponding derivations we follow the notation of Society of Naval Architects and Marine Engineers (SNAME) [74] in order to describe the hydrodynamic derivatives. For example, we write the hydrodynamic added mass force X along the body $x_{\mathcal{B}}$-axis due to an acceleration $\dot{u}$ along the body $x_{\mathcal{B}}$-axis as

$$X = -X_{\dot{u}}\,\dot{u} \quad \text{with} \quad X_{\dot{u}} := \frac{\partial X}{\partial \dot{u}}. \tag{3.33}$$

Various approaches exist to identify the hydrodynamic parameters. Concepts range from simple heuristic approaches to sophisticated and often time-consuming system identification processes using computational fluid dynamic (CFD) simulation and towing experiments. When considering low-cost platforms such as the HippoCampus μAUV it is important to find a suitable trade-off between the invested effort and the desired model accuracy which is required for simulation and model-based control schemes. The reader is referred to [31] for a study presenting a parameter identification procedure for the HippoCampus platform using towing experiments.

Added Mass In good approximation, the structural design of the HippoCampus μAUV possesses three symmetry planes. Hence, we apply the common assumption for non-high speed vehicles that the added mass matrix is dominated by its diagonal elements while the off-diagonal elements contribute little to the dynamics and are, thus, neglected. This is also beneficial from a practical point of view since the identification of off-diagonal elements from experiments is usually tedious and time-consuming. Thus, the simplified added mass matrix reads

$$\boldsymbol{M}_A = -\mathrm{diag}\left(X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}\right). \tag{3.34}$$

The corresponding skew-symmetric matrix $\boldsymbol{C}_A$ which represents the Coriolis and centripetal effects caused by the added mass can be expressed as

$$\boldsymbol{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix}. \tag{3.35}$$

Added Damping Various types of hydrodynamic damping exist and can have a considerable effect on the motion of a marine vessel. The main contributors include potential damping, skin friction, wave-drift damping, and damping due to vortex shedding. Following the previous discussion – including the no-wave assumption – the dominant damping effects result from skin friction and vortex shedding.

Damping due to skin friction results from the laminar and turbulent boundary layer depending on the vehicle velocity. However, skin friction is usually considered in a linear context and often neglected for high speed motion. Damping due to vortex shedding is the result from the shedding of vortex sheets at sharp edges of the vehicle hull. A simple but common approximation for this case reads

$$f(u) = -1/2\,\rho\,C_{\mathrm{D}}\,(R_n)\,A|u|u, \tag{3.36}$$

where u is the vessel's velocity, A represents the projected cross-sectional submerged area, ρ the water density, and $C_{\mathrm{D}}\,(R_n)$ the corresponding drag coefficient. Note that the drag coefficient depends on the Reynolds number R_n which is defined as

$$R_n = \frac{u\,D}{\nu}. \tag{3.37}$$

Here, D denotes the vehicle's characteristic length and ν the kinematic velocity coefficient.

Damping effects at higher velocity can be highly nonlinear and coupled. However, we use the approximation proposed in [37] suggesting the assumption of non-coupled motion. In contrast to other torpedo-shaped AUVs, the HippoCampus hosts its propulsion systems midships insead of the common design with propeller and fins at the vehicle's aft, as used in the LAUV system [18]. This midship configuration leads to three symmetry planes

and results in a diagonal form of the nonlinear added damping matrix $\boldsymbol{D}_A$. The matrix consists of the linear term $\boldsymbol{D}_{A,\text{lin}}(\boldsymbol{\nu})$ and the quadratic term $\boldsymbol{D}_{A,\text{quad}}(\boldsymbol{\nu})$ reading

$$\boldsymbol{D}_A(\boldsymbol{\nu}) = \boldsymbol{D}_{A,\text{lin}} + \boldsymbol{D}_{A,\text{quad}}(\boldsymbol{\nu}), \tag{3.38}$$

with

$$\boldsymbol{D}_{A,\text{lin}} = -\text{diag}(X_u, Y_v, Z_w, K_p, M_q, N_r) \quad \text{and} \tag{3.39}$$

$$\boldsymbol{D}_{A,\text{quad}}(\boldsymbol{\nu}) = -\text{diag}\left(X_{|u|u}|u|, Y_{|v|v}|v|, Z_{|w|w}|w|, K_{|p|p}|p|, M_{|q|q}|q|, N_{|r|r}|r|,\right). \tag{3.40}$$

Notice, quadratic damping terms are dominant for medium- to high-speed profiles. However, as emphasized in [37] and [18], ignoring linear terms may lead to an underestimate of the damping at low-speed profiles. This may result in falsely locally unstable models in simulation.

Actuator Modeling

The HippoCampus μAUV is driven by four screw propeller thrusters which apply forces and moments on the vehicle. The chosen thrusters have a high power to weight ratio and are commonly used to power agile aerial drones. Considering this and experience from practice, we see that the motor dynamics are fast in comparison to the μAUV rigid body dynamics and hydrodynamics. Thus, we follow the common assumption from aerial drones reading that the desired motor dynamic state can be achieved instantaneously [80].

Notice, marine propeller theory is a complex research field in its own and does not lie within the scope of this work. Hence, we refer the interested reader to two excellent works [49, 74] for more details on this interesting topic.

For this dissertation the screw propeller model presented by Newman [91] is applied. From a modeling point of view, a screw propeller consists of multiple hydrofoil-like lifting surfaces. These surfaces are mounted onto a shaft in a helicoidal setup and induce an axial thrust and torque when the shaft rotates. The resulting thrust depends on the propeller's geometrical parameters, e. g. its diameter d, and the shaft's angular velocity $n = f(u)$ which itself depends on the control input u. Moreover, we consider the vehicle velocity relative to the surrounding fluid U_{vel} which has a direct effect on the propeller blade's local angle-of-attack. This angle is the difference between the fluid's inflow angle and the pitch angle of the blade.

Each propeller thruster induces a force $\boldsymbol{f}_i$ and a torque $\boldsymbol{m}_i$ to which we refer as *control* force and moment. For his model, Newman [91] assumes no cavitation and a *non excessive* angle-of-attack. Thus, we can write the thrust and torque in non dimensional form depending only on the *advance ratio* $J = \frac{U_{\text{vel}}}{n\,d}$

$$\frac{F_i}{\rho n^2 d^4} = \kappa_{\text{F},i}\,(J) \quad \Rightarrow \quad F_i = \kappa_{\text{F},i}\,(J)\,\rho n^2 d^4, \tag{3.41}$$

$$\frac{M_i}{\rho n^2 d^5} = \kappa_{\text{M},i}\,(J) \quad \Rightarrow \quad M_i = \kappa_{\text{M},i}\,(J)\,\rho n^2 d^5, \tag{3.42}$$

where $\kappa_{F,i}(J)$ and $\kappa_{M,i}(J)$ denote the thruster coefficients.

The thrusters are mounted at the positions $\boldsymbol{r}_i$ parallel to the vehicle's x-axis, as depicted in Figure 3.4. Hence, each propeller induces a control force F_i in x-direction and a moment M_i around the x-axis. Thus, we obtain

$$\boldsymbol{\tau}_i = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{r}_i \times \boldsymbol{f}_i + \boldsymbol{m}_i \end{bmatrix} = \begin{bmatrix} F_i \\ 0 \\ 0 \\ 0 \\ r_{i,z} F_i \\ -r_{i,y} F_i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ M_i \\ 0 \\ 0 \end{bmatrix} \tag{3.43}$$

for the i-th thruster mounted at position $\boldsymbol{r}_i$. The sum of all four thrusters can be expressed in compact form reading

$$\boldsymbol{\tau} = \begin{bmatrix} X & Y & Z & K & M & N \end{bmatrix}^\top = \sum_{i=1}^{4} \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{r}_i \times \boldsymbol{f}_i + \boldsymbol{m}_i \end{bmatrix}. \tag{3.44}$$

However, implementation-wise, it is usually convenient to follow a pure matrix-vector multiplication notation which we state in the following for the sake of completeness. Hence, we summarize all control forces *and* moments in a vector $\tilde{\boldsymbol{f}}$ reading

$$\tilde{\boldsymbol{f}} = \boldsymbol{K}\,\boldsymbol{u}, \tag{3.45}$$

with

$$\boldsymbol{K} = \begin{bmatrix} F_1 & 0 & 0 & 0 \\ M_1 & 0 & 0 & 0 \\ 0 & F_2 & 0 & 0 \\ 0 & M_2 & 0 & 0 \\ 0 & 0 & F_3 & 0 \\ 0 & 0 & M_3 & 0 \\ 0 & 0 & 0 & F_4 \\ 0 & 0 & 0 & M_4 \end{bmatrix}, \tag{3.46}$$

being the force and moment coefficient matrix in standard linear form and $\boldsymbol{u} = [u_1, u_2, u_3, u_4]^\top$ the vector of all control inputs. However, following our models for thrust and torque we write

$$\boldsymbol{K}(\boldsymbol{u}) = \begin{bmatrix} F_1(u_1) & M_1(u_1) & F_2(u_2) & M_2(u_2) & F_3(u_3) & M_3(u_3) & F_4(u_4) & M_4(u_4) \end{bmatrix}^\top, \tag{3.47}$$

to account for the nonlinearity in the models defined by Equation (3.41) and Equation (3.42).

In order to determine the effective actuator forces and moments related to the control forces and moments we introduce the thrust configuration matrix $\boldsymbol{T}$. This matrix represents the geometrical configuration of the thrusters onboard the vehicle. Considering of the HippoCampus μAUV the configuration matrix $\boldsymbol{T}$ is constant since the thrusters are mounted at fixed positions and cannot be rotated, as in pod-drives on harbor tugs.

Hence, we can write the vector of actuator forces and moments on the vehicle

$$\boldsymbol{\tau} = \boldsymbol{T}\boldsymbol{K}(\boldsymbol{u}), \tag{3.48}$$

with the configuration matrix $\boldsymbol{T}$ reading

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ r_{1,z} & 0 & r_{2,z} & 0 & r_{3,z} & 0 & r_{4,z} & 0 \\ -r_{1,y} & 0 & -r_{2,y} & 0 & -r_{3,y} & 0 & -r_{4,y} & 0 \end{bmatrix}. \tag{3.49}$$

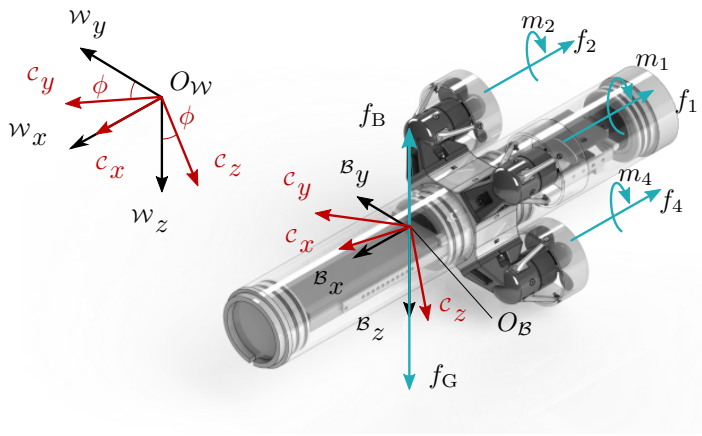


Figure 3.7: Free body diagram of the HippoCampus μAUV, including the buoyancy force f_B, the gravitational force f_G, and the controlled thruster forces and torques $f_{1:4}$ and $m_{1:4}$. The coordinate frames denote the world frame $\mathcal{W}$, the body-fixed frame $\mathcal{B}$, and the intermediate frame $\mathcal{C}$.

3.3 Control-Design

From the dynamic point of view, agile small-scale aerial vehicles such as quadrotors possess a variety of similarities with the HippoCampus μAUV design. Hence, it is appealing to transfer successful and powerful low-level control concepts from the aerial domain towards the submerged domain.

Thus, we firstly explore this dynamic similarity and show that, analogue to quadrotor drones, the HippoCampus dynamics are *differentially flat* in Subsection 3.3.1. Exploiting this property greatly facilitates trajectory planning tasks by reducing the number of states required to fully describe defined trajectories. Next, we present a light-weight low-level control concept which builds on prior work in agile aerial quadrotors [73, 80] in Subsection 3.3.2.

3.3.1 Differential Flatness

Differential flat systems constitute a group of dynamic systems which allow to express their states and inputs in terms of carefully chosen *flat outputs* and their derivatives. This results in an one-to-one correspondence between trajectories in the space of flat outputs and their representation in state and input space. The concept of differential flatness can be traced back to a series of papers by Fliess et al. [36]. The wide range of examples includes quadrotor drones, conventional aircraft, overhead cranes, cars with trailers, induction motors, and active magnetic bearings.

By exploiting differential flatness, trajectories can be directly planned in output space and then lifted to the state and input space using algebraic mapping [126]. As a result computationally light-weight algorithms for online trajectory generation have been developed [5, 86].

In the following, we show that the dynamics of the underactuated HippoCampus μAUV with its four inputs are indeed differentially flat. This is a nontrivial but very helpful insight given the non-negligible hydrodynamic effects in submerged environments. By definition, the dynamics of flat systems, namely its states and control inputs, can be fully described by so called flat outputs. Since the number of flat outputs is considerably smaller than the full state space of the vehicle, the flatness property reduces the complexity of planning and control tasks.

The similarity between aerial quadrotors and the HippoCampus dynamics motivates to build-on the quadrotor control approach described by Mellinger and Kumar [80]. In the following, we extend their technique to the underwater domain. Thereby, we point out important differences between aerial and submerged vehicles such as buoyancy and hydrodynamic loads. In order to show the differential flatness, we must express the 12 vehicle states and its four control inputs $u_{1:4}$ in terms of the self-chosen flat outputs and their derivatives. As a result, any smooth trajectory $\boldsymbol{\sigma}(t)$ defined in the space of flat outputs can be tracked by the underactuated μAUV given sufficiently bounded derivatives of $\boldsymbol{\sigma}$.

We choose the four flat outputs as

$$\boldsymbol{\sigma} = \begin{bmatrix} \boldsymbol{p}^\top & \phi \end{bmatrix}^\top = \begin{bmatrix} x & y & z & \phi \end{bmatrix}^\top, \tag{3.50}$$

where $\boldsymbol{p} = [x, y, z]^\top$ is the vehicle position, precisely its center of gravity, with respect to the world frame $\mathcal{W}$ and ϕ is the vehicle's roll angle.

In the following, we show that we can indeed express HippoCampus' translation, orientation, the corresponding velocities and accelerations as well as the net force and torque inputs purely in terms of the chosen flat outputs, Equation (3.50). This is equivalent to showing the differential flatness of the system [80].

Vehicle Position, Linear Velocity, and Acceleration Given the definition of the flat outputs in Equation (3.50) it is straight-forward to describe the vehicle's position, linear velocity, and acceleration in terms of the flat outputs and their derivatives reading $\boldsymbol{\sigma}$, $\dot{\boldsymbol{\sigma}}$, and $\ddot{\boldsymbol{\sigma}}$, respectively.

Vehicle Orientation The vehicle's absolute orientation, namely the rotation between the body-fixed frame and the world reference frame, is described by ${}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}}$. Hence, our goal is to represent this rotation in terms of the flat outputs $\boldsymbol{\sigma}$, Equation (3.50).

For this purpose, we introduce an intermediate coordinate frame $\mathcal{C}$ such that ${}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{C}}$ describes the vehicle's *roll* motion. The effect of *pitch* and *yaw* motions is left to ${}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{B}}$, reading

$$ {}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}} = {}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{C}}{}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{B}}. \tag{3.51} $$

The corresponding reference frames are depicted in Figure 3.7.

We can write the vehicle body-frame's $x_{\mathcal{B}}$-axis in $\mathcal{W}$ coordinates using Equation (3.29) and ${}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}} = {}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}}\boldsymbol{e}_1$

$$ \tilde{\boldsymbol{M}}\ddot{\boldsymbol{p}} + \tilde{\boldsymbol{D}}(\boldsymbol{\nu})\dot{\boldsymbol{p}} = u_1{}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}} \tag{3.52} $$

where

$$ \tilde{\boldsymbol{M}} = {}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}}\left(\boldsymbol{M}_{\mathrm{RB}} + \boldsymbol{M}_{\mathrm{A}}\right){}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}}^{\top} \quad \text{and} \quad \tilde{\boldsymbol{D}} = {}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}}\boldsymbol{D}_{v,\mathrm{A}}{}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}}^{\top}, $$

denote the corresponding mass, and damping matrices in the world reference frame. Note, we omit writing the explicit dependency on $\boldsymbol{\nu}$ in the following. Since our vehicle design aims for neutral buoyancy, we neglect the term $\boldsymbol{g}(\boldsymbol{\eta})$ in our subsequent derivations.

We can now express ${}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}$ in terms of the flat outputs, reading

$$ {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}} = 1/u_1\left(\tilde{\boldsymbol{M}}\ddot{\boldsymbol{p}} - \tilde{\boldsymbol{D}}\dot{\boldsymbol{p}}\right). \tag{3.53} $$

Moreover, the orientation of the intermediate frame $\mathcal{C}$ is defined by means of the roll angle ϕ which is part of the flat outputs, namely $\sigma_4 = \phi$. Thus, the $y_{\mathcal{C}}$-axis in $\mathcal{W}$ coordinates reads

$$ {}^{\mathcal{W}}\boldsymbol{y}_{\mathcal{C}} = \begin{bmatrix} 0 & \cos(\sigma_4) & \sin(\sigma_4) \end{bmatrix}^{\top}. \tag{3.54} $$

Thus, it follows

$$ {}^{\mathcal{W}}\boldsymbol{z}_{\mathcal{B}} = \frac{{}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}} \times {}^{\mathcal{W}}\boldsymbol{y}_{\mathcal{C}}}{||{}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}} \times {}^{\mathcal{W}}\boldsymbol{y}_{\mathcal{C}}||} \tag{3.55} $$

and

$$^{\mathcal{W}}\boldsymbol{y}_{\mathcal{B}} = {}^{\mathcal{W}}\boldsymbol{z}_{\mathcal{B}} \times {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}, \tag{3.56}$$

which yields given $\boldsymbol{x}_{\mathcal{B}} \times \boldsymbol{y}_{\mathcal{C}} \neq 0$. Thus, we obtain the orientation of the HippoCampus μAUV through $^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}}$ reading

$$^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}} = \begin{bmatrix} ^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}} & ^{\mathcal{W}}\boldsymbol{y}_{\mathcal{B}} & ^{\mathcal{W}}\boldsymbol{z}_{\mathcal{B}} \end{bmatrix}. \tag{3.57}$$

As all three columns of $^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}}$ can be expressed via the flat outputs, cf. Equation (3.50), we can represent the vehicle orientation in terms of the flat outputs. However, this yields, provided we do not encounter the singularity when both, the $x_{\mathcal{B}}$- and $y_{\mathcal{C}}$-axis are parallel.

Vehicle angular velocities and accelerations Next, we show that the angular velocity can be represented as a function of the flat outputs.

We take the first derivative of Equation (3.52), in which we let $\boldsymbol{a} = \ddot{\boldsymbol{p}}$ denote the acceleration and $\dot{\boldsymbol{a}} = \dddot{\boldsymbol{p}}$ the jerk for the sake of readability,

$$\tilde{\boldsymbol{M}}\dot{\boldsymbol{a}} + \tilde{\boldsymbol{D}}\boldsymbol{a} = u_1 {}^{\mathcal{W}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}} \times {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}} + \dot{u}_1 {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}. \tag{3.58}$$

Its projection along the $\boldsymbol{x}_{\mathcal{B}}$-axis yields

$$^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}^{\top}\left(\tilde{\boldsymbol{M}}\dot{\boldsymbol{a}} + \tilde{\boldsymbol{D}}\boldsymbol{a}\right) = \dot{u}_1 {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}^{\top} {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}. \tag{3.59}$$

Note that $^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}$ has unit length by definition. We obtain

$$\dot{u}_1 = {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}^{\top}\left(\tilde{\boldsymbol{M}}\dot{\boldsymbol{a}} + \tilde{\boldsymbol{D}}\boldsymbol{a}\right). \tag{3.60}$$

We insert our expression of $\dot{u}_1$ into Equation (3.58) and define the vector $\boldsymbol{h}_{\omega}$

$$\begin{aligned} \boldsymbol{h}_{\omega} &= {}^{\mathcal{W}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}} \times {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}, && (3.61) \\ &= 1/u_1 \left(\tilde{\boldsymbol{M}}\dot{\boldsymbol{a}} + \tilde{\boldsymbol{D}}\boldsymbol{a} - \left({}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}^{\top}\left(\tilde{\boldsymbol{M}}\dot{\boldsymbol{a}} + \tilde{\boldsymbol{D}}\boldsymbol{a}\right)\right) {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}\right). && (3.62) \end{aligned}$$

We observe that $\boldsymbol{h}_{\omega}$ corresponds to the projection of $\tilde{\boldsymbol{M}}\dot{\boldsymbol{a}} + \tilde{\boldsymbol{D}}\boldsymbol{a}$ onto the $y_{\mathcal{B}}$-$z_{\mathcal{B}}$ plane. Note, the angular velocity (Equation (3.6)) can also be written in terms of the body frame components

$$^{\mathcal{W}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}} = p\,{}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}} + q\,{}^{\mathcal{W}}\boldsymbol{y}_{\mathcal{B}} + r\,{}^{\mathcal{W}}\boldsymbol{z}_{\mathcal{B}}. \tag{3.63}$$

Hence, we obtain the q and r components of the angular velocity ${}^{\mathcal{W}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}}$ via the projection along the $y_{\mathcal{B}}$ and $z_{\mathcal{B}}$ axis, reading

$$q = {}^{\mathcal{W}}\boldsymbol{y}_{\mathcal{B}}^{\top}\,\boldsymbol{h}_{\omega} \quad \text{and} \quad r = {}^{\mathcal{W}}\boldsymbol{z}_{\mathcal{B}}^{\top}\,\boldsymbol{h}_{\omega}, \tag{3.64}$$

respectively. Note that in $\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}} = \boldsymbol{\omega}_{\mathcal{B}/\mathcal{C}} + \boldsymbol{\omega}_{\mathcal{C}/\mathcal{W}}$ the latter term $\boldsymbol{\omega}_{\mathcal{C}/\mathcal{W}}$ has no $x_{\mathcal{B}}$-component. Thus,

$$p = {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}^{\top}\,{}^{\mathcal{W}}\boldsymbol{\omega}_{\mathcal{C}/\mathcal{W}} = {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}^{\top}\,\dot{\sigma}_4\,\boldsymbol{e}_1. \tag{3.65}$$

Reviewing Eqs. (3.64) and (3.65), we observe that the angular velocities can be expressed in terms of the flat outputs.

In order to express the angular accelerations $\boldsymbol{\alpha}_{B/W}$ in terms of the flat outputs we take the second time derivative of Equation (3.29). Next, we proceed analogously to the above to find expressions for the $y_{\mathcal{B}}$- and $z_{\mathcal{B}}$-components. We find the $x_{\mathcal{B}}$-component by examining

$${}^{\mathcal{W}}\boldsymbol{\alpha}_{\mathcal{B}/\mathcal{W}} = {}^{\mathcal{W}}\boldsymbol{\alpha}_{\mathcal{B}/\mathcal{C}} + \boldsymbol{\omega}_{\mathcal{C}/\mathcal{W}} \times {}^{\mathcal{W}}\boldsymbol{\alpha}_{\mathcal{B}/\mathcal{C}} + {}^{\mathcal{W}}\boldsymbol{\alpha}_{\mathcal{C}/\mathcal{W}} \tag{3.66}$$

and directly observe that ${}^{\mathcal{W}}\boldsymbol{\alpha}_{\mathcal{B}/\mathcal{C}}{}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}} = 0$ and ${}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{C}/\mathcal{W}} \times {}^{\mathcal{W}}\boldsymbol{\alpha}_{\mathcal{B}/\mathcal{C}} = 0$. Hence, for the angular accelerations $x_{\mathcal{B}}$-component, we obtain

$${}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}^{\top}{}^{\mathcal{W}}\boldsymbol{\alpha}_{\mathcal{B}/\mathcal{W}} = {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}^{\top}{}^{\mathcal{W}}\boldsymbol{\alpha}_{\mathcal{C}/\mathcal{W}} = {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}^{\top}\ddot{\sigma}_4\boldsymbol{e}_1, \tag{3.67}$$

which again is a function of the flat outputs and their derivatives.

Net thrust and moments Finally, we show that the system's control inputs can be expressed in term of the flat outputs Equation (3.50). By applying the Euclidean norm to Equation (3.29) the collective thrust u_1 reads

$$u_1 = \|\tilde{\boldsymbol{M}}\ddot{\boldsymbol{p}} + \tilde{\boldsymbol{D}}\dot{\boldsymbol{p}}\|. \tag{3.68}$$

Finally, we obtain the torque inputs u_2, u_3, and u_4 as functions of the flat outputs from Equation (3.30), since we can express both, angular velocity and acceleration, in terms of the flat outputs.

3.3.2 Geometric Tracking Control

Tracking of prescribed trajectories is a key capability of autonomous mobile robots. Research in recent years drastically pushed the agility of aerial quadrotor drones resulting in the need for control schemes which are able to handle very large attitude errors. As a consequence, geometric control schemes have gained wide attention in field of aerial drones [72, 73, 80]. Geometric schemes constitute an appealing class for this control problem as they aim to avoid singularities and ambiguities. The concept of geometric control addresses the control problem of dynamic systems which evolve on nonlinear manifolds and cannot be globally identified by Euclidean spaces. Therefore, the attitude control error is defined in the special orthogonal group $SO(3)$, also referred to as the space of rotations. It consists of the group of 3×3 orthogonal matrices $\boldsymbol{R}$ with determinant *one*, i. e. $SO(3) = \{\boldsymbol{R} \in \mathbb{R}^{3\times3} | \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}, \det \boldsymbol{R} = 1\}$.

In the following, we first present a control scheme for trajectory tracking building on the aerial drone scheme presented by Mellinger and Kumar [80]. At the end of this subsection, we point out simplifications which allow to use the presented control scheme for waypoint tracking and direct attitude control. Finally, we discuss considerations for practical implementations onboard μAUVs .

Consider the tracking control problem of following a prescribed trajectory $\boldsymbol{\sigma}_{\text{des}}(t)$ in space. We have seen in Subsection 3.3.1 that the property of differential flatness can be exploited to reduce the complexity of trajectory generation within the space of flat outputs, namely $\boldsymbol{\sigma}(t) = [\boldsymbol{p}(t)\, \phi(t)]^\top$. Hence, our aim is to design a controller which takes such a trajectory $\boldsymbol{\sigma}_{\text{des}}$ as an input and computes the required control inputs $u_{1:4}$. Therefore, we can derive the reference signals such as the desired vehicle attitude $\boldsymbol{R}_{\text{des}}$ as algebraic expressions directly from $\boldsymbol{\sigma}_{\text{des}}(t)$ analogue to Subsection 3.3.1.

The design of a suitable tracking error formulation is a critical step in the control design process. Given the desired trajectory $\boldsymbol{\sigma}_{\text{des}}(t)$ we define the control errors with regard to the vehicle position and velocity as

$$\boldsymbol{e}_p = \boldsymbol{p} - \boldsymbol{p}_{\text{des}}, \tag{3.69}$$

$$\boldsymbol{e}_v = \dot{\boldsymbol{p}} - \dot{\boldsymbol{p}}_{\text{des}}. \tag{3.70}$$

Based on these, we can compute the desired thrust force $\boldsymbol{f}_{\text{des}}$ which is required to drive the vehicle along the trajectory. The desired force reads

$$\boldsymbol{f}_{\text{des}} = -\boldsymbol{K}_p \boldsymbol{e}_p - \boldsymbol{K}_v \boldsymbol{e}_v + \tilde{\boldsymbol{M}} \ddot{\boldsymbol{p}}_{\text{des}} + \tilde{\boldsymbol{D}} \dot{\boldsymbol{p}}, \tag{3.71}$$

where $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$ are positive definite diagonal gain matrices and $\tilde{\boldsymbol{M}}\ddot{\boldsymbol{p}}_{\text{des}}$ and $\tilde{\boldsymbol{D}}\dot{\boldsymbol{p}}$ denote feed-forward terms. Given the actuator setup of the HippoCampus μAUV , thrust can only be applied along the vehicle $x_{\mathcal{B}}$-axis. Hence, we obtain the thrust control input u_1 by projecting Equation (3.71) along the $x_{\mathcal{B}}$-axis, reading

$$\begin{aligned} u_1 &= {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}^{\top}\boldsymbol{f}_{\text{des}}, \\ &= {}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}^{\top}\left(-\boldsymbol{K}_p\boldsymbol{e}_p - \boldsymbol{K}_v\boldsymbol{e}_v + \tilde{\boldsymbol{M}}\ddot{\boldsymbol{p}} + \tilde{\boldsymbol{D}}\dot{\boldsymbol{p}}\right). \end{aligned} \tag{3.72}$$

Additionally, we compute required torque inputs $u_{2:4}$ to precisely follow $\boldsymbol{\sigma}_{\text{des}}$. Therefore, we must determine the desired orientation ${}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B},\text{des}}$ which we write short as $\boldsymbol{R}_{\text{des}}$. We obtain the desired vehicle's ${}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}}$ in world coordinates obviously via

$${}^{\mathcal{W}}\boldsymbol{x}_{\mathcal{B}} = \frac{\boldsymbol{f}_{\text{des}}}{||\boldsymbol{f}_{\text{des}}||}, \tag{3.73}$$

assuming that $\boldsymbol{f}_{\text{des}} \neq 0$. Note that, we omit the superscript $\mathcal{W}$ and the subscript "des" for the sake of readability for the following derivation. We can now calculate the desired $y_{\mathcal{B}}$ and $z_{\mathcal{B}}$-axis following an analogue concept to our previous derivation in Subsection 3.3.1. We remember that $\boldsymbol{y}_{\mathcal{C}} = [0 \sin(\phi) \cos(\phi)]^{\top}$. It follows

$$\boldsymbol{z}_{\mathcal{B}} = \frac{\boldsymbol{x}_{\mathcal{B}} \times \boldsymbol{y}_{\mathcal{C}}}{||\boldsymbol{x}_{\mathcal{B}} \times \boldsymbol{y}_{\mathcal{C}}||} \quad \text{and} \tag{3.74}$$

$$\boldsymbol{y}_{\mathcal{B}} = \boldsymbol{z}_{\mathcal{B}} \times \boldsymbol{x}_{\mathcal{B}}, \tag{3.75}$$

which again yields, provided $\boldsymbol{x}_{\mathcal{B}} \times \boldsymbol{y}_{\mathcal{C}} \neq 0$. As a result, we define the desired vehicle orientation $\boldsymbol{R}_{\text{des}} = [\boldsymbol{x}_{\mathcal{B},\text{des}}\, \boldsymbol{y}_{\mathcal{B},\text{des}}\, \boldsymbol{z}_{\mathcal{B},\text{des}}]$. Note that from a theoretical point of view the singularity $\boldsymbol{x}_{\mathcal{B}} \times \boldsymbol{y}_{\mathcal{C}} = 0$ is only a single point in $SO(3)$. However, when examining Equation (3.74) it is obvious that we face undesirable large changes when computing ${}^{\mathcal{W}}\boldsymbol{z}_{\mathcal{B}}$ in the vicinity of the singularity. Mellinger and Kumar [80] suggest a convenient fix to this issue which we apply to our control problem: We observe that also the negative of the desired axes, namely $-\boldsymbol{y}_{\mathcal{B},\text{des}}$ and $-\boldsymbol{z}_{\mathcal{B},\text{des}}$ is consistent with the desired roll-angle and the body-frame's $z_{\mathcal{B}}$-axis. Thus, we compute both solutions and choose the one which is closer to the μAUV's current orientation to determine $\boldsymbol{R}_{\text{des}}$.

We define the vectorial orientation error $\boldsymbol{e}_R \in \mathbb{R}^3$ based on the current and desired vehicle orientation as

$$\boldsymbol{e}_R = \begin{bmatrix} e_{\text{Rx}} & e_{\text{Ry}} & e_{\text{Rz}} \end{bmatrix}^{\top} = \frac{1}{2}\left(\boldsymbol{R}_{\text{des}}^{\top}\,{}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}} - {}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}}^{\top}\boldsymbol{R}_{\text{des}}\right)^{\vee}, \tag{3.76}$$

where $^{\vee}$ denotes the *vee map* operator which is used to obtain the $\mathbb{R}^3$ representation of the orientation error $SO(3) \times SO(3) \to \mathbb{R}^3$.

Next, we define the error on the angular velocity in the vehicle's body-frame

$$\boldsymbol{e}_{\omega} = {}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W}} - {}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{W},\mathrm{des}}. \tag{3.77}$$

Finally, we define the control law for the torque control inputs $u_{2:4}$ based on the errors Eqs. (3.76) and (3.77), reading

$$\begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = -\boldsymbol{K}_R \boldsymbol{e}_R - \boldsymbol{K}_{\omega} \boldsymbol{e}_{\omega} + \boldsymbol{\omega} \times \hat{\boldsymbol{J}}\boldsymbol{\omega} - \boldsymbol{\omega} \times \hat{\boldsymbol{M}}\boldsymbol{v} - \hat{\boldsymbol{D}}\boldsymbol{\omega}, \tag{3.78}$$

where $\boldsymbol{K}_R$ and $\boldsymbol{K}_{\omega}$ denote gain matrices in diagonal form. Moreover, we include $\omega \times \tilde{J}\omega$ as a feedback linearization term in order to cancel the corresponding term in Equation (3.30).

In this sense, the control moment in Equation (3.78) constitutes an attitude tracking controller, while the control thrust Equation (3.72) corresponds to a tracking controller on the translational dynamics. For instance, if the force $\boldsymbol{f}_{\mathrm{des}}$ and the vehicle's $x_{\mathcal{B}}$-axis align, i. e. the attitude tracking error in Equation (3.76) is zero, Equation (3.72) denotes the translational tracking controller. Obviously, this is not necessarily the case. Thus, for the sake of stability we prioritize the minimization of the attitude tracking error by temporary scaling down the thrust input Equation (3.71) as we project $\boldsymbol{f}_{\mathrm{des}}$ along the $x_{\mathcal{B}}$-axis Equation (3.72).

Lee et al. [73] provide a stability proof for the above control scheme considering the application example of aerial quadrotors. Our system and the controller hold most of their assumptions such as the consideration of feedforward and feedback terms to cancel out e. g. $\omega \times J\omega$ as well as negligible motor dynamics. However, from a practical point of view, the authors' assumption of perfect knowledge on the vehicle's mass and rotational inertia matrices hardly holds for a physical platform such as the HippoCampus μAUV. The same applies for the requirement of the gain matrices being diagonal with identical elements. Still, the control scheme shows high accuracy and strong robustness in real-world deployment for which we refer to two experimental case studies in Section 3.4.

Remark – Adaptation to Path- and Way-Point-Tracking: We derived the control scheme above to address the goal of tracking a prescribed trajectory $\boldsymbol{\sigma}(t)$. However, the scheme can also be used for less complex tasks which are common in mobile robotics, for instance path or way-point tracking. In this case, we provide a path or a sequence of way-points which are tracked subsequently by the μAUV. Various methods exist to compute the desired orientation while tracking, examples include among others the line-of-sight

and pure-pursuit approaches [37] as well as the computation of splines. In these cases it is often convenient to set the desired angular velocity $\boldsymbol{\omega}_{\text{des}}$ to zero, cf. Equation (3.70). However, while tracking a path or sequence of way-points is in general straight-forward, stabilizing at a single point in space is challenging given the underactuated configuration of the HippoCampus μAUV. Thus, a common approach is to tightly circle the path's final destination for loitering similar to fixed-wing aircraft.

Remark – Actuator Saturation: In practice, the vehicle actuator system is subject to various physical constraints. Note, while from a control perspective it is convenient to define the control inputs in terms of linear surge thrust and torques physical constraints are imposed by the individual thrusters. As the vehicle's thrusters cannot produce infinite high forces and torques this can result in saturation effects, see Figure 3.8. The thruster constraints are mostly of concern for the linear translational thrust and rotations in pitch and yaw. These constraints arise from the coupling as the rotational torques induced by force pairs, cf. Equation (3.49). An obvious but conservative approach is to limit the desired surge thrust and the pitch/yaw moments such that saturation is avoided in any case. However, this considerably reduces the vehicle's agile maneuvering capabilities. Thus, we prefer an adaptive approach which temporary prioritizes single control inputs over the others. Roughly speaking, we rank the control of pitch and yaw motions over linear thrust and roll. A practical implementation of this approach is shown in Equation (3.72). The projection of the desired force $\boldsymbol{f}_{\text{des}}$ along the vehicle's $x_{\mathcal{B}}$-axis dynamically reduces the forwards thrust for large deviations between the desired and the current vehicle orientation. As a consequence, more actuator bandwidth is temporary made available for attitude control commands which supports the agile maneuvering of the vehicle.

Remark – Deadzone Effects: The propulsion system of the HippoCampus μAUV is driven by brushless DC-motors, as described in Section 3.1.1. These motors come with the advantage of a high power-to-mass ratio resulting in attractive dynamics. Among others, this is one of the main reasons for the wide spread application of this motor type in aerials drones. However, these motors are naturally subject to deadzone effects which means they cannot spin at arbitrary low velocities. This effect is sketched in Figure 3.8. While this may be appealing to avoid wear and tear due to noise on the control signal it limits the control systems ability to apply small forces and torques. This aspect constitutes an interesting difference when comparing marine robots to aerial drones. Aerial drone motors usually only produce *positive* thrust and do not change their spinning direction during mission as a minimum thrust is needed at all times to counter the gravitational force. This is not the case in the marine domain where vehicle designs aim for neutral buoyancy

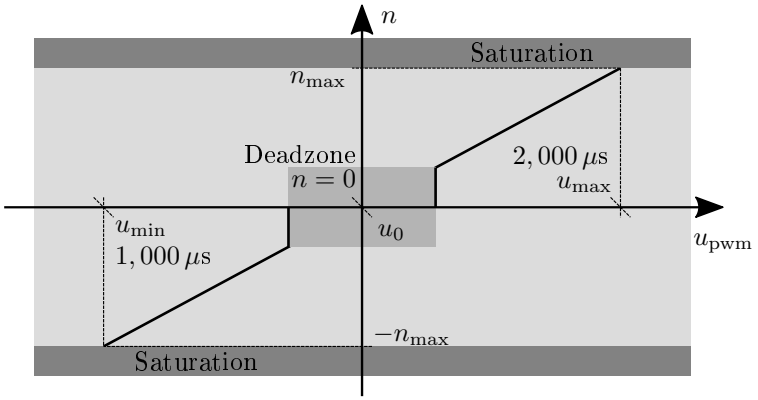


Figure 3.8: Deadzone and saturation effect in brushless DC motors: a minimum control signal is required for the motor to start spinning. For control signals above u_{max} and below u_{min} the motors enter the saturation area with its maximum speed $\|n_{max}\|$.

and thus do not need to counter gravitational forces. However, in practice this negative impact from deadzone effects is mostly limited to slow motions. Given a sufficiently large forward thrust force where all motors are spinning well above the deadzone, pitch and yaw moments can be realized via differences in positive thrust. This yields particularly for vehicle designed exploratory application scenarios whose motion spectrum is dominated by surge motions. However, stepper-motors – despite their higher complexity and cost – can be an alternative and have been deployed for instance on the μAUV platform in [47].

3.4 Case Study: Hydrobatic Motion Control

An experimental performance study is conducted in order to show that the developed HippoCampus μAUV platform meets the maneuvering requirements for operations in complex confined environments which we established in Subsection 2.3.1. Therefore, we design a case study which targets the dynamic and control capabilities of the platform with regard to robustness and accuracy, illustrated in Figure 3.9.

In the following study, we target the vehicle attitude control performance, i. e. independent from surge motion. Our focus lies on the control system's ability to stabilize the vehicle at an desired pose while it is subject to external disturbances. In order to increase the difficulty of this task we extend the regular HippoCampus vehicle setup and use the μAUV as a platform to swing-up and stabilize a Furuta pendulum underwater. This case study has been recently published in [27]. Thus, we present a summary of the concept and the findings in the following. For further details, an extensive discussion, and an accompanying video of the experiments, we refer the reader to the original work:

- D.-A. Duecker, A. Hackbarth, T. Johannink, E. Kreuzer, and E. Solowjow. Micro Underwater Vehicle Hydrobatics: A Submerged Furuta Pendulum. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7498-7503, Brisbane, Australia, 2018.

Additionally, we conduct a second case study on hydrobatic motion control which examines the agile maneuvering capabilities of the HippoCampus μAUV. Therefore, we consider the problem of accurately tracking a prescribed path in a confined environment at high speeds. This allows to study the previously analyzed attitude control scheme in combination with controlled surge motion in a realistic deployment scenario. In order to stress the control scheme capabilities we extend the classical path tracking problem with complex hydrobatic elements such as a barrel roll. Although hydrobatic maneuvers are not standard elements during everyday missions, they are very helpful to challenge the control scheme and provide insights on the maneuvering capabilities of the μAUV. Besides the aspects on hydrobatic maneuvering and path-tracking this study provides important insights on the robustness of visual underwater self-localization. The related concepts are discussed in detail in Section 4.3. For this reason and to avoid repeating the discussion and findings, we refer the reader to Subsection 4.3.3 for the maneuvering study and keep our focus in this section to the problem of Furuta pendulum control.

3.4.1 Problem Statement

The problem of pendulum swing-up and stabilization is widely used within the control community as a benchmark setup to test and compare newly developed control algorithms. While the original research interest focused on the pendulum dynamics itself, recent works go beyond this original setup. For instance, Hehn et al. [50] study the combined problem of an inverted pendulum which is balanced by an aerial drone to demonstrate the control performance of the robot. In this case study, we consider the problem of an inverted rotary

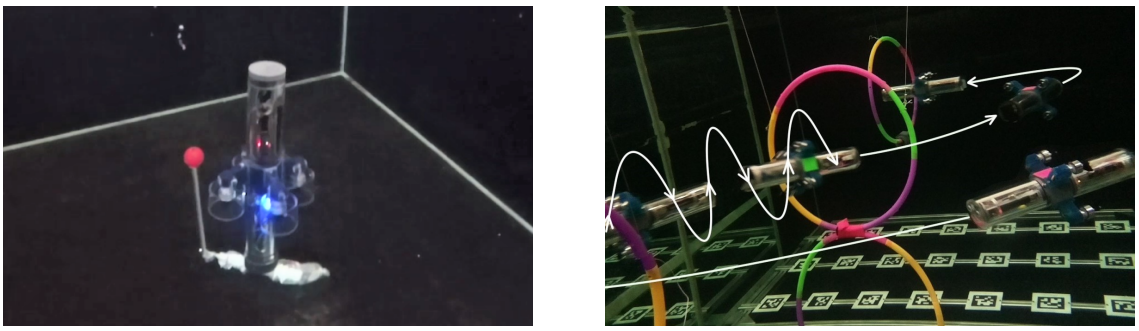


Figure 3.9: Hydrobatic Case Studies: Study I, Furuta Pendulum control (*left*) and Study II, Hydrobatic path tracking (*right*).

pendulum which has been introduced in the early 1990s by Furuta et al. [40, 41]. Thus, it is often referred to as the *Furuta pendulum*. It has become a popular example of a nonlinear oscillator which has been widely studied in the literature [106, 107]. In the following, we use the Furuta pendulum as a hydrobatic control problem for the HippoCampus μAUV to explore the capabilities of the vehicle maneuvering and control system. Our goal is to swing-up and stabilize the Furuta pendulum using the μAUV-platform to examine its control performance. For this purpose, we firstly extend the well-known Furuta pendulum dynamics to the submerged case. Secondly, we extend our control scheme from Subsection 3.3 by a pendulum controller.

3.4.2 Furuta Pendulum Dynamics

The Furuta pendulum is mounted to the HippoCampus and consists of two slender poles, see Figure 3.10. The base arm is fixed on the μAUV's aft while the pendulum pole can rotate around the arm's axis. The pendulum pole is supported on the arm by two ball bearings. The arm's orientation and angular velocity are described by the angle α_a and its velocity $\dot{\alpha}_\mathrm{a}$, respectively. Note that in our setup these correspond to the μAUV's roll-axis. The relative angle α_p between the pendulum pole and the vehicle can be measured by an angular potentiometer. The torque applied to the pendulum arm is denoted by m_a.

Pendulum Dynamics with Hydrodynamic Effects

In the following, we extend the Furuta pendulum's equation of motion to the underwater domain and hydrodynamic effects. For this purpose, we assume the pendulum to be hydrodynamically transparent. This is a valid assumption for slender objects as they have

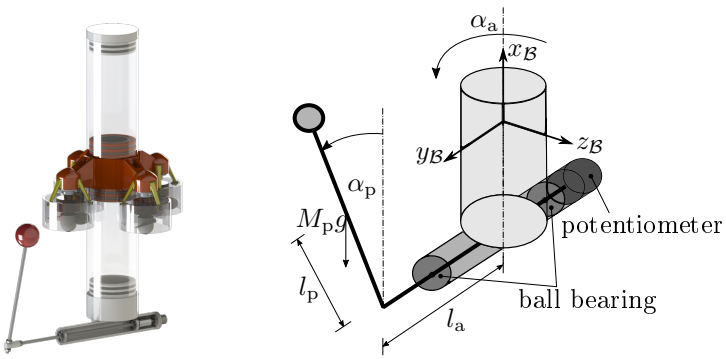


Figure 3.10: CAD-Drawing (*left*) and simplified free-body diagram (*right*) of HippoCampus μAUV with Furuta pendulum mounted on its aft. The angle α_p is measured by a potentiometer. The pendulum pole is supported by two ball bearings within the pendulum arm.

negligible radiation and diffraction effects. Based on this, we use the Morison equation [60] to determine the hydrodynamic loads on the pendulum due to its movement through water. The Morison equation describes the force load $\mathrm{d}F_\mathrm{x}$ on a pole segment $\mathrm{d}z_\mathrm{p}$ caused by a water flow around the pole. Thereby, we follow the common assumption that only the water flow perpendicular to the pole u_flow results in a force load, reading

$$\begin{aligned}\mathrm{d}F_\mathrm{x} &= \left(f_\mathrm{m}\left(u_\mathrm{flow}\right) + f_\mathrm{d}\left(u_\mathrm{flow}\right)\right)\mathrm{d}z \\ &= \left(C_\mathrm{m}\rho\frac{\pi d_\mathrm{p}^2}{4}\frac{\partial u_\mathrm{flow}}{\partial t} + C_\mathrm{d}\frac{\rho}{2_\mathrm{p}}\left\|u_\mathrm{flow}\right\|u_\mathrm{flow}\right)\mathrm{d}z.\end{aligned} \tag{3.79}$$

Hereby, d_p denotes the diameter of the pole and ρ is the water density. The inertia and drag coefficients are denoted by C_m and C_d respectively and are determined experimentally. In order to improve readability we use the short notation

$$\begin{aligned}u_\mathrm{flow,p} &= u_\mathrm{flow,p}\left(\alpha_\mathrm{p}, \dot{\alpha}_\mathrm{p}, \dot{\alpha}_\mathrm{a}, z_\mathrm{p}\right) \\ u_\mathrm{flow,a} &= u_\mathrm{flow,a}\left(\dot{\alpha}_\mathrm{a}, z_\mathrm{a}\right),\end{aligned}$$

where expressions with subscript p belong to the pendulum and the subscript a denotes the pendulum arm, respectively. The resulting moment m_morison on the pendulum and its arm can be determined by solving the corresponding integrals

$$\begin{aligned}m_\mathrm{morison,p} &= \int^{l_\mathrm{p}^*} z_\mathrm{p}\left(f_\mathrm{m}\left(u_\mathrm{flow,p}\right) + f_\mathrm{d}\left(u_\mathrm{flow,p}\right)\right)\mathrm{d}z_\mathrm{p}, \\ m_\mathrm{morison,a} &= \int^{l_\mathrm{a}^*} z_\mathrm{a}\left(f_\mathrm{m}\left(u_\mathrm{flow,a}\right) + f_\mathrm{d}\left(u_\mathrm{flow,a}\right)\right)\mathrm{d}z_\mathrm{a},\end{aligned} \tag{3.80}$$

where l_p^* and l_a^* denote the total length of the pendulum pole and the pendulum arm. Note, the resulting moments still depend on the corresponding water flow. The hydrodynamic effects on the pendulum are then included into the original Furuta pendulum equation of motion [40]. Their conjunction reads

$$\begin{aligned}&\begin{bmatrix} I_\mathrm{a} + M_\mathrm{p}\left(l_\mathrm{a}^2 + l_\mathrm{p}\sin^2\left(\alpha_\mathrm{p}\right)\right) & M_\mathrm{p} l_\mathrm{p} l_\mathrm{a}\cos\left(\alpha_\mathrm{p}\right) \\ M_\mathrm{p} l_\mathrm{p} l_\mathrm{a}\cos\left(\alpha_\mathrm{p}\right) & I_\mathrm{p} + M_\mathrm{p} l_\mathrm{p}^2 \end{bmatrix}\begin{bmatrix}\ddot{\alpha}_\mathrm{a} \\ \ddot{\alpha}_\mathrm{p}\end{bmatrix} \\ &\quad + \begin{bmatrix} 2M_\mathrm{p} l_\mathrm{p}^2\sin\left(2\alpha_\mathrm{p}\right)\dot{\alpha}_\mathrm{p} & -M_\mathrm{p} l_\mathrm{p} l_\mathrm{a}\cos\left(\alpha_\mathrm{p}\right) \\ -M_\mathrm{p} l_\mathrm{p}^2\sin\left(\alpha_\mathrm{p}\right)\dot{\alpha}_\mathrm{a} & I_\mathrm{a} + M_\mathrm{p} l_\mathrm{p}^2 \end{bmatrix}\begin{bmatrix}\dot{\alpha}_\mathrm{a} \\ \dot{\alpha}_\mathrm{p}\end{bmatrix} \\ &\quad + \begin{bmatrix} m_\mathrm{morison,a} \\ -M_\mathrm{p} l_\mathrm{p} g\sin\left(\alpha_\mathrm{p}\right) + m_\mathrm{morison,p}\end{bmatrix} = \begin{bmatrix} m_\mathrm{a} \\ 0\end{bmatrix},\end{aligned} \tag{3.81}$$

where M_p and M_a are the masses of the pendulum pole and the pendulum arm and I_p and I_a are their corresponding mass moments of inertia. Moreover, l_p is the distance between

the rotational axis and the pendulum pole's gravity center. Furthermore, g denotes the gravitational vector. Note, we assume that the pendulum arm is rotating within the horizontal plane which is roughly the case during our pendulum experiments.

Combined Dynamics

The Equations (3.27) and (3.81) describe the full dynamics of the system. Both are coupled, as the pendulum structure is mounted on the vehicle's aft. Thus, the angular velocity of the pendulum arm $\dot{\alpha}_\mathrm{a}$ and its time derivative $\ddot{\alpha}_\mathrm{a}$ correspond to the vehicle roll rate p and its derivative $\dot{p}$. We further assume that the vehicle is capable to keep its $x_\mathcal{B}$-axis upright while carrying the pendulum structure. Due to the mechanical coupling the vehicle controller can apply a moment m_a to the pendulum via the roll control input u_2.

3.4.3 Furuta Pendulum Control

In order to enable the HippoCampus μAUV-platform to swing-up and stabilize an attached Furuta pendulum we extend the original control scheme from Section 3.3 with a control module dedicated to pendulum control, see Figure 3.11. Therefore, we add a linear feedback controller to stabilize the Furuta pendulum. Based on the measured angle of the pendulum arm the desired moment around the $x_\mathcal{B}$-axis is computed. The control signal is fed into the attitude control structure, as depicted in Figure 3.11.

Note that in order to use the HippoCampus μAUV as an actuator base for the Furuta pendulum the geometric controller is used to hold the vehicle in an upright pose at a desired depth. This allows the control output u_2 to be exclusively used for the stabilization of the pendulum. However, as previously discussed, each thruster's force and torque contribute to the total surge thrust and the torques for yaw, pitch, and roll rotation. Thus, saturation effects due to superposition of all control signals may deteriorate performance. The pendulum swing-up phase is realized by a combination of the attitude and pendulum controllers. While the swing-up is typically realized via a rotation of the pendulum arm we extend this control sequence by a downward motion to support the swing-up motion of the pendulum.

3.4.4 Experiments

We conduct two experiments: First, we study the capabilities of solely the geometric attitude controller proposed in Subsection 3.3.2. Second, we examine the combined setup of the attitude controller in conjunction with the pendulum controller. The vehicle swings-

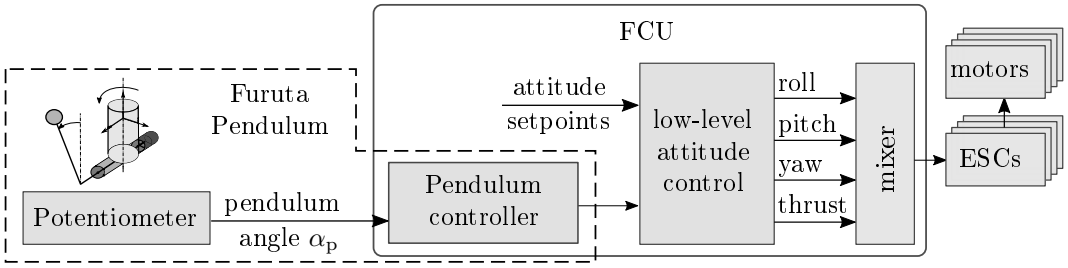


Figure 3.11: Furuta pendulum control architecture of the HippoCampus μAUV. The pendulum controller (dashed box) is only used for the pendulum experiment and has a direct feed-through to the mixer.

up and stabilizes a Furuta pendulum. In both experiments, we apply external disturbances to the system in order to demonstrate the robustness and fast recovery times of the control scheme. Moreover, we refer to the accompanying video in [27], where both experiments are presented.

Experimental Setup

In the physical setup, the Furuta pendulum consists of two aluminum poles, the pendulum pole (length $l_\mathrm{p}^* = 2l_\mathrm{p} = 220\,\mathrm{mm}$, diameter $d_\mathrm{p} = 10\,\mathrm{mm}$) and the hollow pendulum arm (length $l_\mathrm{a}^* = 220\,\mathrm{mm}$, outer diameter $d_\mathrm{a} = 22\,\mathrm{mm}$). Note that due to the design, the effective length of the pendulum arm is $l_\mathrm{a} = l_\mathrm{a}^*/2 = 110\,\mathrm{mm}$. In order to improve the visibility of the pendulum tip a water flooded sphere with a diameter of 35 mm is mounted to the pendulum tip. In order to reduce friction and enable free rotation of the pendulum pole, it is supported by two ball bearings inside the pendulum arm, as sketched in Figure 3.10. The relative orientation α_p between pendulum pole and the μAUV is measured by an analog angular potentiometer mounted on the pendulum arm. The potentiometer and the ball bearings are sealed with off-the-shelf latex membranes.

Experiment I: Geometric Attitude and Depth Control

In this first experiment, the μAUV initially moves to its target depth while holding its upright pose. In order to drive the vehicle upright we set the desired pitch angle to $\pi/2$, which brings the vehicle attitude close to a singularity. We use the y- and z-components of the orientation error $\boldsymbol{e}_\mathrm{R}$, namely e_Ry and e_Rz, to evaluate the control performance, see Figure 3.12. These constitute a convenient error measure, since they fully represent the orientation relative to the desired upright pose, see Equation (3.76) for definition. The experiment is initialized with the HippoCampus μAUV floating on the water surface with its controllers being switched off. When the controllers are started at $t = 10\,\mathrm{s}$ the actuators

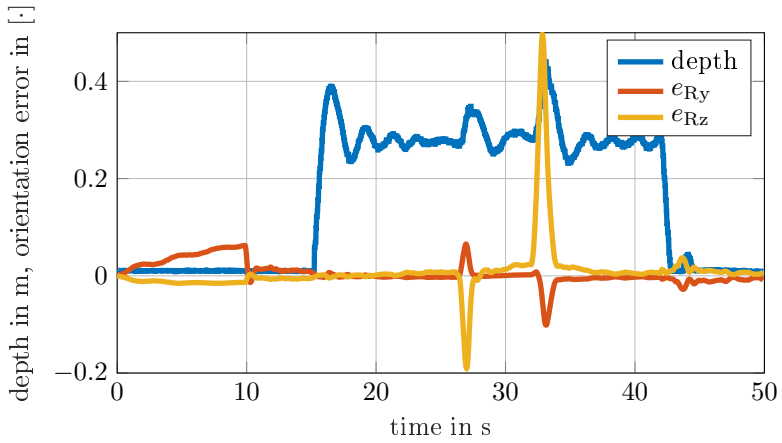


Figure 3.12: Experimental results: The orientation errors e_{Ry} and e_{Rz} describe the orientation mismatch relative to the upright pose. At $t = 15\,s$ the target depth is set to 30 cm. At $t = 26\,s$ and $t = 32\,s$ physical disturbances are applied by poking the μAUV with a pole.

drive the vehicle to its upright target pose. At $t = 15\,s$ the target depth is set to 30 cm and the vehicle moves downwards. However, Figure 3.12 shows that the vehicle first overshoots the target depth and then slightly oscillates around it. This is mainly due to the previously discussed deadzone effect of the propulsion system, see also Figure 3.8. At approximately $t = 26\,s$ and $t = 32\,s$ the vehicle is subject to severe physical disturbances, which are applied by strongly poking the vehicle with a pole. They result in large changes in the orientation and vehicle depth. However, we observe that the μAUV can quickly counter even from large orientation errors. This can be implied from the observation that the errors e_{Ry} and e_{Rz} decrease quickly which demonstrates the robustness of the developed control system.

Experiment II: Swing-Up and Stabilization of the Furuta Pendulum

The attitude hold task of the previous experiment is now extended by the swing-up and stabilization of the Furuta pendulum which is mounted to the HippoCampus μAUV. Starting from an initial depth the μAUV rotates and temporarily moves downwards to swing-up the pendulum, as depicted in the photo series in Figure 3.13. Afterwards, the μAUV stabilizes the inverted pendulum and itself at the desired target depth. The angle of the pendulum α_p is plotted in Figure 3.14. A zero pendulum angle represents the pendulum pole aligned with the vehicle's x_B-axis. Although the angle signal is subject to high noise the controller is able to balance and stabilize the Furuta pendulum. Note, we use an analog angular potentiometer, which is not perfectly waterproof. Thus, this strong noise on the angle signal is likely the result of wear and tear of the potentiometer and

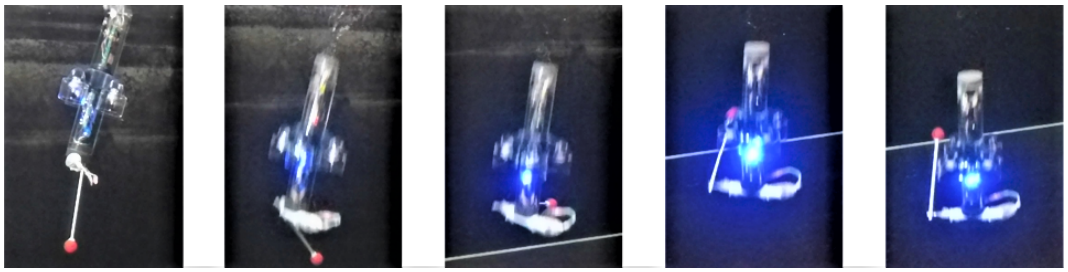


Figure 3.13: Pendulum swing-up by a combination of rotation and downward motion.

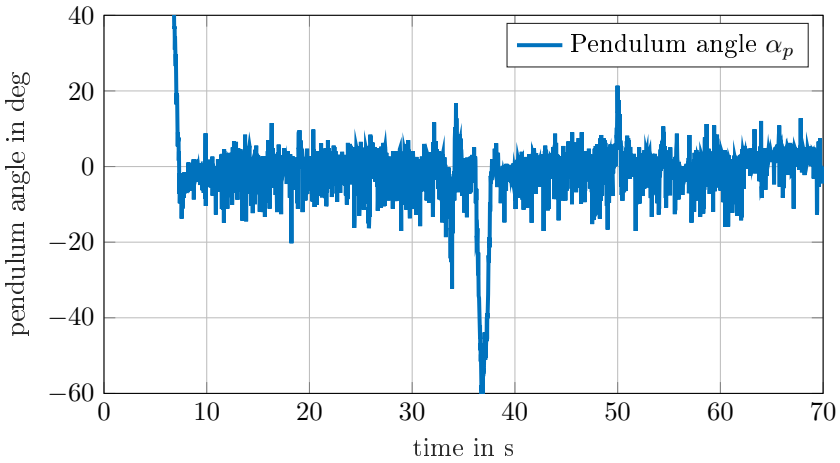


Figure 3.14: Swing-up and stabilization of the Furuta pendulum. At $t = 32\,\mathrm{s}$, $t = 37\,\mathrm{s}$, and $t = 50\,\mathrm{s}$ the pendulum tip is poked with a pole. The system quickly recovers from these disturbances.

is further intensified by the sensor's humid surrounding. The remaining control error is caused by a slight mounting mismatch between the upright position of the potentiometer and the $x_{\mathcal{B}}$-axis. As a consequence, the remaining offset on the pendulum angle α_p results in continuous rotation of the μAUV although the controller stabilizes the pendulum in this unstable up-right position.

In order to demonstrate the robustness of the combined system the pendulum tip is being poked with a pole at $t = 32\,\mathrm{s}$, $t = 37\,\mathrm{s}$, and $t = 50\,\mathrm{s}$. The vehicle controller can recover the pendulum even from deflections of 60 degrees, which demonstrates the agile dynamics of the vehicle. In fact, it is challenging to apply large deviation to the pendulum with the pole, since the vehicle controller counteracts these disturbances instantaneously. Finally, it is worth mentioning that the pendulum can even be stabilized while the μAUV moves upwards. This represents an especially challenging task as the pendulum is subject to the resulting downward water flow which increases the instability of the pendulum system.

3.5 Summary and Discussion

This chapter addressed the identified capability gap of today's underwater robots being able to agile maneuver in a confined fluid volume as it is required e. g. in exploratory missions. In order to address this gap, a modular software architecture in combination with a mechatronic robot design was developed. Hardware-wise this development resulted in a fully revised version of HippoCampus μAUV which founded a new class of hydrobatic underwater vessels. The platform is low-cost, easy to maintain, and focuses on a high adaptability to wide range of missions. An detailed study of the HippoCampus vehicle dynamics revealed a link between agile aerial quadrotor drone drones and the HippoCampus design. It was shown that for certain assumptions the design and propulsion concept yields the property of being differentially flat. Furthermore, a nonlinear geometric low-level control scheme was developed. A case study including a real-world experiments was conducted to demonstrate the agile control performance of the vehicle by swing-up and and balance an inverted Furuta-pendulum. To the best of the author's knowledge, the combination of low-cost, high-agility, and vehicle size of the HippoCampus platform is currently unique within the family of small-scale underwater robots. The findings of this chapter have been published in a series of papers [22, 27, 28, 29, 31].

Chapter 4

Embedded Self-Localization for μAUVs

Localization constitutes a critical component for any mobile robotic system. In fact, it is a fundamental prerequisite for almost any kind of autonomous behavior. At the very simplest level, a human operator perceives the robot's environment, e. g. via a camera. For this task, salient features are usually good reference points to determine the robot's current position. The operator can use the visual information directly (i. e. without a map) to navigate the robot to the desired target. However, most localization approaches, especially those targeting high levels of autonomy, aim to determine the robot's position within a map linked to a world-fixed reference frame.

The requirements on localization systems are continuously rising with the increasing autonomy of robotic systems. However, localization in underwater domains is particularly challenging. The reasons lie, among others, in the absence of common terrestrial localization techniques. For instance, the electromagnetic signals of GNSS attenuate in water just after centimeters. Moreover, underwater domains – in contrast to most terrestrial domains – possess a very limited availability of salient features. Furthermore, external systems such as camera tracking systems are unsuitable as well since they rely on robust low-latency communication links to transmit the position data to the robot. Such a link can usually not be assumed in submerged settings. As a consequence, the demand for on-board self-localization systems arises.

Despite the recent progress on localization techniques for full-scale ocean-going underwater robots, accurate and robust underwater localization remains a severe bottleneck in the development towards higher autonomy levels [116]. Furthermore, localization concepts for full-scale systems usually cannot be transferred to the domain of small-scale robots and applications in confined environments since the localization systems are too large and costly [133]. Moreover, the localization approach of a full-scale system may simply not work in restricted spaces.

As a result from the earlier discussion in Subsection 2.3.2, we can summarize the requirements for μAUV localization systems in confined spaces as follows:

1. Position information must be obtained with respect to an absolute reference frame.
2. The system should allow for onboard self-localization of the robot.
3. The localization system is ideally scalable to the deployment of multiple vehicles.

While the first point results from the desired task of field exploration and monitoring, the second requirement is caused by the absence of reliable and high bandwidth underwater communications. This generally rules out external approaches and favors fully embedded designs. Finally, third, passive approaches are favorable.

Given these requirements, this chapter develops an appropriate localization framework for μAUVs. Thereby our goal is to develop a scheme that flexibly addresses the posed requirements in various deployment scenarios. Therefore, we exemplary target and examine the localization problem posed in the three application scenarios defined in Section 2.1.3. The reference scenarios are briefly restated for the reader's convenience:

Scenario I - Hydrobatic Control in Confined Water Basins:

- highly agile motions with strong requirements on accuracy and robustness
- good visibility conditions

Scenario II - Navigation in Strictly Confined Volumes:

- regular operation with μAUVs in strictly confined volumes
- poor visibility conditions

Scenario III - Navigation in Marinas and Offshore Dumping Sites:

- regular operation with μAUVs in medium to large confined volumes
- poor visibility conditions

The remainder of this chapter is structured as follows: First, we provide a brief review of available localization technologies for underwater vehicles. In the following, we discuss the suitability of these technologies in the light of the requirements imposed by agile μAUV platforms. Thereby, we consider the complex properties of the reference Scenarios I-III

defined in Section 2.1.3. Second, in Section 4.2, we present a modular localization architecture that allows to address the various identified requirements. The third part of this chapter proposes three localization concepts that target the individual scenario requirements. These concepts are based on three physical principles, namely, vision (Section 4.3), electromagnetics (Section 4.4), and acoustics (Section 4.5). We provide an experimental performance study for each localization concept and critically discuss its suitability for the targeted scenario. The chapter concludes with a brief discussion in Section 4.6.

The content of this chapter has been partly published in a series of journal articles [25, 133] and conference proceedings [10, 23, 24, 28, 30, 33].

4.1 Review on Localization Technologies

Underwater localization methods are based on at least one of four fundamental physical principles: inertia, vision, acoustics, and electromagnetics. The following subsections provide a review of fundamental localization technologies, discuss their approaches, and point out the technologies' individual strengths and flaws.

4.1.1 Inertial Navigation

Inertial navigation describes the process of determining the robot's position and velocity based on first-principles of mechanics and the time integration of sensor data. For instance, velocity changes can be obtained via time integration from the measured accelerations. Similarly, the orientation is obtained by integrating the measured angular velocities. Following this, position changes can be computed via a second time integration. Given the initial vehicle pose and velocity, tracking of changes in position, orientation, and their respective velocities over time allows to compute the vehicle's current pose with respect to its start. This process is referred to as *dead reckoning*.

On the input side, inertial navigation systems process measurements of linear acceleration and angular velocity. These data streams are obtained from an onboard inertial measurement unit (IMU). Standard configurations of IMUs house usually three linear accelerometers as well as three gyroscopes. Within these sensor-triplets the sensors are mounted orthogonally to one another in order to provide measurements of motions in arbitrary directions. This setup is commonly referred to as 6-DOF IMU. However, more complex configurations including the combination of multiple 3-axes accelerometers and gyroscopes are available. This allows for higher robustness and wider measurement ranges.

The main advantage of inertial navigation is its independence from external factors and environmental conditions. In contrast, visual approaches require some kind of ambient light and acoustic systems are subject to reflections. Moreover, no additional infrastructure is required. A further advantage constitutes the high update rates of inertial navigation systems which usually lie in the range of at least 100 Hz and often remarkably above. This renders inertial concepts appealing for control tasks of highly dynamic motions.

On the downside, inertial navigation suffers strongly from accumulating errors which are exponentially growing with increasing application time. Thereby, inaccuracies in the orientation estimate constitute a dominant source of errors. Orientation errors are the direct result of uncompensated biases and drifts in the gyroscope signals. This hinders the correct removal of gravity acceleration on the linear accelerometers. This is of special importance in applications where linear accelerations are of lower or same order as the acceleration due to gravity. Both yield for most applications within the marine domain. The resulting acceleration including its additive error is integrated twice to obtain the desired position change. The consequence is an exponential drift when propagating the position over time. Additionally, errors in the orientation estimate lead to a wrong direction in which the linear acceleration and, thus, the position changes are propagated.

In order to reduce the effect of drifting gyroscope signals, some inertial navigation systems feature magnetometers. These sensors provide an absolute measurement of the surrounding magnetic field and, thus, an estimate of the vehicle's yaw orientation. However, magnetometers are heavily affected by various external disturbances, for example, induced by ferromagnetic materials in the vehicle's surroundings. These materials are often present in confined environments due to infrastructure or obstacles. Additionally, disturbances on magnetometers may be induced by electric components, e. g. motor thruster, in the sensor's vicinity. As a consequence, the gain from magnetometers onboard small-scale vehicles is often limited. The main performance metric for inertial navigation systems is their accumulating drift over time. This metric is mainly driven by the quality of the implemented IMU sensors and the capabilities of the software algorithm to identify and compensate sensor biases and drifts.

Nowadays, sophisticated IMU systems are available which can be categorized into unaided *tactical-grade* IMUs which accumulate position errors in the range of 100 km per hour and *navigation-grade* IMUs which accumulate position errors at 1 km per hour. IMUs are available at all price ranges depending on their underlying technology. Core technologies include ring-laser and micro-electro-mechanical systems (MEMS). MEMS-based IMUs faced a rapid development progress in recent years which was greatly boosted by technological advances in sensory equipment for smartphones and similar electronic products. MEMS-IMUs are generally well suited for small mobile robots due to the sensor's small

size and cost. However, commercial MEMS-based IMUs including 6-DOF sensing which match tactical grade requirements may achieve a gyroscope drift as small as 0.8 degrees per hour and lie in the price range of USD 3k. In the context of navigation grade accuracy, most IMU systems are based on fiber optical and ring-laser sensing. These systems lie one magnitude higher with respect to accuracy, size, and cost. A commercially available example is the KVH 1750 achieving gyroscope drift, as low as 0.05 degrees per hour with a cost of approximately USD 15k as of 2021. These price ranges must be considered in the context of the cost range of μAUV-platforms being USD 500-3.000. Hence, the concept of inertial navigation systems which solely rely on dead reckoning is of limited suitability for localization during autonomous missions. However, the high update rates of inertial approaches render their combination with lower-frequency technologies appealing. In such a setting, inertial navigation concepts can be used to leverage low-rate methods towards high update rates as they are required for dynamic motion control.

4.1.2 Visual Localization

Vision-based robot localization constitutes one of the most widely used robot localization technologies among all robot deployment domains. For vision systems, single or multi-camera approaches have become the standard on the sensing side. In order to obtain information from the environment the camera's light-sensitive chip-array measures the intensity of incoming light rays. The challenge is now how to efficiently extract relevant information from the camera image stream, for instance, the contours of perceived objects.

An important advantage of camera-based localization systems is their attractive ratio of high information richness to the comparatively low sensor price. Moreover, vision-based localization for mobile robots greatly benefits from the recent progress in the miniaturization of electronic components used, for instance, in commercial smartphones. Thus, powerful miniaturized cameras have been developed and are available at all price ranges where even low-cost cameras often provide sufficient sensing quality. This makes their integration into low-cost embedded platforms appealing. Additionally, camera-sensing systems are naturally passive. As a result, vision-based systems are not subject to multi-path effects or reflection which constitutes an advantage over acoustic or electromagnetic technologies.

However, vision-based systems require a free line of sight between the observed reference object and the sensing camera. Moreover, their range and accuracy strongly depend on the visibility conditions in the surrounding medium. For instance, light conditions, light absorption, and turbidity constitute important environmental parameters that strongly affect the sensing quality and, thus, may deteriorate the localization performance. This results in considerably reduced detection ranges when comparing submerged scenarios

against aerial and terrestrial settings. Additionally, underwater environments do not possess the feature richness of terrestrial scenarios. Moreover, the reduced texture quality underwater poses challenges for the visual detection of natural contours and structures. Furthermore, light spectrum absorption has to be considered in underwater domains. For instance, longer wavelengths (red light) are subject to stronger absorption than light in the blue spectrum.

Vision-based localization approaches can be distinguished between absolute and relative localization. Absolute localization systems define the first group and provide position information with respect to a-priori known salient landmark features or infrastructure. This also includes natural features. However, a common approach in robotics is to deploy artificial landmarks at known positions in order to ensure a sufficient landmark density within the robot's target workspace. Various promising approaches exist proposing illuminated [10] or reflective markers as well as augmented reality (AR) tags [62] as landmarks. Furthermore, absolute localization also includes external camera tracking systems such as the commercial systems by Vicon and Qualisys. However, focusing on self-localization systems, external systems are not within the focus of this review since they require a permanent communication link from the tracking system to the robot. This contradicts the goal of the underwater robot operating independently from external systems. The second group consists of systems that provide position information relative to a reference object but not in a global frame. This group also includes simultaneous localization and mapping (SLAM) techniques and visual-inertial odometry which both provide a pose relative to an initial pose. Note, non of these groups is exclusive and various techniques cannot uniquely be assigned to one or the other.

4.1.3 Acoustic Localization

Modern acoustic localization and navigation for underwater vessels can roughly be traced back to two events in the 1960s. Firstly, the search of the lost nuclear submarine USS Thresher in 1963 and, secondly, the underwater search and recovery of a nuclear bomb lost at sea in Spain in 1966 [81].

Acoustic localization approaches rely on the sound propagation between a reference object or an active transducer and the robot. The goal is to estimate the distance between both based on the sound pulse's time of flight (TOF). A common approach is to install multiple active anchor beacons at various known locations and determine the distances between the robot and each beacon to estimate the robot's absolute position.

Time of flight is commonly defined as the time span a sound pulse travels between the acoustic transmitter and the corresponding transceiver. Note, the speed of sound in water is about five magnitudes slower than the speed of electromagnetic propagation in air. Thus, all underwater acoustic systems are subject to non-negligible latencies which are proportional to the signal's TOF. Given a nominal speed of sound of 1,500 m/s and a distance of 20 m as common in confined volumes, we obtain 26 ms for the two-way travel of the sound pulse [133]. This time span constitutes an upper limit on the update rate. This update rate is further reduced if the distances to multiple reference beacons shall be determined in order to compute the spatial robot position.

On the upside, acoustic signals dissipate slowly in water compared to other physical principles such as vision or electromagnetics. This allows for potentially long transmission ranges. This is a strong advantage making acoustic approaches the method of choice for large-scale environments such as oceans. However, this property may pose challenges in strictly confined volumes where the acoustic energy has almost no possibility to dissipate into the open water. The consequences are often multiple re-reflections between volume walls which is also referred to as echoing. Another challenge of acoustic systems arises from so-called multi-path effects. These occur when the emitted signal arrives via multiple paths at the transducer which results in various measured TOFs. Hence, a trade-off exists between higher frequencies which dissipate faster and reduce multi-path effects and re-reflections in comparison to lower frequencies which allow for longer ranges. The sampling rate of acoustic systems can be increased by using multiple frequencies in parallel. However, this option considerably increases the complexity of the system. Thus, most systems rely on a single frequency range.

Overall, acoustic ranging with a-priori deployed active beacon anchors constitutes the most common approach for absolute localization. Within this category, the length scales of the setup are defined by the distances between the beacons, referred to as *baseline*. So-called long baseline (LBL) systems have a nominal baseline length around 2 km and are mostly used in oceanic settings [116]. Short baseline (SBL) systems have a nominal baseline length of 20-50 m which allows their deployment beneath the hull of a large mother vessel. The lower end is defined by ultra short baseline (USBL) approaches with a baseline length of 10 cm or less which can be realized in form of a compact hydrophone array. This compact setting allows to deploy the baseline hardware at a single spot, e. g. a ship hull, and, thus, facilitates handling. However, in most configurations, time-synchronization between the individual beacons and the robot transceiver is critical to accurately determine the sound pulses' TOF [42]. As a consequence, highly accurate clocks are required and constitute a remarkable cost driver within the system. Recent progress on chip-scale atomic clocks [20] may provide a promising approach towards this. Another drawback of beacon-based

acoustic systems is the required deployment of the beacon infrastructure prior to the experimental trials.

Doppler velocity logs (DVL) are used to estimate the vehicle's velocity relative to the ground. In a standard setup the Doppler velocity log sends multiple (often four) sonar beams towards the sea ground which are then reflected [141]. Depending on the vehicle speed, the frequency of the received signals changes. This is referred to as the *Doppler shift*. The frequency shift is proportional to the vehicle speed over ground. The advantage of this technology is that no infrastructure is required. However, it provides only estimates of the robot's velocity. Hence, it is often combined with other localization techniques, especially to stabilize the estimates of dead reckoning systems.

Sonar SLAM systems constitute the submerged pendant of LIDAR-based SLAM which is widely used in terrestrial environments [76]. As with all SLAM techniques the goal is to match the measurement – the sonar scan – to a map. This allows to estimate the vehicle's current position relative to this map. Moreover, the map is built and updated *simultaneously* based on the gathered sonar scans. The noticeable TOF of sound pulses underwater renders sonar SLAM system much less accurate than their vision-based twin. Thus, motion compensation is an important factor when matching map and scan. However, sonar-based SLAM has gained considerable research attention in recent years.

4.1.4 Electromagnetic Localization

Electromagnetic (EM) localization constitutes a niche method limited to short-range localization. As a consequence, it has not been widely covered in recent literature surveys on underwater localization techniques [133]. The concept of robot localization using the attenuation of electromagnetics exploits the strong attenuation of electromagnetic carrier signals waves in water. This attenuation is usually undesirable, e. g. for communication purposes. At the same time, it results in an ideally strong monotonic decaying signal strength which is subject to small noise. The suitable range of electromagnetic approaches depends on the emitted signal strength and the properties of the surrounding fluid volume. However, due to their limited effective range of usually $<5\,\mathrm{m}$ available electromagnetic localization concepts target applications in confined settings. Besides maneuvering in confined volumes such as small tanks, docking and manipulation scenarios have been identified as potential application areas [71].

In order to determine the distance between emitting anchor node and the robot's receiver, the received signal strength has to be determined. The distance can then be computed using an a-priori identified range sensor model which takes the received signal strength

as an input and outputs the corresponding distance between emitter and receiver. When designing the range sensor model, parameters capturing permeability, permittivity, and conductivity of the transmission medium have to be considered. Moreover, the carrier signal transmission from the emitting anchor node towards the robot receiver and, thus, the received signal strength strongly depend on the antenna designs on both sides.

In a localization setting, the electromagnetic carrier signals are emitted by multiple spatially distributed anchor nodes at known positions. Each node emits the carrier signal within a unique frequency channel. The receiver onboard the robot captures the complete overlayed power spectrum density. A Fourier transformation allows to identify and extract the individual anchor carrier signals based on their unique frequency. By exploiting the channel allocation, the received signal strengths and, thus, their corresponding estimated ranges can be assigned to the individual anchor nodes. This allows to determine the robot's location e. g. via triangulation. An advantage of electromagnetic localization is its passive nature on the robot receiver side. This allows simultaneous multi-robot localization without performance drops when additional robots are added. Another advantage is the possibility of simultaneous communication between anchor node and robot via the transmitted carrier signal waves.

4.1.5 Discussion on Localization Technologies for μAUVs

Each localization principle, reviewed in the previous subsections, comes with its individual strengths and flaws. Consider the *reference Scenarios I-III*, defined in Subsection 2.1.3 given the background of the reviewed localization technologies. We observe that there is no such technology available that fits all requirements simultaneously. In fact, both sides, localization principles and deployment scenarios, come with their individual specifications which need to be matched in order to optimize localization performance. For this reason, we consider the gain in terms of accuracy and robustness, given the invested effort for cost, infrastructure, etc.

In the following, we discuss the expected suitability of the reviewed localization technologies in the light of the defined reference Scenarios I-III. Based on this, the following Sections 4.2 to 4.5 propose components of an embedded localization framework for μAUVs which addresses the requirements imposed by the individual scenarios.

Considering *Scenario I*, the localization system needs to provide accurate position information at high update rates. A key challenge arises from the requirement that the localization needs to be robust against large and rapid changes of the vehicle's attitude as they occur during agile hydrobatic maneuvers in confined volumes. On the upside, it is

valid to assume high water quality and good visibility conditions. These requirements are well met by vision-based approaches. While EM-based localization may appear suitable as well, it is considerably less robust to large attitude changes since these have to be compensated by the transmission models. Acoustic localization approaches do not meet the posed criteria due to their low update rate and their sensibility to signal reflections from the confined volume walls. Given the absence of natural landmarks, this scenario is best addressed by a vision-based localization approach supplemented with fiducial makers. This concept is introduced in Section 4.3.

Scenario II poses similar requirements as Scenario I. However, the visibility conditions may be interfered by strong turbidity and are thus considered poor. Hence, visual localization is not expected to achieve a sufficient detection range and robustness. Due to the scenario's restricted length scales, acoustic approaches do not provide sufficient accuracy, as they would likely be subject to strong reflections and reverberation. On the upside, the scenario's requirements on robustness required for agile maneuvers are relaxed. This renders localization based on electromagnetic signal attenuation a promising approach given the relatively short ranges which have to be covered. Section 4.4 addresses this setting and proposes a promising localization scheme.

Scenario III comes with considerably larger length scales than Scenario I and II. This results in a larger volume which has to be covered by the localization system. As a consequence electromagnetic and visual approaches are rendered unsuitable for this setting, since the effort for deploying infrastructure quickly increases with the volume size. Furthermore, poor visibility conditions are expected to render vision-based systems infeasible. However, given the environment's large length scales, vehicle maneuvering is likely not as agile as demanded for in Scenario I. This results in a relaxed requirement on the localization update rate. Moreover, Scenario III-type settings such as offshore asset dumping sites usually do not possess large obstacles such as walls which may induce strong reflections. As a result, the requirements are well met by acoustic localization, ideally in combination with inertial navigation techniques which are used to leverage the localization update rate. Section 4.5 presents a study on acoustic localization using the low-cost acoustic *ahoi*-modem [103] in a marina setting.

Overall, we conclude that the selection of a single localization technology does not match with the wide range of various requirements posed by the defined reference scenarios. As a result, a modular approach is proposed which is based on technology-specific localization modules sharing common interfaces. Therefore, the following Section 4.2 discusses potential framework architecture designs in the context of flexibility and integration effort. This allows to adapt the μAUV's localization capabilities to the individual scenario requirements.

4.2 State Estimation for Hydrobatic Maneuvering

Control and planning tasks depend on reliable and accurate information on the vehicle's state. In this context, hydrobatic maneuvering poses strong requirements on the state estimation framework, since the vehicle state may change quickly. Thus, the state estimator must be robust enough to continuously provide smooth estimates on the vehicle state. This includes coping with a reduced quality of sensor measurements that results from the dynamic vehicle motions. An example is an onboard camera being subject to motion blur.

In a simple formulation, the vehicle state is defined by the robot's position and attitude as well as their corresponding velocities. The vehicle attitude can usually be obtained by measuring the orientation of the earth's gravitational vector using a 3-DOF accelerometer. However, tasks such as position feedback control, require information from a robot localization system. As a consequence, the state estimation architecture has to cope with various demands on its input side, namely multiple sensors come with individual update rates, noise, and time delays. The output side serves multiple modules, e. g. planning and control, which require state information at various rates. For instance, the feedback controller requires high-rate signals while the planning module may run at a lower rate.

4.2.1 Estimation Architectures

State estimation architectures typically follow either a loosely- or a tightly-coupled paradigm.

Loosely-coupled systems possess a modular hierarchical design where potentially multiple estimation algorithms are combined. On the low-level, submodules provide state estimates based on their directly connected sensors, e. g. vision and inertia sensors. On the top-level, a final estimation routine fuses the individual state estimates into a final state estimate while treating the submodules as black boxes. The advantage of this configuration lies in its modular design. This limits the complexity of the overall system and facilitates the adaptation to individual sensor configurations. Furthermore, the estimation modules allow a step-by-step development and testing of the single module.

In contrast, tightly-coupled approaches aim to fuse the raw measurements directly within a single filtering routine. This results usually into a more accurate state estimate. Furthermore, the integrated design makes it often easier to detect and compensate for failures of single sensors [141]. As a consequence, tightly-coupled systems achieve usually a slightly higher robustness than loosely-coupled approaches. However, the implementation effort is considerably higher than for loosely-coupled designs due to the high complexity of the fil-

tering routine. This high integration comes at the price that these systems make it harder to define distinct modules and interfaces. Consequently, later adaptations and extensions to the system, e. g. adding more sensors, are usually much more challenging.

4.2.2 Stochastic Sensor Fusion

Estimated quantities, such as the vehicle state, can be inferred from various sensory data. These data streams are subject to noise and latencies and usually available at different update rates. The process of combining these data streams to obtain an estimate of, for instance, the vehicle state, is commonly referred to as the *sensor fusion* problem. In modern robotic systems, stochastic filter algorithms have become the state-of-the-art when coping with multiple sensors [108, 120]. Prominent examples include the class of Kalman-Filters [63] and its derivatives, e. g. the extended Kalman filter (EKF). These filter algorithms allow to conveniently incorporate known uncertainties such as the sensor's measurement noise as well as the inaccuracies of the dynamic model.

When using multiple sensors, stochastic filters allow weighing the influence of signals of accurate sensors higher than signals from lower-quality sensors. In contrast to deterministic approaches, the stochastic concepts additionally provide metrics for the uncertainty of the estimated vehicle state. This uncertainty measure is particularly important for autonomous operations. It allows to provide statistical meaningful information which can be used to ensure that the robot operates within pre-defined safety boundaries. For instance, given high uncertainty in the robot position maneuvering close to obstacles should be less aggressive as if the vehicle state is accurately known.

4.2.3 HippoCampus State Estimation Framework

For the present case of the HippoCampus μAUV, the requirements render a loosely-coupled state estimation architecture favorable. In order to cope with the individual demands of the reference scenarios the flexible integration of various localization technologies is required, as discussed in Subsection 4.1.5. This is well addressed by the modular structure of a loosely-coupled approach. Thus, each localization technology can be capsuled inside an individual estimation module. Software-wise these modules are implemented onboard the vehicle's SBC units. We discuss these considerations in detail in the subsequent sections.

The technology-specific estimation modules share the vehicle's absolute position as a common interface. This facilitates switching between the individual localization principles. Note that the localization data is likely subject to non-negligible latencies which are immanent to the active localization technique, e. g. image processing times for the vision-based localization. These latencies have to be compensated in order to achieve optimal data quality for agile maneuvering control.

The low-level control algorithms (Section 3.3) are running on-board the PX4-based FCU, as described in Section 3.1. Thus, a time-delay compensation within the SBC-based localization module can hardly capture potential latencies on the SBC-FCU-communication. This potential limitation can be avoided by fusing the data stream from the localization module into a main state estimation framework, denoted by EKF* in Figure 4.1. This main module is implemented onboard the FCU and thus on the same computing platform as the low-level controller. It consists of an EKF which is running on a delayed fusion time horizon and a complementary filter that propagates the vehicle state to the current time instance. This allows the framework adequately address and compensate for the individual sensor latencies while providing the low-level controller with on-time state information.

Note that by using loosely-coupled approach, the proposed state estimation and localization framework can easily be adapted to individual use-cases additional to the defined reference scenarios. This includes the platform-independent deployment, such as on the commercial BlueROV robot, see also Subsection 4.5.3.

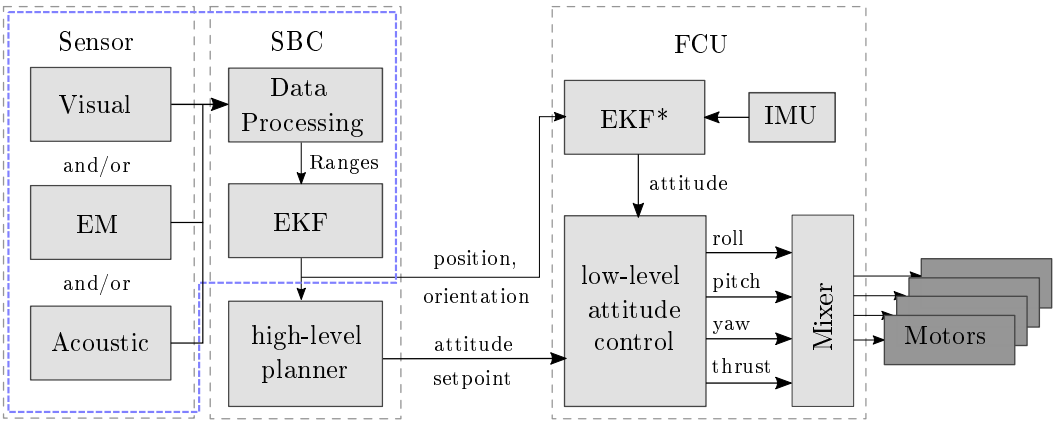


Figure 4.1: Proposed loosely-coupled state estimation architecture as part of the μAUV's GNC framework. Exchangeable localization modules (*blue box*) use various sensing principles to optimize the localization performance for the individual mission scenario.

4.3 Visual Marker-Based Underwater-Localization

In accordance with the requirements of Scenario I, this section presents a vision-based localization scheme for controlled confined environments, e. g. test tanks. By addressing these demands, the proposed localization approach provides a starting point for developing and benchmarking sophisticated underwater control schemes. Both aspects require a highly accurate position estimate. The following presents a localization system previously published by Duecker et al. in [23].

4.3.1 Methodology

We address the described problem setup by a visual localization approach that uses fiducial markers. While submerged environments usually lack salient features, controlled environments such as research facilities allow the deployment of artificial markers as reference points, as depicted in Figure 4.2.

The developed localization approach consists of three onboard components, namely a camera, an IMU, and an SBC, as depicted in Figure 4.3. Furthermore, the target volume, e. g. the research tank, has to be equipped with fiducial markers at known positions. While various marker systems exist, we select – without loss of generality – the AprilTag system [130]. Controlled confined environments usually provide sufficient ambient light and good visibility conditions. Thus, the system is passive on the μAUV-side since no onboard spotlights are required.

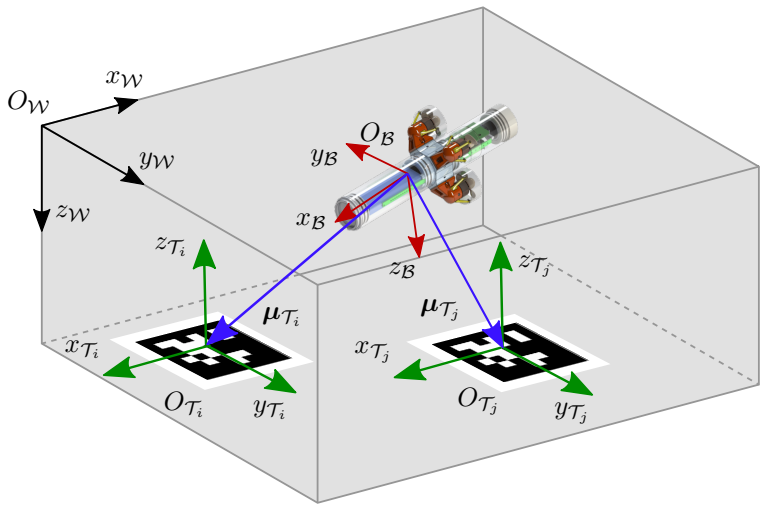


Figure 4.2: μAUV navigating through a confined volume using AprilTag-based self-localization for feedback control.

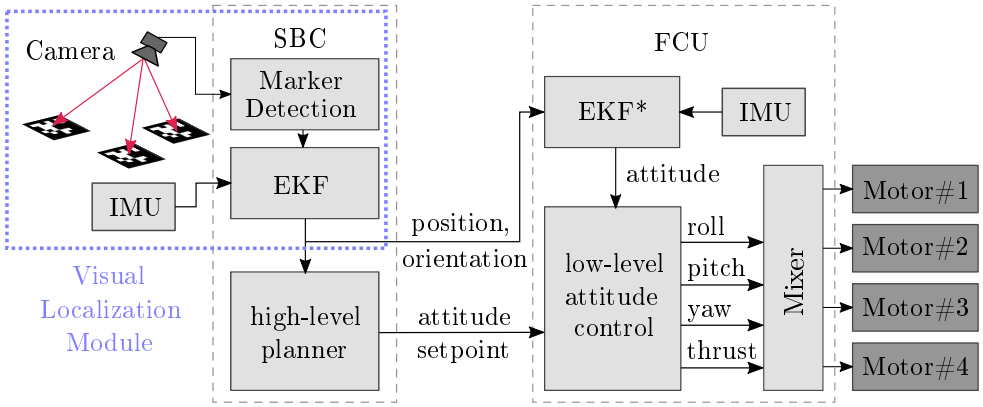


Figure 4.3: Vision-based AprilTag localization module embedded into the architecture of the HippoCampus GNC-framework.

An onboard wide-angle camera collects images of the vehicle's surrounding environment. The AprilTag detection algorithm developed by Malyuta et al. [77] is chosen to extract AprilTag markers from the image. The module computes the AprilTag positions and orientations relative to the camera sensor frame $\mathcal{C}$. This information is then fed into a sensor fusion algorithm which fuses the measured tags' position and orientation and provides uncertainty measures on both. Given the posed demand for high-frequent updates on the robot state, e. g. for agile robot control, inter-observation state predictions are required. Note, in micro underwater robotic state estimation, the robot's position and yaw-orientation constitute the most challenging parts and are thus the focus of the developed algorithm. While position estimates constitute a common problem, robust measurements of the yaw-angle are particularly challenging in confined volumes because relying on onboard magnetometer data is often infeasible. In practice, strong magnetic disturbances lead to severe local deviations of the magnetic field which is a common challenge in small (steel) tanks and renders the magnetometer-based yaw-signal often intractable for most navigation tasks.

In this work, we build on an EKF scheme which we extend by a dynamic measurement noise model. Note that the basic version of this filter is commonly used in the robotics community. Thus, adaptions to individual scenarios remain straightforward. However, the choice of other sensor fusion algorithms, such as particle filtering, is of course possible.

AprilTags for Underwater Localization

Following the definitions in Section 3.2 the position and orientation of the robot's body frame $\mathcal{B}$ in the world-fixed frame $\mathcal{W}$ is denoted by

$$\boldsymbol{\eta} = \left[x, y, z, \phi, \theta, \psi\right]^\top, \tag{4.1}$$

where x, y, z define the vehicle's position and the angles roll ϕ, pitch θ, and yaw ψ represent its orientation, as introduced in Subsection 3.2.1.

The AprilTag detection algorithm provides accurate distance information $d_{\mathcal{T}_i}$ between the body frame $\mathcal{B}$ and the i-th detected AprilTag marker $\mathcal{T}_i$, cf. Figure 4.2. Note, also the relative orientations can be directly obtained from the AprilTag-measurements. However, the signals on the roll and pitch orientation show an unreliable volatile behavior for various robot orientations. This is not acceptable for robust localization in agile maneuvering scenarios. Thus, we focus, first, on the estimate of the robot position $\boldsymbol{p}$ and subsequently provide an alternative approach for the roll and pitch orientation.

The measurement vector $\boldsymbol{\mu}$ can be modeled by the nonlinear observation function $\boldsymbol{h}(\boldsymbol{p})$. It consists of the distances $d_{\mathcal{T}_i}$ between the robot and the detected AprilTags $\mathcal{T}_i$ which are located at known positions and orientations inside the tank. The single tag-distance measurement reads

$$\mu_i = h_i(\boldsymbol{p}) = \sqrt{\left({}^{\mathcal{W}}\boldsymbol{p} - {}^{\mathcal{W}}O_{\mathcal{T}_i}\right)^2} \tag{4.2}$$

where ${}^{\mathcal{W}}O_{\mathcal{T}_i}$ is the known position of the i-th tag. Thus, the measurement vector $\boldsymbol{\mu}$ capturing N detected markers reads

$$\boldsymbol{\mu}_k = \left[\boldsymbol{\mu}_{k\mathcal{T}_1} \cdots \boldsymbol{\mu}_{k\mathcal{T}_N}\right] \tag{4.3}$$

with

$$\boldsymbol{\mu}_{k\mathcal{T}_i} = d_{k\mathcal{T}_i}. \tag{4.4}$$

Note that the dimension of μ changes dynamically with the number of markers N which are detected in every captured camera frame.

Position Estimation through Extended Kalman Filtering

State estimation by means of Kalman filtering consists of two steps. The *prediction* step computes the estimated evolution of the system state $\hat{\boldsymbol{p}}$ and its covariance matrix $\boldsymbol{\Sigma}$ over time. The *innovation* step fuses gathered measurements $\boldsymbol{\mu}$ to update the belief of the system state. Thereby, measurement updates decrease the state covariance while temporal predictions increase the belief uncertainty and, thus, the state covariance.

Prediction Step With regard to the temporal evolution of the system, we assume that the robot's motion model yields

$$\boldsymbol{p}_k = \boldsymbol{f}\left(\boldsymbol{p}_{k-1}\right) + \boldsymbol{w}_k, \tag{4.5}$$

where $\boldsymbol{p}_k$ is the robot position state at the time step k, and $\boldsymbol{w}_k$ the zero-mean Gaussian process noise vector with covariance matrix $\boldsymbol{Q}_k$. Notice motion models of various complexity are possible, ranging from a simple kinematic random walk model towards a complex kinematic model which incorporates hydrodynamic effects as derived in Section 3.2.

Hence, the temporal prediction of the position state and its covariance reads

$$\hat{\boldsymbol{p}}_k^{(-)} = \boldsymbol{f}\left(\hat{\boldsymbol{p}}_{k-1}^{(+)}\right) \quad \text{and} \tag{4.6}$$

$$\boldsymbol{\Sigma}_k^{(-)} = \boldsymbol{F}_{k-1}\boldsymbol{\Sigma}_{k-1}^{(+)}\boldsymbol{F}_{k-1}^{\top} + \boldsymbol{Q}_k, \tag{4.7}$$

respectively. Following Equation (4.5), the system's motion model is denoted by $\boldsymbol{f}(\cdot)$. Moreover, $\boldsymbol{F}_{k-1}$ denotes its Jacobian evaluated at $\hat{p}_{k-1}$ and reads

$$\boldsymbol{F}_{k-1} = \left.\frac{\partial}{\partial \boldsymbol{p}}\boldsymbol{f}\left(\boldsymbol{p}\right)\right|_{p=\hat{p}_{k-1}}. \tag{4.8}$$

The superscript (-) indicates a value gained in the filter prediction, before taking the measurements $\boldsymbol{\mu}$ into account, and the superscript $(+)$ denotes a value gained in the filter innovation step.

Innovation Step The measurement innovation update reads

$$\hat{\boldsymbol{p}}_k^{(+)} = \hat{\boldsymbol{p}}_k^{(-)} + \boldsymbol{K}_k\left(\boldsymbol{\mu}_k - \boldsymbol{h}(\hat{\boldsymbol{p}}_k^{(-)})\right), \tag{4.9}$$

$$\hat{\boldsymbol{\Sigma}}_k^{(+)} = \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_{\boldsymbol{p}_k}\right)\hat{\boldsymbol{\Sigma}}_k^{(-)} \tag{4.10}$$

and $\boldsymbol{H}_{\boldsymbol{p}_k}$ being the corresponding Jacobian matrix of the observation function $\boldsymbol{h}\left(\boldsymbol{p}\right)$ with form

$$\boldsymbol{H}_{\boldsymbol{p}_k} = \left[\nabla_{\boldsymbol{p}_k} h_1(\boldsymbol{p}_k)^{\top} \cdots \nabla_{\boldsymbol{p}_k} h_N(\boldsymbol{p}_k)^{\top}\right]^{\top}. \tag{4.11}$$

The Kalman-gain reads

$$\boldsymbol{K}_k = \hat{\boldsymbol{\Sigma}}_k^{(-)}\boldsymbol{H}_{\boldsymbol{p}_k}\left(\boldsymbol{H}_{\boldsymbol{p}_k}\hat{\boldsymbol{\Sigma}}_k^{(-)}\boldsymbol{H}_{\boldsymbol{p}_k}^{\top} + \boldsymbol{R}\left({}^{\mathcal{S}}\boldsymbol{q}\right)\right)^{-1}, \tag{4.12}$$

where the covariance matrix $\boldsymbol{R}({}^{\mathcal{S}}\boldsymbol{q})$ describes the sensor measurement noise. We use a dynamic measurement noise in order to incorporate the uncertainties which origin from the camera fish-eye lens,

$$\boldsymbol{R}\left({}^{\mathcal{S}}\boldsymbol{q}\right) = 1/\|{}^{\mathcal{S}}\boldsymbol{q}\,{}^{\mathcal{S}}\boldsymbol{e}_3 c_{\text{penalty}}\|, \tag{4.13}$$

where ${}^{\mathcal{S}}\boldsymbol{q}$ represents the measured tag position in x-y-z-camera sensor coordinates, ${}^{\mathcal{S}}\boldsymbol{e}_3$ is the ${}^{\mathcal{S}}z$-axis unit vector of the camera sensor frame, and c_{penalty} is a tuning parameter. Equation (4.13) dynamically increases the measurement noise $\boldsymbol{R}$ for individual tag measurements if they appear on the border of the camera's field of view. These measurements are usually affected by distortions in the image rectification process which we capture by higher measurement noise. This can be seen as a soft outlier penalty and considerably increases localization robustness during dynamic maneuvers when many tags appear at the boundaries of the camera's field of view. An example is the barrel roll maneuver depicted in Figure 4.6 which is discussed in detail in Subsection 4.3.3.

Since computing the pitch and roll orientation from IMU data is robust, we restrict our AprilTag-based estimate to position and yaw-orientation. Thus, the estimated state at time step k reduces to $\hat{\boldsymbol{p}}_k = [x, y, z, \psi]^\top$. We extend Equation (4.4) to include the yaw measurement which is considerably more stable than the roll and pitch measurements

$$\boldsymbol{\mu}_{k\mathcal{T}_i} = \begin{bmatrix} d_{k\mathcal{T}_i} & \psi_{k\mathcal{T}_i} \end{bmatrix}^\top. \tag{4.14}$$

Note that, image processing results in individual time stamp shifts. However, this requires running the filter algorithm on a delayed fusion horizon and was discussed in Section 4.2.

4.3.2 Hardware Setup

The visual localization module mainly consists of three components, depicted in Figure 4.4: a downward-looking wide-angle camera, an SBC and the FCU. The camera is a low-cost wide-angle RaspberryPi camera with an opening angle of 140°. The camera's effective opening angle is approximately 120° due to rectification. A Raspberry Pi 4 with 4 GB RAM is chosen as an onboard SBC running the vision processing and the EKF. We reduce the camera resolution to 640×480 px in order to trade-off accuracy against processing time. This allows the vision processing part of the pipeline to run at 10 Hz. Different camera mounting orientations are possible to adapt the module to the individual robot design.

The localization module is deployed in conjunction with the PixRacer FCU running the PX4-firmware and a dedicated attitude controller. This modular design allows to phys-

ically separate the low-frequent components e. g. the vision processing from the high-frequent components such as the attitude controller. Figure 4.4 shows the complete module in a 3D-printed rack attached to the HippoCampus μAUV. Overall, all system components fit into a total volume of approximately $90 \times 50 \times 30$ mm. The system is powered with 5 V and its power consumption is 9 W at full load which is comparatively low in comparison to the high power consumption of the four thrusters.

4.3.3 Performance Analysis

We evaluate the performance of the visual localization system in two experimental settings. First, we analyze the accuracy of the proposed embedded visual localization system. In a second experiment, we analyze the localization module in combination with the full GNC framework. Hereby, the goal is to push the system to its limits by driving hydrobatic maneuvers fully autonomously.

The experiments are conducted within a $4 \times 2 \times 2$ m freshwater tank. The tank is equipped with an array of 63 AprilTags with 400×250 mm spacing. Note that the AprilTag-array is used on multiple configurations where 63 Tags constitute a high-density setting to investigate an upper limit. However, we use coarser configurations out of the mounted 63 tag-array for the individual experiments. We choose the AprilTag-family 36h11 with tag side-lengths of 9.6 cm. Early testing shows that this tag size can still be detected at 3 m distance even using a low-cost RaspberryPi camera at 640×480 px. This makes the use of AprilTag markers also attractive for larger tanks. Larger marker-to-camera distances become feasible when using a higher resolution which, however, increases the computational load. For all experiments, the localization module is deployed onboard the HippoCampus μAUV, see Figure 4.4. The acryl tube hull has a wall thickness of 3 mm. Prior to the experiments, we calibrate the wide-angle camera underwater by means of standard checkerboard calibration. The reader is referred to the accompanying video attached to the paper by Duecker et al. [23] for an intuitive visual understanding.

Localization Performance In the following, we analyze the performance of the proposed localization module in order to demonstrate its suitability for accurate visual underwater localization and benchmarking tasks. Note, prior works, such as [21], focus their analysis of the AprilTag detection mainly on static or slow-motion settings with high-performance camera equipment. This analysis studies a *dynamic* tracking setup and, thus, supplements these prior works.

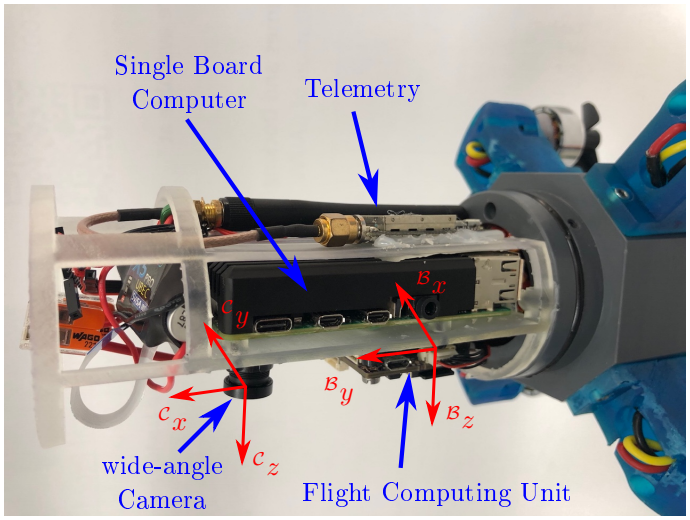


Figure 4.4: Visual localization module including the Pixracer flight computing unit mounted on the μAUV platform HippoCampus.

In order to benchmark our results against ground truth data, we mount the localization module onto an automated gantry system that can follow desired paths in x-y-z-direction with millimeter accuracy. It is of special interest to analyze whether the localization accuracy drops in certain areas of the tank volume. For this reason, we conduct the experiment with three different depths levels. These represent varying distances d between the visual localization module and the AprilTag-array. Furthermore, we examine three tag-array densities, namely a *fine* array with 31 tags, a *coarse* setup with only 23 tags, and the full array with *all* 63 AprilTags as an upper baseline.

The results for all configurations are summarized in Table 4.1. The cross-track error e_{cross} is defined between the gantry's path and the estimated position and is, thus, independent from latencies e. g. due to processing time or timestamp shifts. Moreover, the results show that for larger distances the tag-density has only a small influence on the localization accuracy. However, in the 0.4 m setup, the coarse array leads to an observable accuracy drop as only a few tags appear in the camera's field of view.

Figure 4.5 depicts the position estimate exemplary for the experimental setup with distance $d = 0.8\,\text{m}$ and the dense-array. It can be seen that the module consistently achieves high accuracy throughout the tank volume (RMSE $e_{\text{cross}} = 2.6\,\text{cm}$). The accuracy drops slightly in areas close to the tank walls (see $x \approx 0\,\text{m}$ and $x \approx 3.1\,\text{m}$) due to the reduced number of visible AprilTags in the camera's field of view. Moreover, the camera can be temporarily subject to motion blur which considerably reduces the number of detected tags. We refer to Subsection 4.3.3 for a detailed discussion of this effect. Note that the algorithm only detects markers that lie completely within its field of view. The analysis of the raw camera images shows that often multiple AprilTags lie only partly in the field

of view. This temporarily reduces the localization accuracy. Thus, a trade-off arises when designing the tag-array. Large markers enable longer detection ranges while small markers are more likely fully covered by the camera field of view. Depending on the scenario and the facility a combination of small and large markers may be most suitable. Moreover, we can observe that even tags visible in acute angles from the camera can be detected robustly as long as they lie completely within the field of view.

As expected, the average number of detected tags has a strong effect on the localization performance. Especially larger tanks allow a coarse tag distribution which still ensures that enough tags lie in the camera's field of view. However, the mounting of the tags can be adjusted depending on the task.

Finally, we examine the localization module's capability of estimating its yaw orientation with respect to the marker array. This is an important aspect for μAUVs since determining the robot's yaw orientation is usually strongly affected by distortions of the magnetic field leading to inaccurate measurements of the magnetometers. Therefore, we move the localization module along straight lines through the tank with its own orientation fixed. This allows to determine the deviation of the yaw estimate during the motion. We conduct multiple experimental runs with different fixed orientations of the localization module. The resulting standard deviation remains at less than 2° which can be seen as a drastic improvement in comparison to standard magnetic field-based concepts which are often rendered infeasible in confined tanks consisting of steel frames.

In summary, the experiment demonstrates the high accuracy of the localization module. Note that the above-reported results are achieved with low-cost components of less than USD 80.

Table 4.1: Cross-track error e_{cross} of the position estimated and the average number of AprilTags n_{tags} detected at each camera frame for various camera to tag-array distances d. The corresponding standard deviations are denoted in parentheses. The setup with gray background is depicted in detail in Figure 4.5.

Distance	$\boldsymbol{d}$	**1.3 m**	**0.8 m**	**0.4 m**
all	e_{cross}	3.1 (1.2) cm	2.4 (0.7) cm	3.8 (5.3) cm
	n_{tags}	23.1 (4.7)	12.3 (3.1)	4.2 (2.2)
fine	e_{cross}	3.1 (1.4) cm	2.6 (1.3) cm	5.5 (4.4) cm
	n_{tags}	11.6 (2.3)	6.1 (1.6)	1.8 (0.8)
coarse	e_{cross}	3.4 (1.8) cm	3.2 (1.6) cm	14.9 (20.5) cm
	n_{tags}	8.6 (1.7)	4.8 (1.3)	1.7 (1.1)

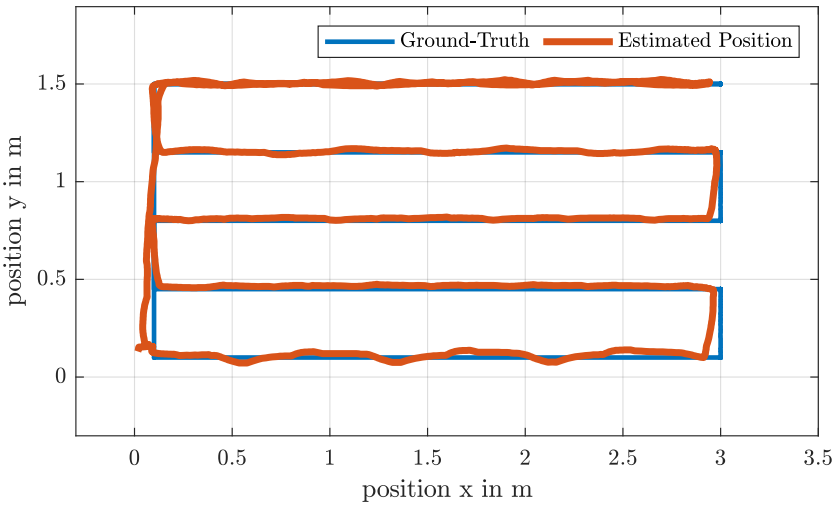


Figure 4.5: Accuracy of the localization module in comparison to ground truth.

Hydrobatic Path Tracking Control The second experiment in this performance analysis focuses on the robustness of the localization system and its integration into the GNC framework. In order to demonstrate the performance of the system, we design a challenging sequence of hydrobatic maneuvers. The sequence consists of an eight-shape path at varying depths and a screw-maneuver, where the robot turns 360° around its roll-axis while following the path, see Figure 4.6 and Figure 4.7. This hydrobatic maneuver is also referred to as *barrel roll* or *corkscrew* maneuver. The outer dimensions of the path are $2.5 \times 0.9\,\mathrm{m}$ with a total length of 4.4 m. We place fixed hoops as gates with 50 cm diameter as reference points on the robot's path. This allows to visualize the path containing envelope even without an external tracking system, see Figures 4.6 and 4.8.

The path is processed onboard the SBC by the high-level planning module with a 35 cm planning horizon. The recorded track is depicted in Figure 4.8 for four consecutive rounds at varying depths. Note that we pose an additional challenge to the system at the beginning of the experiment by throwing the robot through the air to an arbitrary starting

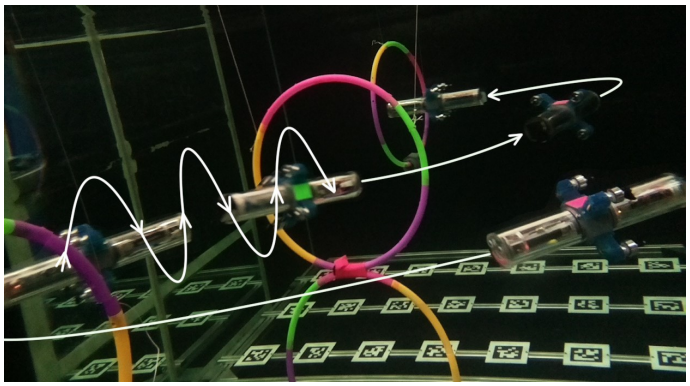


Figure 4.6: Hydrobatic path tracking through hoop-gates with the HippoCampus μAUV.

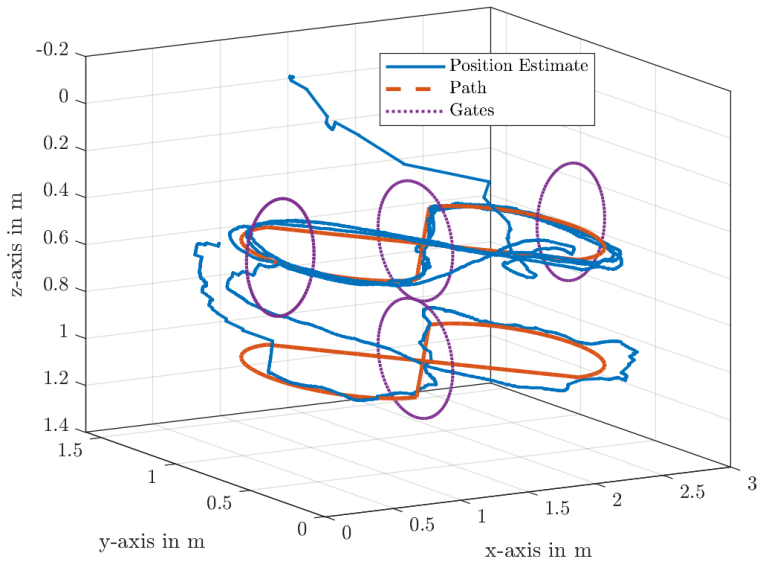


Figure 4.7: HippoCampus autonomously tracking an 8-shaped path through hoop-gates on two depth levels.

position. The localization system recovers within the first few time instances and estimates the correct position, see also the accompanying video in [23].

When driving hydrobatic maneuvers, accurate and robust information on the current robot position is required. Therefore, it is worth pointing out the tracking performance during the barrel roll-maneuver when the system cannot see or detect any marker for 2.7 seconds. During this period the controller relies purely on the EKF prediction. Notice that after regaining visual contact to the tags no noticeable jumps are induced by the Kalman update. Figure 4.9 shows the number of detected markers while following the path for one round, including the zero-sight segment (dark red) which corresponds to the barrel roll-maneuver. When examining the raw camera images, it can be seen that the camera greatly suffers from motion blur when the vehicle is performing agile maneuvers. However, tuning the camera exposure time can reduce this effect.

It is worth mentioning that the robot can track the path at all time instances with small cross-track errors (RMSE 7.2 cm). Notice that the HippoCampus μAUV is able to track its path at a considerably higher speed than similar state-of-the-art autonomous underwater robot platforms. During the experiment, the angular velocities reach up to $[4, 2.8, 1.5]$ rad/s for roll, pitch, and yaw rotations, respectively.

As a result, although we pose several challenges, these experiments demonstrate that the proposed visual localization approach in combination with the developed GNC framework is indeed capable of repeatably and accurately driving the μAUV along the path.

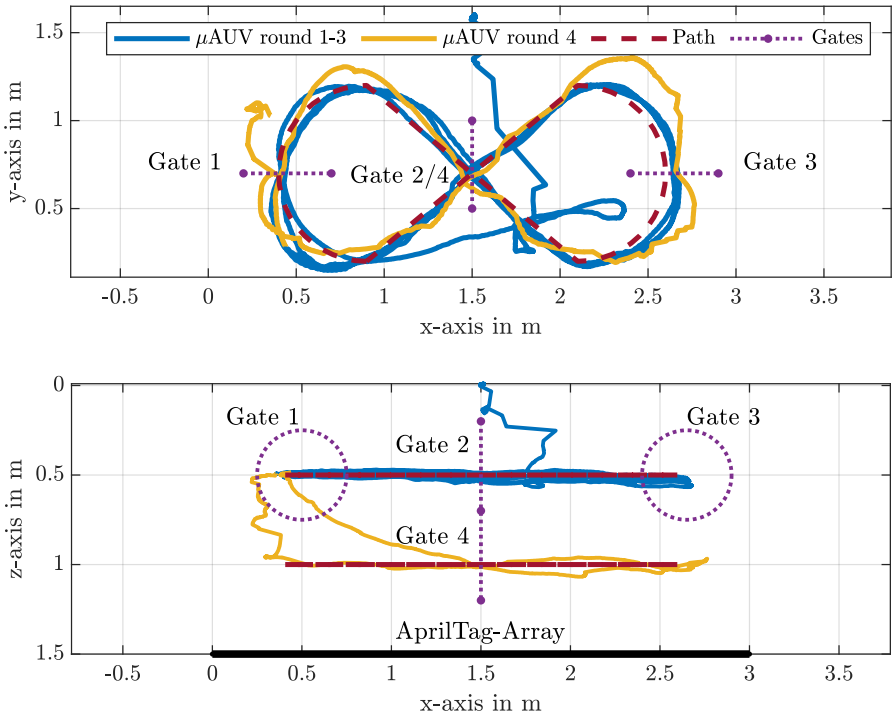


Figure 4.8: Tracking performance of the HippoCampus μAUV driving through four hoop gates.

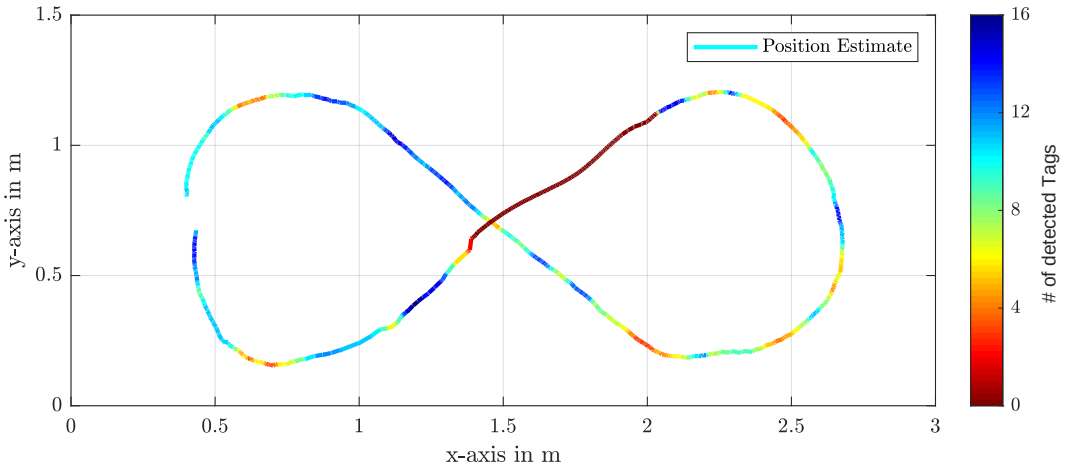


Figure 4.9: Number of detected AprilTags during the first eight-round. The red segment indicates zero tag-detections during the barrel roll maneuver, see also Figure 4.6.

4.4 Electromagnetic Underwater-Localization

Based on the discussion in Subsection 4.1.5, we consider a fleet of underwater robots autonomously navigating in a strictly confined volume with poor visibility conditions. Following the requirements of Scenario II, the dimensions of the strictly confined fluid volume are in the order of meters, as depicted in Figure 4.10.

In the following, we develop an embedded localization concept based on the attenuation of electromagnetic carrier signals. The presented concept has been previously published by Duecker et al. [25, 30] and extends the original works [70, 97, 98, 99] to the domain of μAUVs. The proposed system contributes a considerable reduction in complexity, size, and cost when compared to the approach proposed by the original studies.

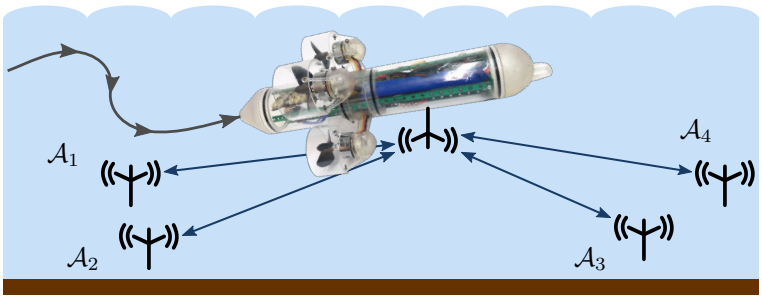


Figure 4.10: μAUV with on-board self-localization module using electromagnetic carrier-signal attenuation for path following and way-point tracking feedback control.

4.4.1 Methodology

The developed localization scheme aims to estimate the robot position based on distance measurements between the robot and multiple anchor beacons placed at known positions. These distances are derived from a range sensor model (RSM) which describes the attenuation of the electromagnetic field emitted by the beacons' antenna. For the following, we consider a setup of six transmitting anchors $\mathcal{A}_i$ at known positions ${}^{\mathcal{W}}O_{\mathcal{A}_i}$. These anchor beacons continuously emit EM carrier signals of constant magnitude. The setup is sketched in Figure 4.10. The anchor beacons are equipped with omnidirectional antennas to provide a circular-shaped attenuation pattern, as depicted in the contour plot in Figure 4.13. Each beacon has an assigned unique transmission frequency. The μAUV at position $\boldsymbol{p}$, measures the received signal strength (RSS). By applying a fast Fourier Transformation (FFT), the superposed measured signal is decomposed and the RSS value for each anchor beacon frequency can be determined.

Underwater Ranging-based on Electromagnetic Carrier Signals

The concept of electromagnetic-based underwater ranging exploits the consistent attenuation characteristic of electromagnetic waves in water. In simple form, this characteristic is described by

$$\frac{P_{\mathrm{Rx}}}{P_{\mathrm{Tx}}} = \mathrm{e}^{-\alpha R}, \tag{4.15}$$

where P_{Tx} and P_{Rx} are the wave powers on the transmission and receiver side and R is their spherical distance. Moreover, α is the attenuation constant defined by conductivity, permeability, and permittivity. Based on this, the design of an underwater range sensor model (URSM) aims to capture this characteristic by a function that takes the RSS as an input and provides an estimate of the corresponding distance

$$R = f_{\mathrm{URSM}}\left(RSS\right), \tag{4.16}$$

denoting the transmitting antenna and the robot receiver unit. With this goal in mind, the URSM extends the original Friis transmission model [38, 39] to the underwater domain. While the Friis transmission formula assumes a free space environment the URSM considers attenuation in a lossy medium. Therefore, the URSM includes influences due to the water's conductivity, permeability, and permittivity. The concept of the URSM has been originally proposed in [73] and was later explored in a series of subsequent publications [70, 97, 98, 99]. Moreover, we refer to [1] for an extensive study of electromagnetic attenuation characteristics for various combinations of signal wave frequency and water conductivity

In the context of underwater localization, we consider the range R_i between the mobile receiver unit, i. e. the robot, at position $\boldsymbol{p}$ and the anchor beacon $\mathcal{A}_i$ at location ${}^{\mathcal{W}}O_{\mathcal{A}_i}$. Its definition reads

$$R_i = \|\boldsymbol{p} - {}^{\mathcal{W}}O_{\mathcal{A}_i}\|. \tag{4.17}$$

In order to define the URSM based on the characteristic described by Equation (4.15) we formally define the RSS as the difference between the wave power on the receiver-side P_{Rx} and the originally emitted wave power on the emitter-side P_{Tx}. This difference is range-dependent. Thus, we model the RSS as a function of the range R_i between the anchor beacon $\mathcal{A}_i$ and the receiver, reading

$$RSS_i = P_{\mathrm{Tx},i} - P_{\mathrm{Rx},i} = -20\,\log_{10} R_i - 20\,R_i\alpha_i \log_{10}\mathrm{e} + \Gamma_i, \tag{4.18}$$

where α is again the attenuation constant and Γ denotes an offset factor which captures antenna and environmental influences. In Park et al. [100] the authors provide an explicit derivation for both URSM parameters. Moreover, the authors conduct a series of experiments in fresh and seawater to validate Equation (4.18). Both URSM parameters can be determined explicitly. This requires accurate knowledge on the antenna gains and the parameters for permittivity, conductivity, and permeability. However, as an alternative, the URSM parameters can be determined via calibration. For this, RSS values have to be measured at various distances R. Subsequently, the parameters α and Γ can then be easily computed via a nonlinear least-squares formulation. See Figure 4.11 for an illustration of the measured RSS values and the fitted URSM. Note that in practice, the calibration approach is usually the preferred approach. In fact, it is often tedious to sufficiently accurately identify the parameters required for an explicit calculation of α and Γ.

While the URSM defined in Equation (4.18) computes the RSS based on a given range R, localization systems usually rely on the inverse relation. Hence, we invert Equation (4.18) to compute the range R for a measured RSS-value. This inverse reads

$$R_i = \frac{1}{\alpha \log_{10} \mathrm{e} \ln 10} \cdot W\left(\alpha \log_{10} \mathrm{e} \ln 10 \exp\left[\frac{-\ln 10}{20} RSS_i - \frac{\Gamma \ln 10}{20}\right]\right), \tag{4.19}$$

where $W(\cdot)$ denotes the Lambert-W function which is the inverse function of

$$f(x) = x \exp(x), \tag{4.20}$$

$$f^{-1}(x) = f(y) = W(y). \tag{4.21}$$

Depending on the final localization algorithm the explicit computation of the range using Equation (4.19) may not be necessary. For instance, stochastic state estimators such as the EKF schemes only rely on the forward measurement model, i. e. the URSM as defined in Equation (4.18).

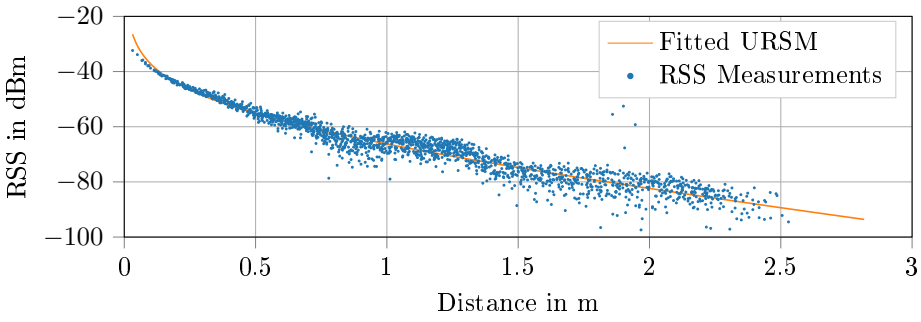


Figure 4.11: Fitted underwater range sensor model (URSM) based on the measured received signal strength (RSS). Measurements are taken in a $4 \times 2\,\mathrm{m}$ freshwater tank.

Signal Strength Compensation for Dynamic Vehicle Attitude Changes

It is worth mentioning that the URSM defined by Equation (4.18) describes the electromagnetic attenuation pattern of omnidirectional antennas. More specifically, the model yields for upright and parallel antenna pairs which lie mostly within the same horizontal plane. This restriction may become a limitation for applications with small-scale underwater robots. While medium- and large-scale robots usually move with small pitch and roll angles this is not necessarily the case for micro underwater robots. In order to address this limitation, an extension to the URSM introduced in Equation (4.18) is presented which allows for non-negligible pitch and roll orientation. In fact, we exploit that Equation (4.18) provides the maximum RSS-value due to the assumption of parallel antennas. Thus, we follow an approach analog to [70] and introduce parameterized loss factors that account and compensate for signal strength losses. These signal strength losses result for instance from non-parallel antenna pairs. This may occur during maneuvering with large roll and pitch angles.

The generalized form of the 3D omnidirectional attenuation model reads

$$\begin{aligned}\frac{P_{\mathrm{Rx}}}{P_{\mathrm{Tx}}} = e_{\mathrm{T}} e_{\mathrm{R}} \times \left(1 - \|\Gamma_{\mathrm{T}}^2\|\right)\left(1 - \|\Gamma_{\mathrm{R}}^2\|\right) \\ \times L\left(R\right)_{\text{Lossy Medium}} \times L\left(R\right)_{\text{Radiation}} \\ \times L\left(\phi\right)_{\text{Polarization}} \times L\left(\Theta, \theta\right)_{\text{Elev.,and Incl.}} ,\end{aligned} \tag{4.22}$$

where $e_{\mathrm{T}} e_{\mathrm{R}}$ represent the antenna efficiencies and $\left(1 - \|\Gamma_{\mathrm{T}}^2\|\right)\left(1 - \|\Gamma_{\mathrm{R}}^2\|\right)$ covers the impedance mismatch caused by the penetration of changing mediums. Moreover, $L\left(R\right)_{\text{Lossy Medium}}$ and $F\left(R\right)_{\text{Radiation}}$ describe the losses due to the signal attenuation depending on the distance between both antenna nodes. Furthermore, $L\left(\phi\right)_{\text{Polarization}}$ captures a reduction in signal strength due to polarization losses. This is the result if the effective antenna area is reduced due to a rotation of ϕ, see Figure 4.12. Finally, the loss factor $L\left(\Theta, \theta\right)_{\text{Elev./Incl.}}$ covers losses due to elevation and inclination according, as sketched in Figure 4.12.

The derivation and identification of these terms are challenging tasks in their own which we omit here for the sake of brevity. However, we refer the reader to the detailed discussion by Kwak et al. in [70].

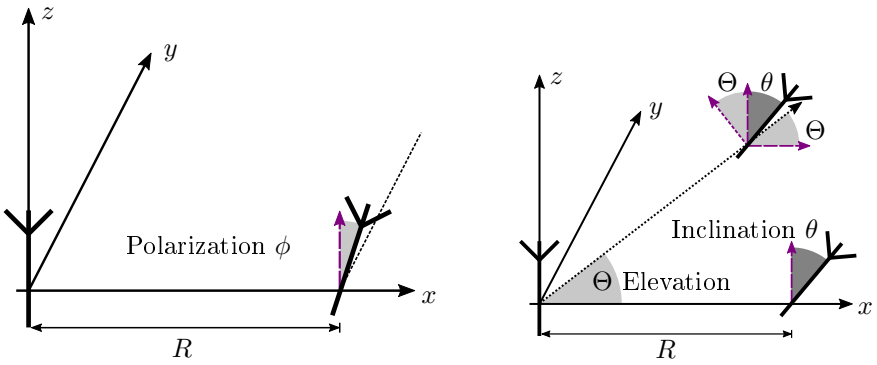
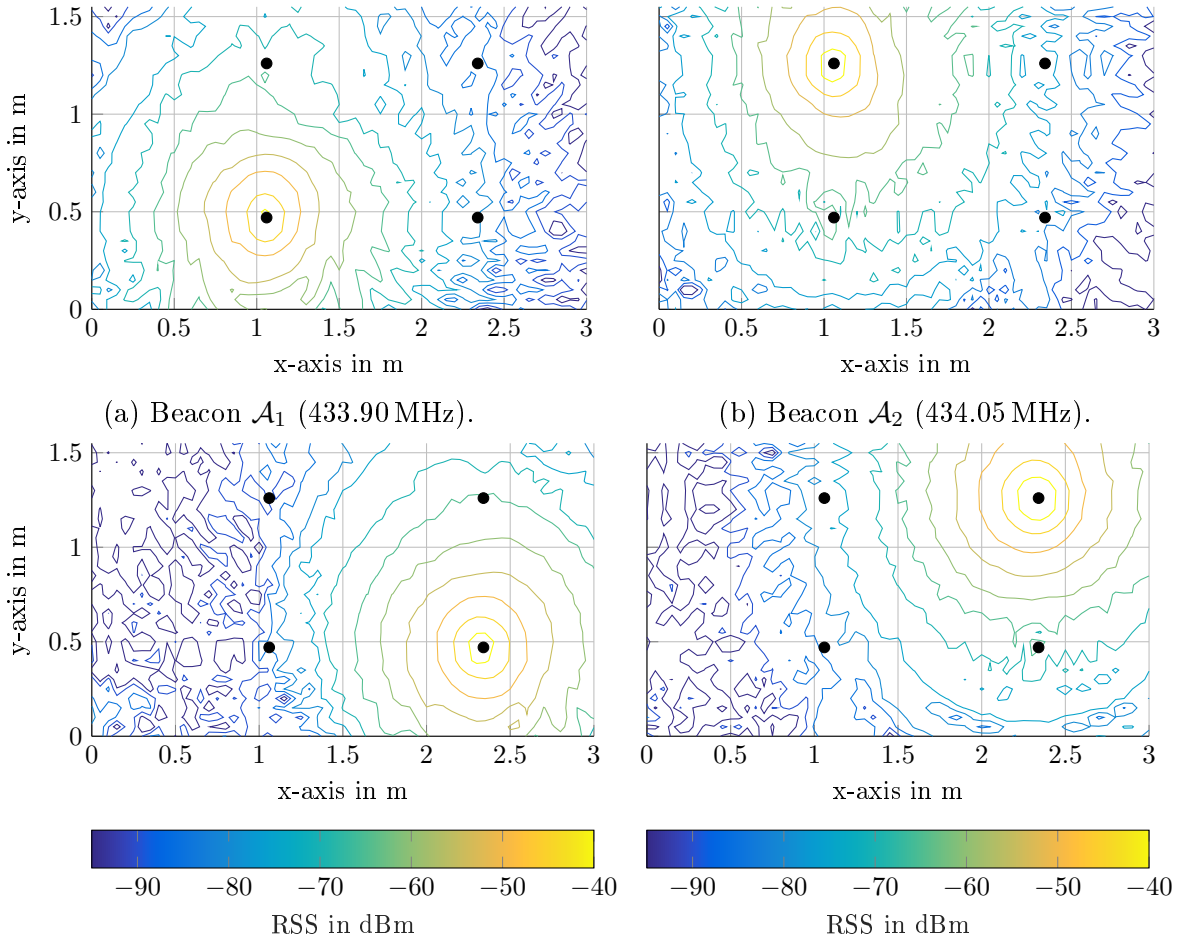


Figure 4.12: Visualization of antenna configurations that result in a reduction of the RSS. *Left:*The loss factor due to polarization angle ϕ. *Right:* The relative antenna orientations caused by the inclination angle θ (*bottom*) and the combination of inclination θ and elevation angle Θ (*top*).



(a) Beacon $\mathcal{A}_1$ (433.90 MHz).

(b) Beacon $\mathcal{A}_2$ (434.05 MHz).

(c) Beacon $\mathcal{A}_3$ (434.20 MHz).

(d) Beacon $\mathcal{A}_4$ (434.35 MHz).

Figure 4.13: Contour plots of RSS values measured over 5 s on each grid point at the anchor beacons' frequencies. Black markers depict the positions of the anchor beacons $\mathcal{A}_i$.

Transmitter Beacon Identification via Channel Allocation

Spherical localization in the horizontal plane requires at least three distance measurements. For the electromagnetic localization, these distances are computed from the RSS measurements which, hence, have to be assigned to the corresponding beacons. Various techniques are available to identify the emitting anchor beacon $\mathcal{A}_i$.

Time-scheduling approaches transmit the signals sequentially. If the transmission order of the beacons is known to the receiver unit, the RSS value can be assigned accordingly. However, this requires clock synchronization between the receiver and the emitting beacons which is challenging underwater [42]. Furthermore, the system's update rate is limited due to the latencies induced by the sequential transmission.

An alternative approach is to identify the emitting beacons by means of channel allocation [70, 100]. Each beacon continuously emits its carrier signal on a pre-defined unique frequency channel. This results in a superposed power spectrum of all signals. The received unit measures this signal spectrum and performs an FFT in order to compute the power spectrum density (PSD). The PSD is exemplarily depicted in Figure 4.14. As can be seen in the figure, the FFT provides RSS values that can be directly assigned to the anchor beacons. Given the identified beacon, the corresponding URSM can be selected to compute the range between beacon and receiver. In comparison to time-scheduling, this approach comes with the advantage of a high update rate, since all beacons transmit their signals simultaneously. Furthermore, the FFT provides access to all RSS values at each processing cycle. This facilitates data fusion and state estimation as no individual latencies need to be considered.

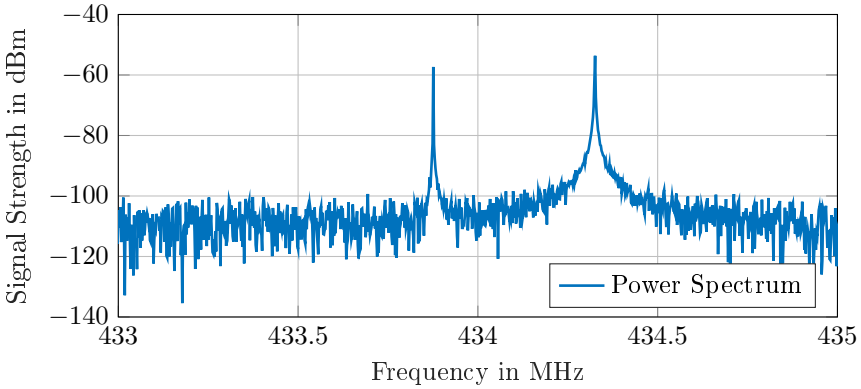


Figure 4.14: Power spectrum density received by the receiving unit. Two beacons emit signals at 433.9 MHz and 434.3 MHz. Their corresponding RSS values can be easily identified and the ranges can be computed from the peak values $RSS_1 = -57.34\,\text{dBm}$ and $RSS_2 = -53.65\,\text{dBm}$ by using the URSM.

Position Estimation through Extended Kalman Filtering

The prior study [25] and the experimental results indicate that the multimodal nature of the observations has only a negligible influence on the accuracy of the position estimate. Thus, we remain with the general EKF-based estimation scheme as presented in Subsection 4.3.1. However, we introduce and derive the relevant models and observation functions needed to adapt the EKF-scheme to electromagnetic-based ranging.

The prediction step is analog to Subsection 4.3.1 omitted here for the sake of brevity with sticking to the same notation. Within the innovation step, the robot position is updated through the vector of measured RSS values

$$\boldsymbol{\mu} = [\mathrm{RSS}_1 \cdots \mathrm{RSS}_N]^\top, \tag{4.23}$$

where RSS_i denotes the signal strength received from the anchor beacon $\mathcal{A}_i$ at position ${}^{\mathcal{W}}O_{\mathcal{A}_i}$. By means of the URSM in Equation (4.18) we can write the corresponding nonlinear logarithmic measurement model h_i as

$$\begin{aligned} h_i(\boldsymbol{p}_k) &= \mathrm{RSS}_i(\boldsymbol{p}_k), \\ &= -20\,\log_{10}\|\boldsymbol{p}_k - {}^{\mathcal{W}}O_{\mathcal{A}_i}\| - 20\,\|\boldsymbol{p}_k - {}^{\mathcal{W}}O_{\mathcal{A}_i}\|\alpha_i \log_{10} e + \Gamma_i. \end{aligned} \tag{4.24}$$

Hence, the estimated measurement of the RSS value yields

$$\hat{\mu}_{i,k} = h_i(\hat{\boldsymbol{p}}_k). \tag{4.25}$$

We assume that the measurements are subject to additive zero-mean Gaussian white noise. Furthermore, we assume that measurement noise is independent between the channels and can be described by the diagonal measurement noise matrix $\boldsymbol{R}$. We define the vector of the simultaneously gathered RSS measurements as

$$\boldsymbol{h}(\cdot) = [h_1(\cdot) \cdots h_N(\cdot)]^\top, \tag{4.26}$$

where $h_i(\cdot)$ is the observation function for anchor beacon $\mathcal{A}_i$. Thus, the corresponding Jacobian matrix has the form

$$\boldsymbol{H}_{\boldsymbol{p}_k} = \left[\nabla_{\boldsymbol{p}_k} h_1(\boldsymbol{p}_k)^\top \cdots \nabla_{\boldsymbol{p}_k} h_N(\boldsymbol{p}_k)^\top\right]^\top, \tag{4.27}$$

where

$$\nabla_{\boldsymbol{p}_k} h_i(\boldsymbol{p}_k) = \left[-\frac{20}{\ln 10 \cdot \|\boldsymbol{p}_k - {}^{\mathcal{W}}O_{\mathcal{A}_i}\|^2} \cdot [\boldsymbol{p}_k - {}^{\mathcal{W}}O_{\mathcal{A}_i}]^\top - 20\alpha_i \log_{10} e \frac{[\boldsymbol{p}_k - {}^{\mathcal{W}}O_{\mathcal{A}_i}]^\top}{\|\boldsymbol{p}_k - {}^{\mathcal{W}}O_{\mathcal{A}_i}\|}\right]_{\boldsymbol{p}=\hat{\boldsymbol{p}}_k^{(-)}}.$$

In order to adjust the prediction in accordance to the measurement prediction error $\boldsymbol{\mu}_k - \boldsymbol{h}(\hat{\boldsymbol{p}}_k^{(-)})$ the Kalman-gain $\boldsymbol{K}_k$ is computed as

$$\boldsymbol{K}_k = \boldsymbol{\Sigma}_k^{(-)} \boldsymbol{H}_{\boldsymbol{p}_k} \left(\boldsymbol{H}_{\boldsymbol{p}_k} \boldsymbol{\Sigma}_k^{(-)} \boldsymbol{H}_{\boldsymbol{p}_k}^\top + \boldsymbol{R}(\boldsymbol{\mu}) \right)^{-1}. \tag{4.28}$$

The state update and its covariance can be computed as follows

$$\hat{\boldsymbol{p}}_k^{(+)} = \hat{\boldsymbol{p}}_k^{(-)} + \boldsymbol{K}_k \left(\boldsymbol{\mu}_k - \boldsymbol{h}(\hat{\boldsymbol{p}}_k^{(-)}) \right), \tag{4.29}$$

$$\boldsymbol{\Sigma}_k^{(+)} = \left(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_{\boldsymbol{p}_k} \right) \boldsymbol{\Sigma}_k^{(-)}. \tag{4.30}$$

Note, when studying Figure 4.11, we observe that with increasing distances the signal to noise ratio of the RSS decreases. As a consequence, the corresponding measurement model represented by the URSM in Equation (4.18) becomes less reliable. Additionally, the influence of signal reflections, e. g. from the tank walls, increases. We account for these effects by proposing a dynamic measurement noise covariance $\boldsymbol{R}(\boldsymbol{\mu})$. This matrix function dynamically increases the expected measurement uncertainty as the measured RSS values decreases. Furthermore, this allows to perform outlier rejection based on the dynamic measurement uncertainty. For example, it is suitable to reject a single RSS measurement which deviates strongly from the value expected from the URSM. We exemplary refer to Figure 4.11 and specifically the measured RSS value at $R = 2\,\mathrm{m}$.

4.4.2 Hardware Setup

The hardware setup consists of two elements, emitting beacons at known positions and a mobile receiver unit that is mounted to the robot.

Receiver Unit

Since the beacons emit their signals at various frequencies, the power spectrum has to be computed on the receiver side and ideally fully embedded onboard the robot. The receiver unit consists of three main components: an antenna, a frequency spectrum processing unit, and a computing unit. In its standard configuration, this setup is not suitable for small-scale underwater robots such as μAUVs. The main bottleneck from a micro robot perspective rises from the dimensions and cost of the frequency spectrum processing unit. The original works [70, 98, 99] use an offboard full-fledged spectrum analyzer which is able to compute the PSD and corresponding RSS values at an update rate of 1 kHz. However, these high-performance spectrum analyzers are mostly designed for 19"-racks and their

price range lies usually well beyond USD 10 k. These two aspects render this configuration infeasible for embedded deployment onboard a μAUV-platform.

As an alternative, Duecker et al. [25] propose a low-cost embedded design. It consists of a commercial 2 dBi dipole wifi antenna, a software defined radio (SDR)-capable Digital Video Broadcasting-Terrestrial (DVB-T) USB dongle, and an SBC. The components are depicted in Figure 4.15 (*left*). The system is designed to fit the HippoCampus μAUV, as depicted in Figure 4.15 (*right*).

The NooElec^{TM} NESDR Mini DVB-T dongle can process analog electromagnetic signals within a range of 24-1 700 MHz (USD 25). After demodulation and analog-digital conversion, the signals are transmitted as a data sample to the SBC. The SBC runs an FFT on the samples to identify the corresponding transmitter beacon and its corresponding RSS value in the power spectrum. Additionally, the SBC runs the filtering algorithm, e. g. an EKF, to estimate the robot position based on the obtained RSS values. By the combination of DVB-T dongle with an SBC, e. g. a Raspberry Pi 3, is capable of running signal processing and the EKF measurement update cycle at 10-15 Hz.

Anchored Beacons

The transmitter beacons consist of the Radiometrix^{TM} USX2 module to generate EM carrier signals in the 433 MHz band. As the wavelength is shortened underwater, the wavelength of the 433 MHz band used underwater sufficiently matches the characteristic wavelength of wifi systems in air, which operate at 2.4 GHz. This allows to use of off-the-shelf wifi-antennas on the transmitter and receiver side. A setup consisting of six beacons mounted to a submerged rack is depicted in Figure 4.17.

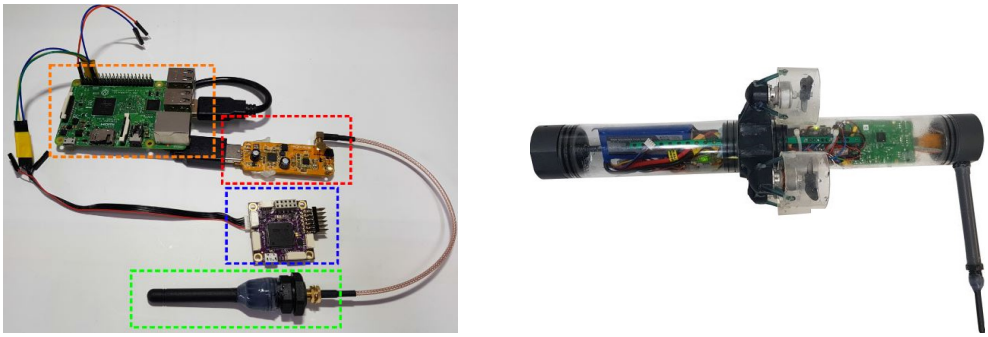


Figure 4.15: The assembled components of the embedded localization system (*left*): Antenna (green box), DVB-T dongle (red box), and the Raspberry Pi SBC (orange box) which is connected to the Pixracer flight controller (blue box). HippoCampus μAUV housing the EM-based localization system with external antenna (*right*).

4.4.3 Performance Analysis

In the following, we study the performance of the embedded EM localization approach using a low-cost DVB-T dongle and an SBC. The performance study consists of two parts, the localization against ground truth and an experimental demonstration of position feedback control with the fully embedded EM-localization system onboard the micro robot HippoCampus. We discuss these findings in the context of the original publications by Park et al. [70, 98], in which the authors study an offboard localization system with impressively high localization accuracy. In fact, the authors report positioning errors in sub-centimeter range. However, as discussed in Subsection 4.4.2, the original works' setup is not suitable for μAUV systems due to the setup's complexity, dimension, and cost. Thus, the comparison of the proposed embedded design against the original works is of special interest. This results in the following claims: (i) the low-cost approach to electromagnetic localization is indeed able to provide sufficient accuracy to allow position feedback control with a μAUV in a confined volume. (ii) the proposed architecture provides sufficient robustness for operation within restricted confined volumes.

The first experiment focuses on the performance of the embedded localization combination with a way-point tracking algorithm. For this reason, we deploy a restricted configuration of the HippoCampus μAUV, which has constrained degrees of freedom. The experiment is conducted in a public swimming pool (side lengths 7 m by 7 m, depth 1.3 m).

In the second experiment, we examine the system in a considerably smaller test tank ($4\,\mathrm{m} \times 2\,\mathrm{m} \times 1.5\,\mathrm{m}$) in the presence of various disturbances including interferences of the electromagnetic field within the tank and uncompensated attitude offsets, see Figure 4.16.

Both experiments are conducted using the HippoCampus μAUV platform. The way-point tracking controller is implemented on the PX4-based Pixracer FCU [79] which supplements the low-rate localization estimate with high-frequency inertial measurements unit and magnetometer data.

Experimental Setup In both experiments, the introduced localization system with six active beacons is used, see Figure 4.16. The positions of the beacons and their frequencies are listed in Table 4.2. The beacons are powered via 10 m RG-58 coaxial cables from a base station equipped with six Radiometrix$^{\mathrm{TM}}$ USX2 modules.

The HippoCampus μAUV is equipped with the embedded localization module, as described in Subsection 4.4.2, using a Raspberry Pi 3 as an SBC and the NooElec NESDR DVB-T dongle. The localization antenna is mounted outside the vehicle to avoid disturbances from the μAUV hull, see Figure 4.15 (right) for a photo of the setup.

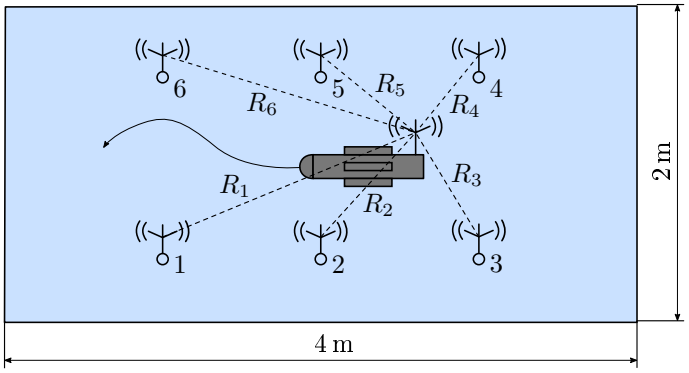


Figure 4.16: Test setup for electromagnetic-based self-localization for μAUVs consisting of the on-board receiver and six anchor beacons.

Table 4.2: Specification of the anchor beacons inside the test basin. The positions are determined with centimeter accuracy.

Beacon	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$	$\mathcal{A}_5$	$\mathcal{A}_6$
Position $[x, y]$ in m	[0.0, 0.0]	[1.0, 0.0]	[2.0, 0.0]	[2.0, 0.8]	[1.0, 0.8]	[0.0, 0.8]
Frequency in MHz	434.00	434.15	434.30	434.45	434.60	433.90

Way-Point Tracking Control – Constrained Configuration In this first experiment, we study the system performance in a constrained configuration to examine the ability of the embedded localization system to provide sufficiently accurate positioning data to allow for way-point tracking. In order to minimize external disturbances on the localization system such as EM reflections from the walls, the experiment is conducted in a swimming pool with length dimensions of 7 m by 7 m and a depth of 1.3 m. The beacons are placed with a distance of at least 1.5 m away from the pool walls. This has been shown to be sufficient to avoid EM reflections. Note, in this experiment we aim to study the performance of the localization and the way-point tracking algorithm independent of the performance of the pitch and roll controllers. Thus, we attach the μAUV to a free-moving swimming rig. This rig constantly keeps the robot at the desired depth, with zero roll and zero pitch angle.

In order to demonstrate the practicality of our integrated navigation and control system a sequence of way-points is defined, see Table 4.3 and Figure 4.18. The μAUV's objective is to follow the sequence in counterclockwise order. This constitutes a challenging task as the way-points lie close to each other. Thus, switching to the next way-point requires large changes in the robot's yaw orientation. When the μAUV approaches the vicinity of 0.2 m around a way-point, it is considered as successfully reached and the subsequent way-point

Figure 4.17: Setup of the experimental trials in the public swimming pool in Hamburg-Wilhelmsburg: The HippoCampus μAUV is autonomously navigating relying on signal strength measurements from six beacons. The driven track is plotted in Figure 4.18. During the experiment the μAUV is mounted to a swimming rig to stabilize the depth. Underwater view (*left*) and top view (*right*).

is set as a new target way-point. The sequence is repeated six times to demonstrate the reproducibility of the way-point tracking.

Figure 4.18 shows the path driven by the μAUV when following the way-point sequence for six rounds. The position control error e_p and the yaw-orientation error $e_{r,\psi}$ are depicted in Figure 4.19. After a new way-point is assigned to the robot, comparatively large orientation errors occur, because the μAUV's orientation has not been aligned with the new way-point yet. Due to its quadrotor-inspired design, the HippoCampus μAUV is able to change its orientation without forward motion. Note that we prioritize the orientation error over the position error analog to Section 3.3 which leverages the vehicle's maneuverability. Thus, when approaching the vicinity of the target, forward thrust is reduced and the robot slows down. We refer to time steps $t = 5\,\mathrm{s}$ and $41\,\mathrm{s}$, where the orientation error is minimized first, while the position error remains almost constant.

The path depicted in Figure 4.18 demonstrates the reproducibility of the way-point tracking. Note that the displayed path is reconstructed from the robot's logged estimated position and can only be analyzed qualitatively, as no ground truth, i. e. from a motion capture system, is available. We refer to the accompanying video of [30] for a bird's eye view of the experiment. Overall, the control errors show systematic behaviors, which are likely caused by distorted position estimates. These distortions can be minimized by a more accurate calibration process or compensated by a powerful guidance module. In summary

Table 4.3: Way-Point positions relative to beacon $\mathcal{A}_1$.

Way-Point	**#1**	**#2**	**#3**	**#4**
[x, y] in m	[0.3, 0.2]	[1.7, 0.2]	[1.7, 0.7]	[0.3, 0.7]

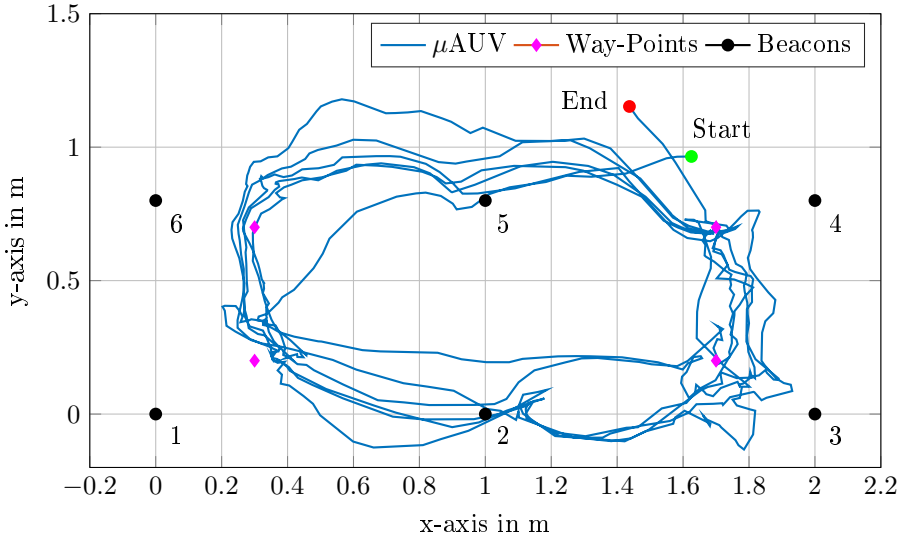


Figure 4.18: HippoCampus μAUV following a sequence of four way-points in counter-clockwise order for six rounds.

and with regard to claim (i), we observe that the presented embedded localization system is able to provide sufficiently accurate positioning information to allow for reproducible way-point tracking.

Way-Point Tracking Control in a Confined Tank A second experiment is conducted to examine the performance and robustness of the free-floating μAUV in the challenging environment of a restricted confined water tank with the side dimensions of 4 m by 2 m and a water depth of 1.5 m. This setup is similar to the one sketched in Figure 4.16. Note that this test tank contains a massive steel structure which induces strong local disturbances on the μAUV's onboard magnetometer.

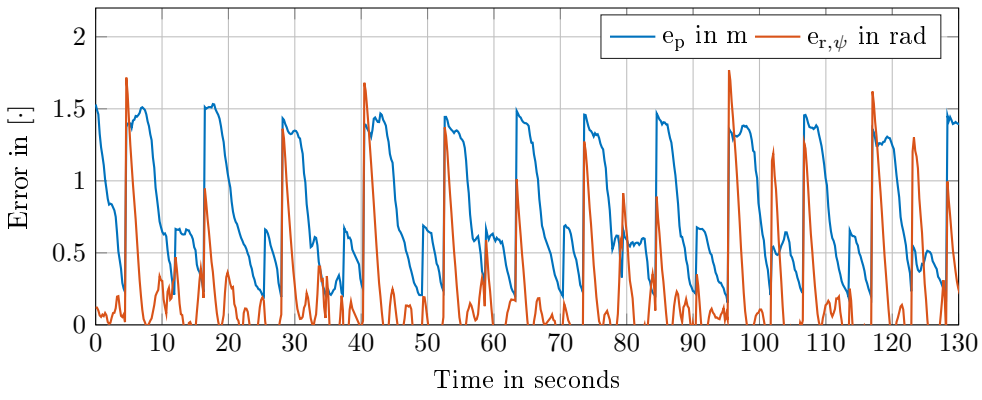


Figure 4.19: Position and yaw orientation errors for six rounds.

In order to allow a comparison with the results from the previous experiment, the relative position of the beacons and the way-points remains unchanged. However, due to the small size of the water tank, the distance between the beacons and the tank walls is only 0.6 m in y-axis-direction which increases the effect of EM reflections. The switching condition of the way-points is set to 0.35 m and the robot's target depth is set to 0.74 m.

The main challenges of this experimental setup are two-fold, the localization accuracy and the shared limited actuator bandwidth. The localization system may be heavily influenced if the robot antenna and the beacon antennas are vertically shifted or not oriented in parallel. This misalignment cannot always be fully covered by the URSM especially if the robot changes its operating depth or attitude due to large roll and pitch motions. Thus, we set the desired roll and pitch angles to zero. This setting ideally keeps the robot at the same attitude states as in the previous experiment, where the μAUV was attached to the swimming rig. The second challenge arises from the actuator bandwidth. The limited actuator bandwidth of each thruster needs to be decomposed to serve the robot's attitude controller (depth, roll, pitch) and the tracking controller (yaw orientation and thrust) at the same time.

The path driven by the HippoCampus μAUV is depicted in Figure 4.20. The depth and attitude profiles are shown in Figure 4.22, position and orientation errors are shown in Figure 4.21. Regarding the challenges of this experimental setting, the orientation error peaks at $t = 20\,\mathrm{s}$ and $45\,\mathrm{s}$ are of special interest. We observe that the position error increases after a new target way-point has been assigned. Moreover, the robot drives backwards and away from the target way-point. This may seem unexpected since the tracking controller is designed to minimize orientation errors first and pursue forward thrust second.

The reason for this is due to the limited actuator bandwidth of the μAUV's thrusters which is not fully compensated by underlying low-level control architecture. Due to the HippoCampus mixing concept, the actuator bandwidth of the robot's thrusters is shared among the control channels. In the experiment, switching to the subsequent way-point results in a jump of the yaw error and afterwards an abrupt change in the yaw control signal, Figure 4.21. Since all control signals are superimposed this results in an overshoot beyond the limit of the thrusters' bandwidth. Due to this limit, the control signals may influence each other, which can result in a backward motion, albeit no explicit backward command is given by the controller, see Figure 4.20.

Additionally, the roll angle is affected, as depicted in Figure 4.22. This impacts the localization system's accuracy, as the antenna orientation changes. Although the roll angle recovers quickly to the desired value of zero, the time series on depth, roll, and pitch in

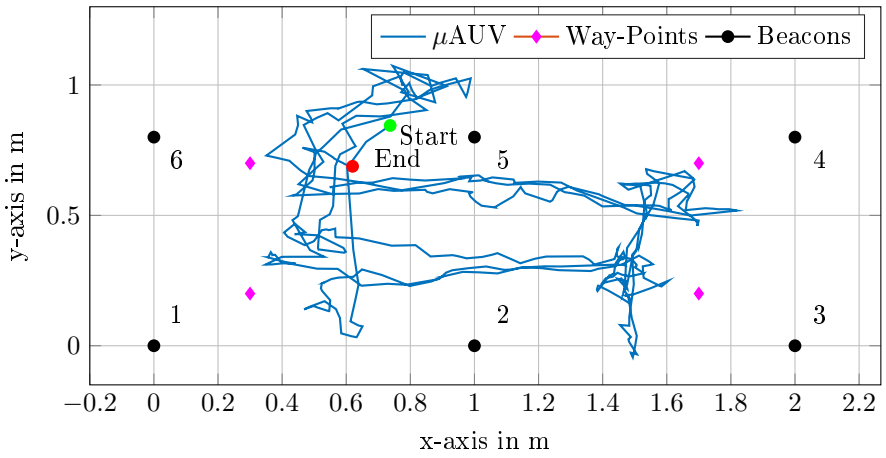


Figure 4.20: HippoCampus μAUV following a sequence of four way-points in counter-clockwise order for three rounds.

Figure 4.22 show considerable deviations from their original set points. According to [98] deviations of the antenna orientation, possibly resulting from these uncompensated robot motions, may heavily impact the measured RSS. This often results in jumpy position estimates by the EKF, as antenna orientation may not be fully covered by the URSM. This explains the zig-zag-shaped path in the region around the start and end points and indicates a locally poor localization quality.

Strong magnetic field deviations caused by the tank's steel frame were observed in the vicinity of the top-left way-point. This locally biases the magnetometer resulting in deviations of the robot's yaw estimate. As a consequence, and fostered by locally poor localization, the yaw orientation error is higher when approaching the vicinity of this way-point, compared to the remaining way-points, Figure 4.21.

Overall, the tracking performance is lower when compared to the previous experiment. Another reason for the observed performance drop is the reduced accuracy of the localization system. It is the result of disturbances induced by wall reflections and antenna orientations which are not compensated by the URSM. However, we observe good and reproducible tracking performance especially in areas between the way-points, where the robot logged path is very similar for all three rounds. In fact, the electromagnetic-based localization system and tracking controller show their strong robustness, as the μAUV is continuously able to follow the way-point sequence despite the strong disturbances.

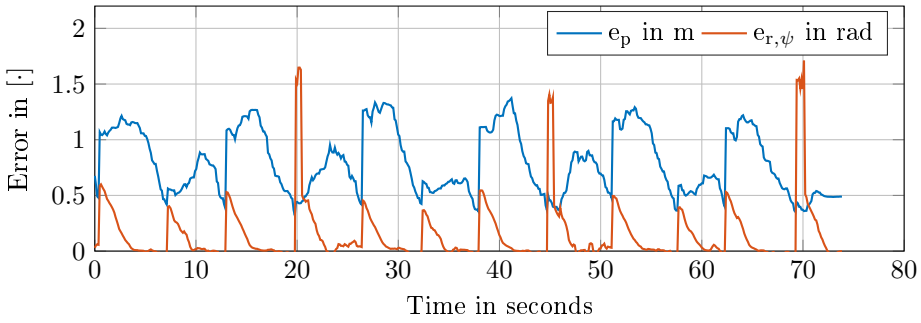


Figure 4.21: Position and yaw orientation errors.

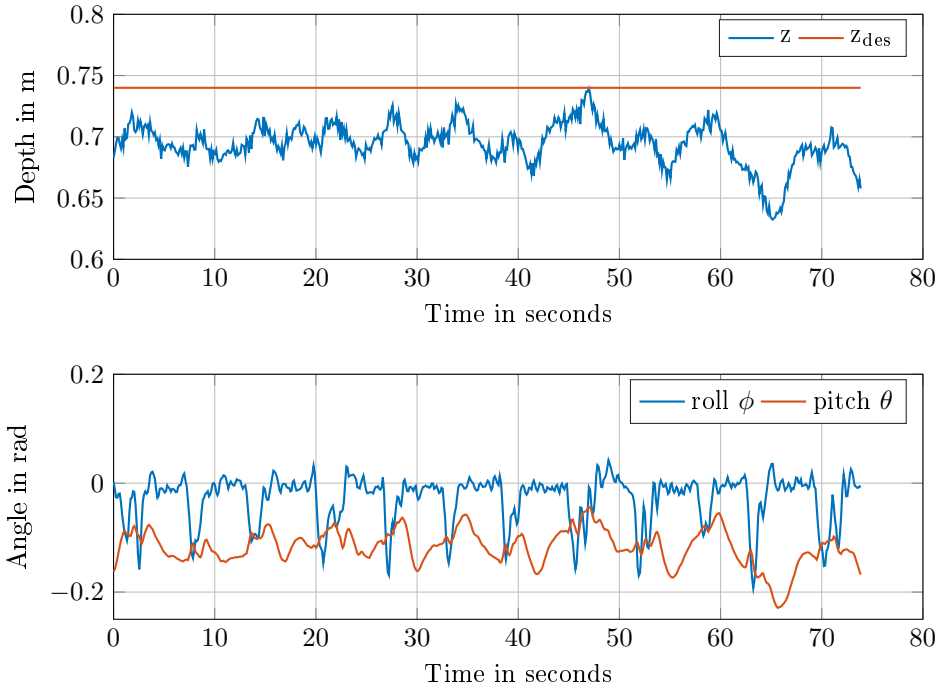


Figure 4.22: Control performance on depth z, roll ϕ, and pitch θ.

4.5 Acoustic Underwater Localization

Supplementing the previous discussions, this section targets micro robot localization in environments represented by reference *Scenario III*. As pointed out, acoustic localization is found the most promising approach to this setting due to its longer range and independence from good visibility conditions.

However, even today existing systems usually do not fit the requirements of small-scale underwater robots. Commercially available systems include the Subsonus USBL offered by Advanced Navigation [89] starting at USD 20 k for a USBL receiver and the Underwater GPS by Waterlinked AS [131] which is an SBL-system. These prices have to be seen in the context of robotic platform price where the BlueRobotic's commercial BlueROV2 platform starts at approx. USD 3,500 and research systems such as the HippoCampus lie in the range of USD 1,000. This motivates the development and testing of low-cost approaches in acoustic localization.

The following subsections, thus, study the application of the recently introduced low-cost and low-power ahoi-modem by Renner et al. [103] for the localization of micro underwater robots, see Figure 4.23. The ahoi-modem targets μAUVs and underwater wireless sensor networks by design. Moreover, in addition to most baseline systems, the ahoi-modem allows piggybacked communication between the robot and the anchor beacons. This makes the system particularly interesting for tetherless AUV platforms which do not possess a cable data link such as ROVs do. The price of a single modem is circa USD 200 for the printed circuit boards and their components. The price range of the required external hydrophone ranges from USD 100 to 400, depending on the required characteristics. In this sense, the ahoi-setup can be considered low-cost in comparison to competing acoustic approaches. In its default setup, the modem has a net data-rate of 260 B/s. The modem was successfully tested over communication ranges above 150 m.

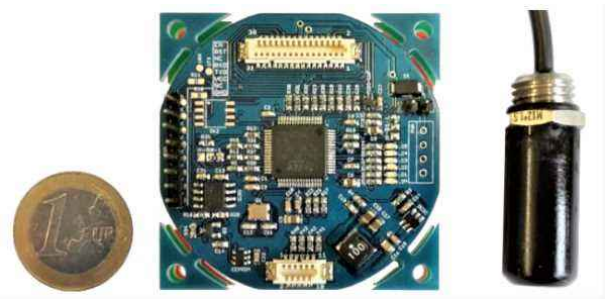


Figure 4.23: ahoi-modem with a hydrophone and a one-Euro coin for scale.

4.5.1 Methodology

The concept of the acoustic localization module follows the structure proposed in Section 4.2. It is based on two components, namely the process of two-way-ranging (TWR) and the extension to acoustic localization. Figure 4.24 depicts the integration into the HippoCampus GNC-Framework.

Acoustic Two-Way-Ranging Consider, first, the problem of estimating the distance between two anchor beacons, each equipped with an acoustic modem. TWR constitutes a simple and common approach to this task as it naturally does not require highly accurate synchronization between the modem. This first modem initiates the ranging process by emitting a *poll* signal packet. After receiving the poll the second modem replies an acknowledgment *acknowledgement* (ACK). After the ACK is received by the first modem, the distance between both modems can be estimated based on the measured TOF. Note that the accurate computation of the TOF requires knowledge of the propagation speed of the acoustic wave in the surrounding fluid volume. This propagation speed depends on parameters conductivity, salinity, and temperature which have to be measured during the deployment time. For more detailed aspects on packet-based TWR, we refer to the discussion in [104].

TWR-based Localization In order to enable TWR-based localization, the described setup of ranging between two modems has to be extended to multiple anchor beacons. In the localization setting, the mobile modem, i. e. the robot, initiates the process by broadcasting the poll signal. In this example, we assume the presence of four anchor beacons at known positions, denoted by $\mathcal{A}_{1:4}$. Note the requirement of avoiding packet collisions. Thus, the anchor nodes $\mathcal{A}_i$ respond after a predefined time delay t_{del}. This time delay is chosen based on the wave propagation speed, the maximal expected distance between the anchors, and the signal processing time onboard the modems. Starting from the initial poll packet, anchor $\mathcal{A}_1$ responds directly after the poll reception with its acknowledgment. The following anchors $\mathcal{A}_{2:4}$ sent their response after waiting for the timespan $(i-1) \cdot t_{\text{del}}$, respectively. This sequential poll-acknowledgment cycle is illustrated in Figure 4.25. While this process is not limited to four anchors, four constitutes a common anchor number to cover most volumes of interest which do not possess large obstacles. Moreover, the time span required for a full polling-acknowledgment cycle increases with additional anchors. As a consequence the system update rate decreases. After the robot successfully received the acknowledgments from the anchors a localization algorithm can estimate the robot position based on the measured TOFs, the corresponding ranges, and the known anchor

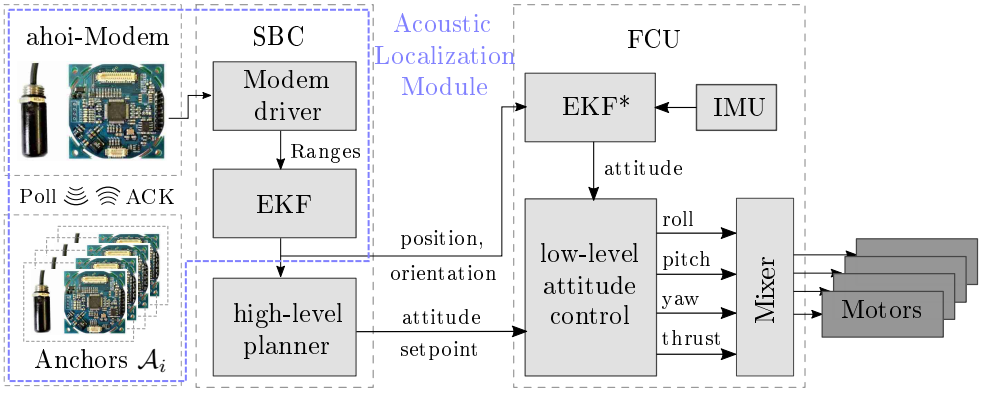


Figure 4.24: Acoustic localization module (blue box) including the onboard ahoi-modem and the remote anchor modems which are placed at known positions. The onboard modem communicates with Anchors $\mathcal{A}_i$ via *polls* and *ACK*-responses to determine the distances between each other. Depicted anchor architecture is simplified.

positions. Note that TWR systems naturally do not require synchronized clocks. However, the recent development of chip-scale atomic clocks may be an attractive extension and allow to increase update rates. In the exemplary configuration of four anchors and one mobile node the polling packet interval ranges between 3 to 5 s which corresponds to 0.33 Hz and 0.2 Hz, respectively. In comparison, commercial synchronized systems offer higher update rates, e. g. 4 Hz using Waterlinked's Underwater global positining system (GPS).

Note, an important difference of acoustic localization to the previously presented schemes is the sequential processing and fusion of the measured ranges between the robot and the anchor beacons. For example, for the visual and the electromagnetic approach the range measurements origin from the same data-sample (electromagnetic), cf. Figure 4.14, or from the same camera-frame (vision), respectively. Thus, although the number of range measurements may change, all ranges are determined at the same time stamp. This does not yield for acoustics. In fact, for the exemplary case of four anchor beacons, range measurements initiated by a poll signal arrive over a scheduled time span of 3-5 s, cf. Figure 4.25. Given this time period, it is important to incorporate their individual time stamps within the EKF innovation step to account for vehicle motions between the range

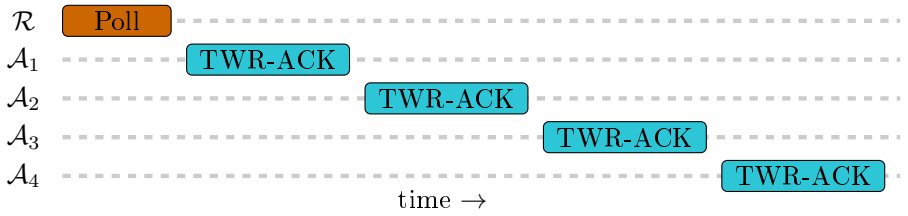


Figure 4.25: Visualization of an exemplary ranging cycle with robot $\mathcal{R}$ and anchors $\mathcal{A}_1$ to $\mathcal{A}_4$ over time. The robot initializes the cycle with a poll packet and the anchors reply (TWR-ACK) with different delays to avoid packet collisions.

measurements. This increases the importance of accurate velocity estimates which constitute the basis for accurate inter-measurement time predictions. Otherwise, considerable estimation errors are induced when mistakenly fusing measurements at false timestamps.

Packet-losses are an immanent challenge in acoustic packet-based TWR. Thus, with a considerable likelihood, an anchor misses replying to the poll with a valid acknowledgment. If this happens, no valid range to this anchor can be calculated for this poll-cycle.

In general, the spherical localization problem can also be formulated as a linear least squares triangulation problem. For triangulation, at least three range measurements to different beacons are required. However, this approach comes with two fundamental flaws: First, due to the sequential nature of the measurement, we have to buffer range measurements which results in undesirable inaccuracies as buffer measurement over a whole poll cycle. Moreover, second, this approach is less robust than EKF fusion since we require at least three valid range measurements from each poll cycle. Otherwise, the poll cycle has to be dismissed. In contrast, the EKF scheme allows fusing all individual measurements as they arrive. This allows to still update the vehicle position even if multiple range measurements temporarily fail, e. g. due to packet losses. Thus, as with the previously discussed approaches the EKF scheme is preferred for its robustness and expected accuracy. For the sake of brevity, we point out the extensions and adaptations to the detailed EKF scheme presented in Subsection 4.3.1. Besides these, the acoustic TWR scheme can directly be integrated into the filter equations. The general TWR observation function reads

$$\mu_i = h_i(\boldsymbol{p}) = \sqrt{({}^{\mathcal{W}}\boldsymbol{p} - {}^{\mathcal{W}}O_{\mathcal{A}_i})^2}, \tag{4.31}$$

where $\boldsymbol{p}$ is the robot position and ${}^{\mathcal{W}}O_{\mathcal{A}_i}$ is the known position of the i-th anchor beacon.

4.5.2 Hardware Setup

On the μAUV-side the acoustic localization module consists of the ahoi-modem, an external hydrophone, and an SBC. Additionally, the robot is equipped with a sensor suite for measuring the water's conductivity, temperature, and depth-pressure. The ahoi-modem itself consists of three stacked printed circuit boards with total dimensions of 50 mm × 50 mm × 25 mm. The modem is controlled by a specific modem driver which runs onboard the SBC. The μAUV-modem initializes the cycle while the beacons reply, as illustrated in Figure 4.25. Other transmissions schemes are of course possible but lie out of the scope of this dissertation. On the anchor beacon-side, each beacon is equipped with its own ahoi-modem and a hydrophone. Moreover, the beacons carry an SBC, a GPS receiver, and a battery power supply.

4.5.3 Performance Analysis

In the following, we study the performance of the described acoustic localization approach in an outdoor setting. The field test goal is to demonstrate that the presented approach is able to provide accurate and continuous data on the robot's position and velocity. As a reference, the vehicle's GPS track is recorded and used as a ground-truth baseline. This allows to judge the performance of the acoustic localization scheme.

The field trial was conducted in a small marina in Hamburg-Finkenwerder, Germany, in February 2020. The marina was chosen as it conforms with the requirements of reference Scenario III and allows good accessibility, see also Figure 4.26.

For the following, we are interested in the performance of the EKF-scheme in combination with acoustic TWR. Thus, we focus on the condensed version of the original results by Duecker et al. [33]. However, we refer to the original work for an extensive discussion and comparison between EKF and linear least squares localization performance.

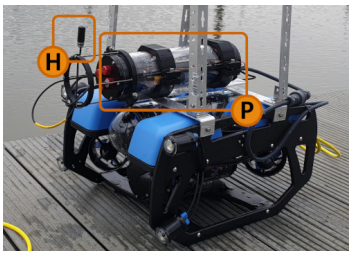
Experimental Setup

For the field experiment, four fixed reference anchor beacons, denoted by $\mathcal{A}_1$ to $\mathcal{A}_4$, are installed on the marina's jetties. The anchors' hydrophones are mounted submerged at 1.5 m depth. This setting is shown in Figure 4.26. Their positions are listed in Table 4.4.

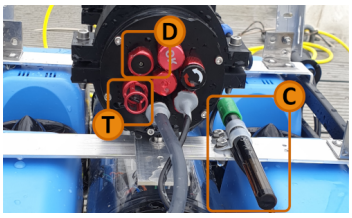
As previously discussed, the hardware setup is designed such that all necessary components fit inside the HippoCampus μAUV-platform. However, for this special examination, we deploy a fifth ahoi modem on a BlueRobotics BlueROV2 robot. The reason for this setting lies in the BlueROV's larger size which allows to mount an additional rig for the GPS receiver, as depicted in Figure 4.27. This allows to record a GPS-reference track dur-

Figure 4.26: Birds-eye view on the experimental area at Finkenwerder Marina in Hamburg. The origin of the world frame $\mathcal{W}$ coincides by definition with anchor $\mathcal{A}_4$. Table 4.4 lists the positions of all anchors.

(a) BlueROV2 (B) and attached GPS receiver (G)



(b) Hydrophone (H) and payload tube (P)



(c) Sensor suite for Temperature (T), pressure/depth (D) and salinity/conductivity (C)

Figure 4.27: BlueROV2 configuration and additional GPS receiver. This setup was used for the real-world evaluation.

Table 4.4: Anchor positions during the experimental trial. The positions are determined using differential GPS.

	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$
x	42.17 m	18.15 m	-24.25 m	0.00 m
y	26.65 m	64.18 m	37.63 m	0.00 m
z	-1.50 m	-1.50 m	-1.50 m	-1.50 m

ing the experiment. The 2 m rig top of the BlueROV2 keeps the GPS receiver well above water level while the robot can remain submerged at 1.5 m depth, see Figure 4.27a. For reference tracking, we use a Navilock NL-8001U GPS receiver which provides a position accuracy of 2.5 m [90].

During the experiment, the ahoi-modem is deployed in an additional tube on top of the BlueROV2, as shown in Figure 4.27b. The modem's hydrophone is placed in front of the tube to achieve omnidirectional sensitivity. During the evaluation, the recorded distance measurements are with respect to the hydrophone position. In addition to the modem, the tube houses a Raspberry Pi Zero, a power supply, and conductivity, temperature, and depth (CTD) sensors. Depth measurements $z_{\mathcal{R}}$ are provided by the robot's onboard pressure sensor.

As discussed the speed of sound depends on depth, temperature, and salinity [123]. Therefore, an integrated sensor suite is used which enables a self-estimation of the speed of sound within the operating area. Additionally, we deploy a professional CTD-probe to obtain a reference value for the speed of sound. From the measured water temperature of 5.0 °C and 0.47 ppt salinity at a depth of 1.3 m we obtain a speed of sound was 1,427 m/s.

During the evaluation, the BlueROV2 transmits a poll packet ($T_{\text{poll}} = 820\,\text{ms}$) every 3.5 s. This poll packet carries 16 B random payload, which we block to simulate transmitted information from the μAUV to the anchors. This way, the anchors can be used as a relay station to provide a live-data link to a surface network. The ACK has a length of $T_{\text{ack}} = 451\,\text{ms}$ and the response delays were adjusted to $T_{\text{del}} \in \{0\,\text{s}, 0.7\,\text{s}, 1.4\,\text{s}, 2.1\,\text{s}\}$ corresponding to anchor $\mathcal{A}_1$ to $\mathcal{A}_4$, respectively.

Acoustic Self-Localization

In the following, we evaluate the localization performance of our EKF self-localization algorithm in comparison to the GPS-reference path which was recorded during the trials.

Therefore, we drive the robot with a speed of ca. 0.5 m/s two laps along the path at a desired depth of 1.5 m. The period from 0 s to 400 s corresponds to the first lap, while the second lap corresponds to the span from 400 s to 900 s. The total path length of both laps is about 281 m according to the recorded GPS reference track. Figure 4.28 shows the recorded track.

Packet Reception Rate In order to obtain a high reliability of the localization scheme a high rate of successfully received acknowledgments is desirable. Ideally, each poll packet is replied with an acknowledgment from each anchor beacon. During the evaluation (900 s long), the robot's modem transmitted 255 poll packets. Out of these the robot received 249 ACKs from anchor $\mathcal{A}_1$ (97.7 % received ACKs), 250 ACKs from $\mathcal{A}_2$ (98.0 %), 247 ACKs from $\mathcal{A}_3$ (96.7 %), and 249 ACKs from $\mathcal{A}_4$ (97.7 %). When compared with earlier field studies [51, 103] this constitutes a noticeable high successful reception rate.

Localization Accuracy The estimated path driven by the robot together with the corresponding GPS reference track are depicted in Figure 4.28. Both paths match accurately for the two recorded laps.

Additionally, the robots' position profile over time is depicted in Figure 4.29. Notice the jumps in the GPS position after ca. 180 s, which corresponds to the path segment close to anchor $\mathcal{A}_3$.

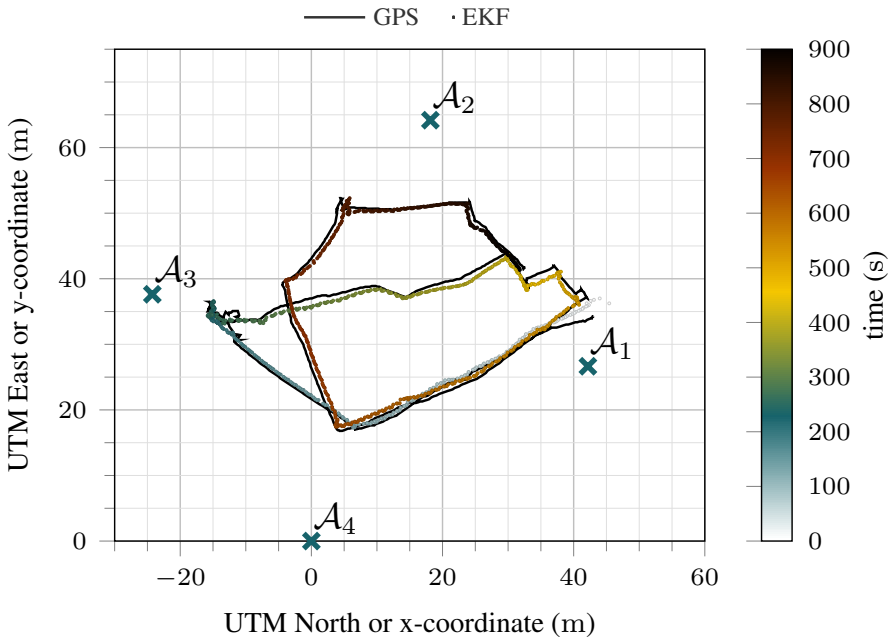


Figure 4.28: Recorded EKF and GPS tracks over two laps. The robots starts close to anchor $\mathcal{A}_1$ and proceeds for two rounds in clock-wise direction.

Note the anchor-agent distances are measured sequentially. This naturally leads to a localization error, since the agent proceeds along its path during the measurement sequence.

In contrast, our EKF scheme fuses the individual anchor distance measurements immediately as they are received by the robot's modem. This is advantageous as no measurements representing *old* robot positions from previous time instances are fused into the position estimate.

Due to its motions model, the EKF is able to estimate the agent velocity. This is important since distance measurements are only available at low update rates. The corresponding velocity-time profiles are depicted in Figure 4.30. As a reference, we plot the GPS-velocity reconstructed from the GPS-position data points. We can observe that the EKF velocity estimate matches the GPS velocity. In fact, the underlying EKF-motion model provides a smooth estimate on the velocity's $x-$ and $y-$component and, thus, increases the robustness of the position estimate. Note that accurate and smooth velocity information is a key requirement when predicting position estimates between the anchor distance measurements. This is highly relevant as it allows to provide the robot controller with the vehicle state at a sufficiently high update rate.

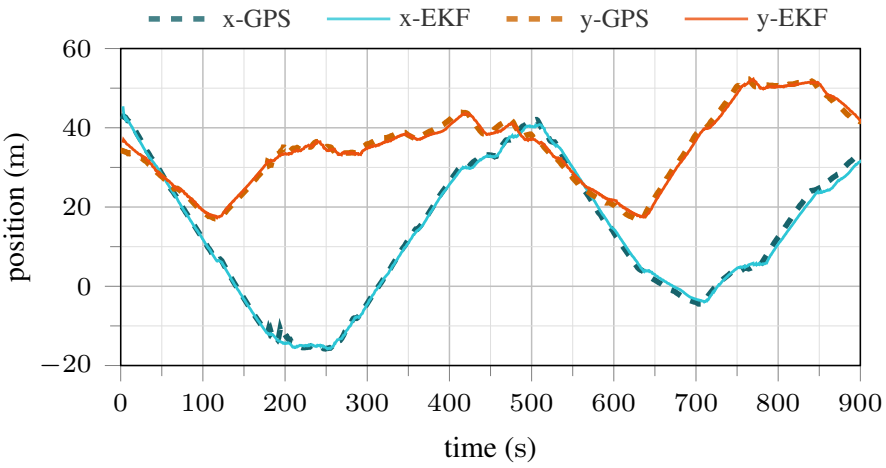


Figure 4.29: EKF and GPS position profiles of the robot during the field experiment.

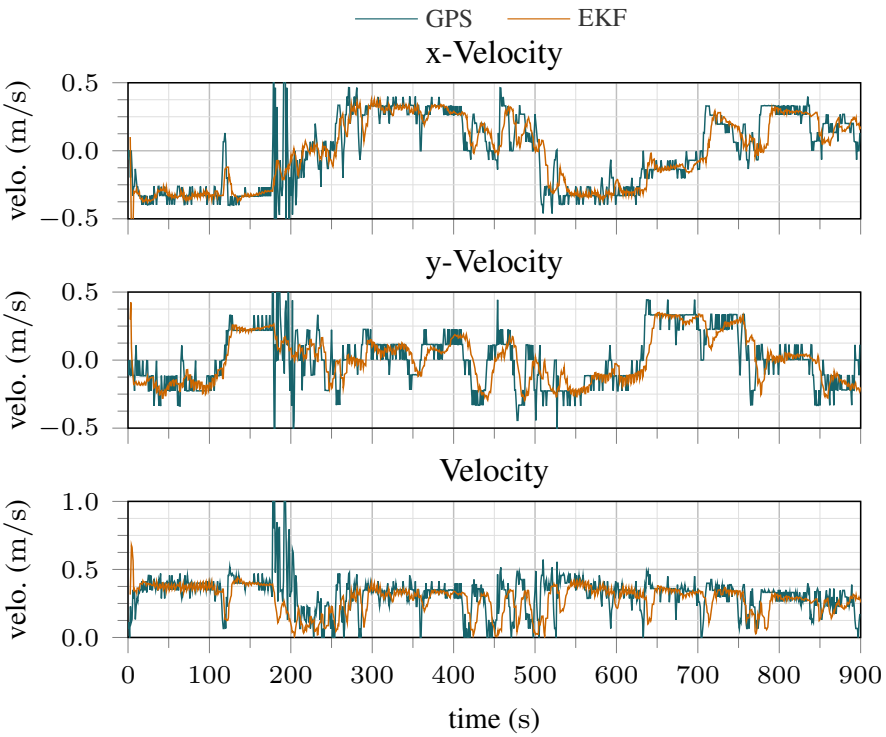


Figure 4.30: Velocity profiles of the underwater robot over two laps based on EKF estimates and GPS-position.

4.6 Summary and Discussion

This chapter proposed three approaches to the problem of μAUV self-localization in confined underwater scenarios. First, three promising ranging technologies were identified based on a detailed problem analysis and the defined reference scenarios. It was found that the representative references scenarios need to be addressed by individual localization technologies. A modular approach was chosen which allows individually addressing the individual requirements. Namely, two approaches for short-range localization one based on vision and fiducial markers, the other approach based on the attenuation of electromagnetic carrier signals. They were supplemented by an acoustic two-way ranging localization scheme to address also medium-scale scenarios. Each approach was implemented onboard the μAUV hardware platform and tested in a physical experimental setting. The experimental trials included outdoor testing at the Hamburg-Finkenwerder Marina, in the Hamburg-Wilhelmsburg public swimming pool, and inside a research basin at the TU Hamburg's Institute of Mechanics and Ocean Engineering. Overall, the three approaches demonstrated very promising performance in these real-world settings. Specifically, the proposed concepts fulfill the posed key criteria of being able to run on very restricted computational resources. Furthermore, they provide accurate and robust information of the μAUV location. The findings of this chapter have been published in a series of papers [10, 23, 24, 25, 30, 33, 133].

Chapter 5

Autonomous Field Exploration with μAUV Fleets

In this chapter, we propose a novel online informative path planning (IPP) and control framework which enables autonomous environmental field exploration with a multi-robot fleet. For this, we combine deep reinforcement learning, specifically double deep Q-learning (DDQN), with a Gaussian Markov random field (GMRF) belief representation. The result is a high-level planning module that is suitable for field exploration with mobile robots.

Consider a robot exploring its unknown surrounding by taking observations of its environment. In this setting, IPP formulates the problem of finding paths that are expected to maximize the robot's information gain about its environment with respect to some a-priori chosen metric. This fundamentally differs from standard path planning problems which usually focus on finding a feasible and fast paths from point A to point B. In this sense, IPP constitutes the information-theoretic extension to regular path planning methods.

Planning algorithms can be generally split into *online* and *offline* algorithms. As their category name indicates, offline methods compute and optimize paths a-priori, i. e. before the robot is deployed. Obviously, these methods do not allow to actively adapt the planning process to new information gathered during the mission. On the upside, a-priori computation allows the use of strong computational resources. Hence, prior knowledge of complex physical processes can be integrated into the planning process as no real-time capability is required. Online methods come with a clearly different profile. They allow live adaptation of their planning and, thus, their actions to newly gathered information by the robot's environment sensors. This is also referred to as the *perception-action-loop* which is illustrated in Figure 5.1. Algorithms for online planning are thus usually designed to be lightweight because the majority or even all related computation processes have to run embedded onboard the robot. The robot's limited computational resources, thus, pose an upper limit on planning horizons and the robot's ability to optimize its decisions.

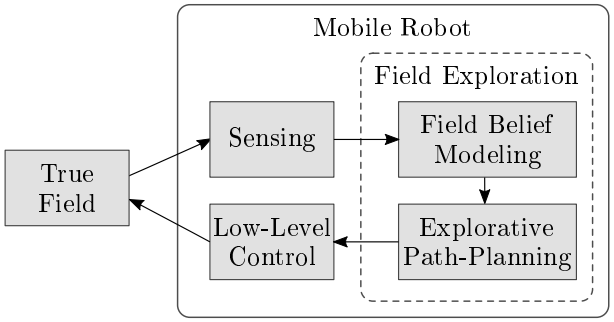


Figure 5.1: Four steps of the exploration methodology, referred to as perception action loop.

The remainder of this chapter is structured as follows. In Section 5.1, we formally state the problem of IPP for underwater field exploration with μAUV fleets. Section 5.2 proposes a full framework to solve the IPP problem. In detail, we propose a combination of a deep reinforcement learning agent (Subsection 5.2.6) and an environmental field belief based on Gaussian Markov random fields (Section 5.3). The proposed field exploration framework is evaluated in a case study in Section 5.4. The chapter concludes with a brief summary and a discussion of the findings in Section 5.5.

5.1 Problem Formulation

Consider the scenario of exploring a spatio-temporal environmental field $\mathcal{F}_{\text{real}}$ within a confined fluid volume. For this purpose, a fleet of n_{AUV} μAUVs is deployed and moves through this environment to efficiently gather information on the field of interest, e. g. a radiation or flow field. The mission goal is to obtain spatially sufficiently well-distributed samples of the present environmental field such that a reliable field estimate $\mathcal{F}$ can be inferred. For this purpose, an exploratory control law is required that computes robot control actions such that the error on the desired field estimate is minimized.

Let $\boldsymbol{x}^i$ denote the state of the i-th μAUV capturing its position, velocity, orientation, and angular velocity with respect to a world-fixed reference frame. The temporal evolution of the i-th μAUV can be described by

$$\dot{\boldsymbol{x}}^i = \boldsymbol{f}^i\left(\boldsymbol{x}^i, \boldsymbol{u}^i\right) + \boldsymbol{G}\left(\boldsymbol{x}^i\right), \tag{5.1}$$

where $\boldsymbol{f}^i$ denotes the μAUV dynamics analog to our analysis in Section 3.2, $\boldsymbol{u}^i$ is the corresponding control input. Moreover, $\boldsymbol{G}$ represents the fluid flow at the robot location affecting its motion. For the following, let the superscript i denote the i-th μAUV agent and $\bar{i}$ denote all other agents but the i-th.

Along their mission the μAUVs continuously take noise corrupted measurements of the surrounding environmental field $\mathcal{F}_{\text{real}}$. The measuring process of taking the measurement $\boldsymbol{z}^i$ of the real field $\mathcal{F}_{\text{real}}$ at $\boldsymbol{x}^i$ is described by the function $\boldsymbol{h}^i$

$$\boldsymbol{z}^i = \boldsymbol{h}^i \left(\mathcal{F}_{\text{real}} \left(\boldsymbol{x}^i \right), \boldsymbol{w} \right), \tag{5.2}$$

which incorporates the sensor characteristics. We assume that the measurements are subject to uncorrelated zero-mean Gaussian measurement noise $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_z)$ with covariance matrix $\boldsymbol{\Sigma}_z$. The measurement function $\boldsymbol{h}^i$ depends on the individual μAUV agent and its implemented sensor. However, for the following, we assume that all vehicles follow the same dynamic model and carry an identical sensor suite.

Note, the underlying goal is to minimize the error between the inferred field estimate $\mathcal{F}$ and the true field $\mathcal{F}_{\text{real}}$. From an information-theoretic point-of-view, this is equivalent to minimizing the uncertainty of the field estimate. Intuitively, obtained field measurements reduce the field uncertainty at the collection location and its vicinity. Hence, it is desirable to incorporate and exploit this property within the exploratory control loop. This favors probabilistic field representations, for instance, Gaussian random fields (GRFs), because these belief models naturally provide direct access to the belief's second-order moments, i. e. the field covariance. This moment can be used as an uncertainty measure along with the estimated mean values of the spatial field. From the perspective of exploratory control, it is thus necessary to have access to a fictional online representation of the field to which we refer to as *field belief*. This belief is maintained throughout the mission and updated by collected field measurements.

Especially when considering multi-robot exploration tasks, the algorithmic implementation of the field belief has to be lightweight to allow fast inference cycles. This is crucial for decentralized implementations where each agent maintains its own field belief since these run embedded onboard the robot agent. This embedded design is often a key requirement for underwater exploration missions since limited communication bandwidth and high latencies render centralized concepts infeasible. Furthermore, the field belief representation should remain computationally tractable over time. This is an important criterion since the computational burden of many belief representations, including the widely used GP-based representation, scale badly as more and more field measurements are collected.

However, it is worth noting, that the field estimate characteristics directly depend on the individual mission goal. As a consequence, field estimate models may come with various sets of requirements. For instance, applications exist which do not require a lightweight implementation because it is not running on-board the mobile agent. An example are complex high-fidelity physics-based fluid models, e. g. Navier-Stokes models, which are

fitted during post-processing based on previously gathered data samples. This fitting is usually done offboard using high-performance computer clusters. However, this requires that information rich data has been gathered beforehand. For this application of fitting physics-based models, the field's first stochastic moment, i. e. the field's mean values are paramount. Application-wise one would choose a light weight field belief model for the exploratory control and a complex high-fidelity field estimate model for the post processing step. However, for this work, we assume the field estimate and field belief to be identical and keep $\mathcal{F}$ for their notation.

Considering the task of IPP, it is intuitive that the gained information from measurements taken at locations with high field uncertainty is larger and more favorable than that of measurements at regions with low uncertainty. Thus, to avoid being prone to local regions of low uncertainty a non-myopic planning and control strategy is beneficial which takes upcoming information gains into account. Moreover, a lightweight implementation is required to run the exploratory control loop onboard.

5.2 DRL for Multi-Robot Informative Path Planning

Handcrafting control policies for multi-robot systems in complex and unstructured environments is at best hard and tedious but often just intractable. Following our prior discussion in Subsection 2.4.3, the application of DRL methods to the IPP problem has recently gained attention but is still in its infancy. The approach proposed in this section aims to exploit the recent advances in deep learning research to learn a policy that is optimized for the IPP problem. We propose a novel IPP methodology based on deep reinforcement learning and a Gaussian field belief to address the earlier introduced multi-robot field exploration problem.

Various recent studies [118, 127, 128] address the problem of path planning for monitoring and coverage control by considering the full target environment in form of a global map which serves as an input to their planning algorithms. Regarding our target application, we consider this approach suboptimal since the learned controller cannot be transferred to environments with different side lengths without repeating the training process. This is particularly problematic in confined environments which usually have remarkably varying side dimensions. Furthermore, these scenarios contain obstacles. For this reason and given the limited onboard computing resources of micro underwater robots we suggest to focus the IPP on a scalable local vicinity of the robot.

The concept of our IPP approach follows four key ideas:

1. Each robot agent runs its own decentralized instance of the informative path planner and the environmental field belief. Taken measurements and their collecting positions are shared among the robots.

2. The robot agent's field belief is interpreted as an imaginary image that is used as an information-rich container. The image's color-channels encode various spatial quantities, for instance, the field uncertainty, the physical field value (e. g. temperature), obstacles, and boundaries. These field images are then fed into a convolutional neural network to extract regions of interest, e. g. areas of high uncertainty.

3. Field images are rendered from the full field belief but capture only the local vicinity around the agent for two reasons: (i) the trained convolutional neural network (CNN) does not depend on the geometrical shape of the fluid environment and the field belief, (ii) the input shape of the CNN is kept small to reduce the number of trainable parameters and the computational burden for the agent.

4. The agent's main success metric is to maximize the reduction in field uncertainty. However, additional aspects can be added, see Subsection 5.2.5.

The rough structure of the proposed planning methodology is inspired by Viseras et al. [127]. However, our proposed approach remarkably differs on the functional level. For instance, we suggest a *local* observation renderer in combination with a *stochastic* field belief representation. This allows formulating the agent reward in terms of uncertainty reduction rather than error minimization which is found an intuitive and general metric.

5.2.1 Informative Path Planning-Architecture

This subsection explains the architecture and briefly introduces the main components of the IPP methodology. The corresponding block diagram is depicted in Figure 5.2.

Each robot agent i runs an identical instance of IPP algorithm. It interacts with the world via its sensors, the communication interface, and the computed control signal. The *world* represents the real world, i. e. the fluid volume, the robot agents, and their physics. For training and simulation purposes, these are replaced by a simulated field and dynamic models of the robot platform. In the following, we exemplary describe our algorithm for a single robot denoted by superscript i. Quantities and information related to the remaining fleet robots are denoted by the superscript $\bar{i}$.

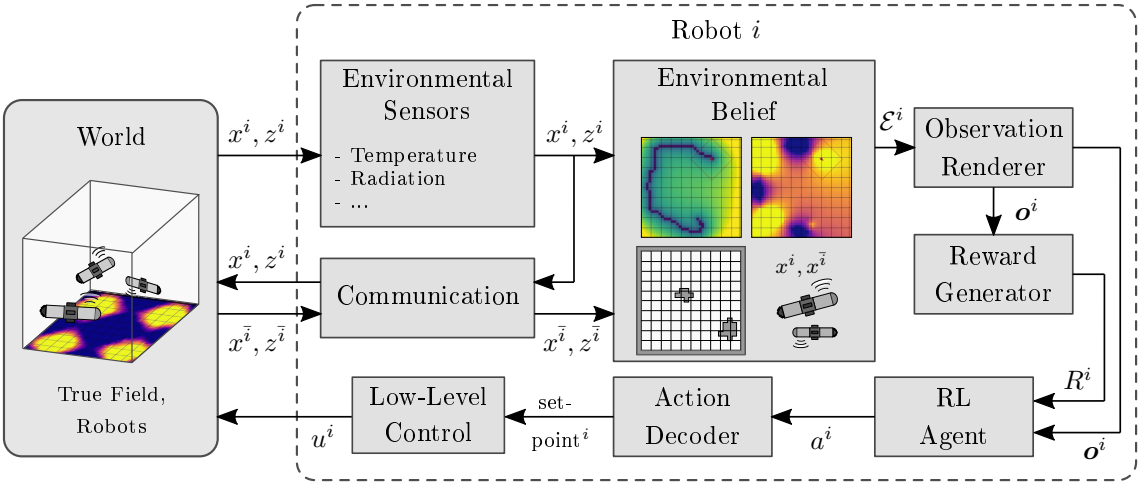


Figure 5.2: Architecture of the decentralized informative path planning framework.

The environmental sensors gather field measurements z^i at the robot's position $\boldsymbol{x}^i$ according to Equation (5.2). The obtained field measurement $(\boldsymbol{x}^i, z^i)$ is shared with the other agents via the *communication* module. Furthermore, it is fused together with the measurements from the other agents $(\boldsymbol{x}^{\bar{i}}, z^{\bar{i}})$ into the robot's *environmental belief*. The environmental belief module hosts all information on the world environment available to robot i. Most importantly, the module maintains and updates the field belief (Subsection 5.3.2). The *observation renderer* provides the agent with its local observation $\boldsymbol{o}^i$ of the field belief (Subsection 5.2.4). The *reward generator* computes the reward signal R^i. This provides the agent with positive or negative feedback on the outcome of its last chosen action in form of a real-value number (Subsection 5.2.5). The observation $\boldsymbol{o}^i$ and the reward R^i are then fed into the *RL-Agent* module. The reinforcement learning (RL) algorithm computes the agent's next action a^i. This process is described in detail in Subsection 5.2.6. The obtained action a^i is processed by an action decoder which maps the action to control setpoints for the robot's low-level controller which computes the control signal $\boldsymbol{u}^i$, as discussed in Section 3.3.

Within our approach, the robot agents solely exchange gathered measurements and the corresponding locations through their communication module. Thus, a low bandwidth communication link between the robots is sufficient. For the scope of this work, we assume this link to be ideal. As a consequence, the robots share their own instance of the same field belief.

5.2.2 Decentralized Partially Observable Markov Decision Process-Formulation

In their simplest form, planning problems can be formulated as an Markov decision process (MDP). The concept of MDPs in their standard form holds only for single-agent configurations. Thus, for multi-agent settings, the concept is generalized to Markov games.

A common assumption in planning problems is that all robot agents have perfect knowledge about the environment state $\boldsymbol{s} \in \mathcal{S}$. This is indeed not realistic, as the robots have imperfect sensors. Moreover, the complexity of the full state $\boldsymbol{s}$ is too large to be handled by the robot's onboard computational resources. In this case, the MDP becomes a partially observable Markov decision process (POMDP).

Given the considered multi-robot scenario, we face a set of multiple agents with individual observations, i. e. each agent observes its local vicinity. However, due to limited communication bandwidth and computational resources, we cannot assume that all agents have access to all observations. For this setting, Oliehok [92] suggests the framework of a decentralized partially observable Markov decision process (Dec-POMDP). Following this, we formulate the presented IPP problem as a Dec-POMDP.

Let the Dec-POMDP be defined by the tuple $(\mathcal{S}, \{\mathcal{A}^i\}, P, \{R^i\}, \Omega, \{\mathcal{O}^i\}, \gamma)$. In this Dec-POMDP-formulation, the environment is defined by the state $\boldsymbol{s} \in \mathcal{S}$. The agent i can chose an action $a^i \in \mathcal{A}$ from the action space $\mathcal{A}$ which causes a state transition from $\boldsymbol{s}_t$ to the subsequent state $\boldsymbol{s}_{t+1}$. This transition is described by the state transition probability function $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R} \in [0, 1]$. Furthermore, $R : \mathcal{O} \times \mathcal{A} \times \mathcal{O} \to \mathbb{R}$ denotes the reward function which maps the current observation, the applied action, and the next observation to a real-valued reward. The observation function is denoted by $\Omega : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathcal{O}$ where $\mathcal{O}$ is the set of possible observations perceived by the agent. Thereby, $\boldsymbol{o}^i \in \mathcal{O}$ denotes the agent i's partial observation of the environment. We elaborate this observation step in more detail in Subsection 5.2.4. Finally, the discount factor $\gamma \in (0, 1]$ is a tuning parameter that weighs the importance of long- and short-term rewards.

In detail, the state of the IPP problem is described by

$$\mathcal{S} = \underbrace{\mathbb{R}^{M \times M \times n_{\text{env}}}}_{\text{Environment Map}} \times \underbrace{\mathbb{R}^{M \times M \times n_{\text{bel}}}}_{\text{Exploration Map}} \times \underbrace{\mathbb{R}^{3 \times n_{\text{AUV}}}}_{\text{Agent State}}, \tag{5.3}$$

in which the elements $\boldsymbol{s}_t \in \mathcal{S}$ read

$$\boldsymbol{s}_t = (\boldsymbol{M}, \boldsymbol{F}_t, \boldsymbol{p}_t)\,, \tag{5.4}$$

where $\boldsymbol{M}$ is the static environment map consisting of n_{env} layers for field boundaries and obstacles. Moreover, $\boldsymbol{F}_t$ describes the field belief $\mathcal{F}$ in tensor form consisting of n_{env} layers for the spatial field uncertainty and the estimated spatial field value, e. g. the local values of the field. The field belief can also consist of further layers representing additional quantities of interest. Furthermore, $\boldsymbol{p}_t$ represents the state of the present n_{AUV} agents at time instance t. The i-th agent's action $a_t^i \in \mathcal{A}$ at time t is selected from the set of possible actions

$$\mathcal{A} = \{\text{left}, \text{straight}, \text{right}\}. \tag{5.5}$$

Note that the selected action at each time step is not applied directly to the robot agent's actuators. Instead, the selected action is sent to the *action decoder* module which determines the corresponding set-point for the robot's low-level controller which then computes the corresponding actuator commands. Note that for the readers convenience, we introduce the commonly used short notation with $\boldsymbol{s} = \boldsymbol{s}_t$, $\boldsymbol{s}' = \boldsymbol{s}_{t+1}$, $\boldsymbol{o} = \boldsymbol{o}_t$, $\boldsymbol{o}' = \boldsymbol{o}_{t+1}$, and $a = a_t$, respectively.

5.2.3 Environmental Belief Representation

As discussed, each robot agent i runs its own belief representation $\mathcal{E}^i$ of the surrounding environment. In this sense, the belief representation constitutes a managed container for all information available to robot i. Considering our scenario of field exploration, this container includes map representations of the spatial field mean and its uncertainty. This is illustrated in Figure 5.3. For a detailed discussion of these field map representations, we refer to Section 5.3. Furthermore, the environmental belief representation manages available information on other robot agents as well as maps on known obstacles.

In order to avoid the available information becoming outdated, the belief representation conducts continuous inference if new information becomes available. The information is updated by either the robot's own sensors or by external data sources via a communication module, for example information shared by other robots, e. g. measurements and robot states updates.

5.2.4 Observation Renderer

The observation renderer creates the observation $\boldsymbol{o}^i$ of the i-th agent. Note that since we cannot assume that the environment is fully observable the agent has only access to its environmental belief representaion $\mathcal{E}^i$. Depending on the dimensions and resolution of

the environmental belief, processing an observation that represents the complete environmental belief may become intractable for the exploring μAUV agent. Furthermore, the limited computational resources motivate to reduce the complexity of the agent's observation. This aligns with the expectation that information associated with positions far away from the agent likely has a small direct impact on the agent's next decision. However, limiting the observation horizon ℓ_o only to the agent's direct vicinity, e. g. areas which are reachable within single or a few timesteps, likely leads to a very greedy exploration behavior. As a consequence, the agent may get stuck in a local information minimum as areas of high information do not lie within the observation horizon.

In this work, we propose to represent the agent's observation in form of an image-like representation of the environmental belief, see Figure 5.3. The observation renderer maps the information provided by the environmental field belief into an image observation in a multi-layer format. The data rendered in this multi-layer format can then be efficiently fed into the convolutional encoders of the agent's neural network architecture. Here, analog to color channels in an image, each observation channel contains a class of information. These classes capture the belief's individual quantities, for instance, field uncertainty, field values, obstacles, and field boundaries. The structure of the observation renderer is illustrated in Figure 5.3 for a three-layer example of the field belief including field uncertainty, field value, and obstacles. In order to reduce the complexity and size of the agent's observation, we only consider information within a pre-selected observation horizon radius ℓ_o when rendering the observation from the full environmental belief. This extraction is performed by means of a mapping function $\boldsymbol{\Phi}$ reading

$$\boldsymbol{o}^i = \boldsymbol{\Phi}^i \left(\mathcal{E}^i, \boldsymbol{p}^i \right). \tag{5.6}$$

It samples the desired information from the full belief based on the agent's current position and pose $\boldsymbol{p}^i$ and the parameters defining the sampling density ξ and the observation horizon ℓ_o. Various geometrical shapes of the observation are indeed possible, however, the idea of an image-like representation motivates to sample the environmental belief on a regular lattice with side-lengths $2\ell_o$ and spacing ξ. This is illustrated in Figure 5.3 for each information channel by means of red boxes surrounding the agent (red arrow).

Note that in most cases when a CNN architecture is combined with a map-like representation two aspects have to be taken into account. First, the CNN input layer needs to consider the dimension of the map representation. Second, we require some way of incorporating the agent's position with respect to the belief map. A common approach for matching the CNN input to the map is to simply fix the CNN input layer to the global map dimensions. However, this limits the ability to use the same DRL agent for

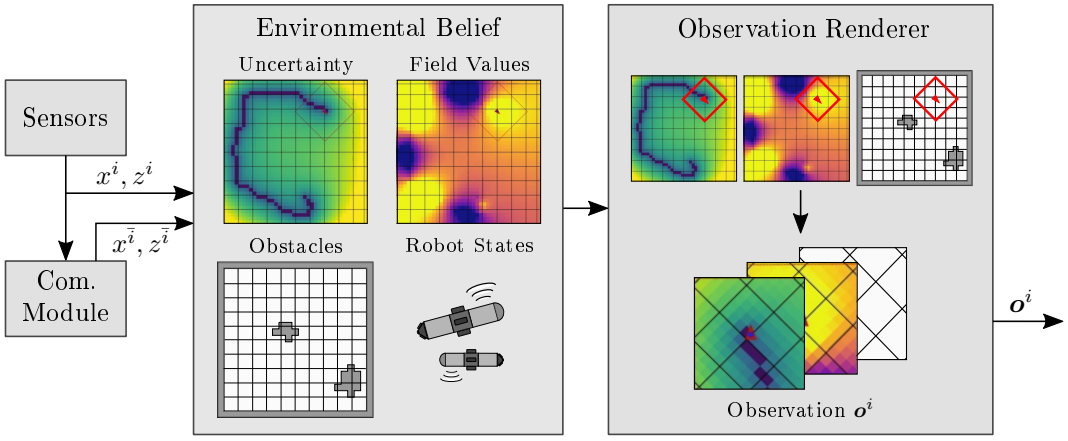


Figure 5.3: Block Diagram of the Environmental Belief and the Observation Renderer. The environmental belief module manages the environmental field maps and updates them based on the collected field information. The observation renderer samples information from the environmental belief on a regular lattice (*red box*) in the vicinity of the agent (*red arrow*). The resulting output is a multi-layer local representation of the full environmental belief.

different map shapes. In the worst case, a simple change to the map shape would require to restart training the DRL agent. In order to represent the agent's position in such a setting, various publications suggest a *one-hot* representation of the agent, for example [118]. In simple words, these approaches add an extra map-layer which consists solely of zero values and a single *one*-value representing the agent position inside the map. In contrast to this, *local sampling* of the observation avoids both problems elegantly. The information captured by the observation is by definition given with respect to the agent position, e. g. the agent position lies in the middle of the lattice as depicted in Figure 5.3. Furthermore, we can avoid the extra CNN-input layer which otherwise would only track the position.

5.2.5 Reward Generator

Shaping the reward function is a critical and nontrivial step in the design process of learning frameworks for robotic agents. The reward function allows affecting the desired agent behavior on the one hand by penalizing undesirable outcomes of agent actions. On the other hand desirable outcomes can be rewarded and, thus, the likelihood of their future selection be increased. Regarding the present IPP problem, we are mainly interested in favoring actions that effectively reduce the uncertainty of the field belief. Depending on the specific mission, focusing the exploration on areas of interest, e. g. areas of high concentration may be desirable and should, can be incorporated within the agent's reward signal. At the same time, actions that result in collisions with obstacles and other robot

agents can be penalized. Within our field exploration framework, we compute the agent's reward based on the available information stored in the agent's environmental belief. We refer to Subsection 5.2.1 and Figure 5.2 for an intuitive illustration of the data streams. In the following, we first present a generalized formulation of the reward function R which can be easily adapted to the individual mission requirements. Afterwards, we briefly present an exemplary realization of this general form.

In general, the goal of IPP is to minimize the error between the estimated field and the real-world field. Recent works such as [127] use a direct error computation for which they take the normalized root mean squared error (RMSE) between both quantities. However, in this work we follow a more probabilistic approach. Specifically, we aim to favor the maximization of field uncertainty reduction. By this, the agent is encouraged to explore regions of high uncertainty. This is found an appealing and promising approach since it allows to incorporate the robot sensors' measurement noise.

In order to quantify uncertainty and information gain, various evaluation metrics are available [67]. For this work, we chose the predictive variance sum conditioned on the collected and fused field measurements to compute $r_{\text{uncertainty}}$. Predictive variance sum constitutes a widely used metric in spatial statistics and is also referred to as Bayesian A-optimality. The predictive variance for given field reference points $\boldsymbol{x}_{\text{ref}}$ reads in general form

$$\text{Var}\left[\mathcal{F}\left(\boldsymbol{x}_{\text{ref}}\right)|\boldsymbol{z}\right] = \sum_{\boldsymbol{x}_{\text{ref}}} \mathbb{E}\left[\left(\mathcal{F}\left(\boldsymbol{x}_{\text{ref}}\right) - \mathbb{E}\left[\mathcal{F}\left(\boldsymbol{x}_{\text{ref}}|\boldsymbol{z}\right)\right]\right)^2 |\boldsymbol{z}\right], \tag{5.7}$$

where $\mathbb{E}[\cdot]$ is the expectation operator and $\boldsymbol{x}_{\text{ref}}$ denotes the local observation area covered by the observation $\boldsymbol{o}^i$.

Consider a generalized reward function $R\left(\boldsymbol{o}_t, \boldsymbol{a}_t, \boldsymbol{o}_{t+1}\right)$. The reward is computed based on the current observation $\boldsymbol{o}_t$, the chosen action $\boldsymbol{a}_t$ and the subsequent observation $\boldsymbol{o}_{t+1}$ reading

$$R\left(\boldsymbol{o}_t, a_t, \boldsymbol{o}_{t+1}\right) = c_{\text{uncertainty}}\, r_{\text{uncertainty}} + c_{\text{mean}}\, r_{\text{mean}} + \ldots + \sum_i c_i\, r_i, \tag{5.8}$$

where r_i-terms denote the individual rewards and c_i their corresponding weights. The reward terms representing field *uncertainty* and field *mean* are chosen as an example. The reward function can consider a wide variety of aspects. However, in the context of IPP, the most prominent rewards are briefly introduced in the following. Note that many more reward designs are possible and their selection depends on the specific application scenario.

$r_{\text{uncertainty}}$ In IPP we aim to maximize the reduction of the field uncertainty in form of the predictive variances comparing $\boldsymbol{o}$ and $\boldsymbol{o}'$.

r_{mean} Depending on the mission goal, it could be of interest to find areas of high or low field values, e. g. sources of pollution. Increasing the weight on this term biases the exploration focus towards these areas. However, it has to be ensured that the overall field is sufficiently covered such that areas with potential peak values are not left unexplored.

r_{control} This term incorporates control costs with a negative reward and allows to handle missions with energy budgets, e. g. due to limited battery capacity.

r_{obstacle} In the presence of obstacles, this (negative) reward keeps the agent at a safe distance to avoid collisions. This leads to a more conservative exploration strategy.

5.2.6 Reinforcement Learning-Agent

In general, the goal of reinforcement learning (RL) is to learn behavioral policies to optimize the outcome of sequential decision making problems. Given its environment, an RL agent chooses an action according to its policy. It receives a reward signal depending on the positive/negative outcome of the action. The agent's policy is improved by optimizing a cumulative future reward signal. The reader is referred to the textbook by Sutton and Barto [113] for a throughout introduction to the concept of RL.

Within the proposed exploration framework, the *RL-agent* module corresponds to the robot's brain. It is responsible for choosing the best action a_t at each time instance t given a policy π. In the following, we present a DRL algorithm to learn a behavioral policy π which enables the robot to efficiently explore the surrounding environmental field by taking spatially distributed measurements.

Double Deep Q-Learning for Field Exploration

Q-Learning is one of the most popular methodologies within the suite of RL algorithms and can be traced back to the prominent work by Watkins and Dayan [132]. It aims to find a state-action-value function, referred to as Q-value function $Q : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. The Q-function describes the expected reward of a state-action pair referred to as Q-value given a policy π. It reads

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\left[R_t | s_t = s, a_t = a\right], \tag{5.9}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator. Given a state $\boldsymbol{s}_t$, a higher Q-value for an action a_t indicates that this action is expected to result in a more desirable long-term result. Consequently, the goal is to find the optimal Q-function. The Q-values are learned iteratively by updating the current Q-values with gained experiences measured by the achieved reward. The update rule reads

$$Q(s,a) := Q(s,a) + \alpha \left(R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a) \right), \tag{5.10}$$

where α denotes the so-called *learning rate*. A traditional and common approach to Q-learning is the use of Q-tables which store the Q-values of state-action pairs. However, this renders intractable for most real-world problems where the number of state-action pairs increases and leads to huge tables with dimension $\mathcal{S} \times \mathcal{A}$. An alternative approach is to approximate the Q-function by a neural network, referred to as a deep Q-network (DQN). This network is parameterized by the vector $\boldsymbol{\theta}$. Training the DQN aims to minimize the expected temporal difference error described by the loss function

$$L(\boldsymbol{\theta}) = \mathbb{E}_{\pi} \left[\left(Q_{\theta}(s,a) - Y(s,a,s') \right)^2 \right], \tag{5.11}$$

where $Y(\cdot)$ is the target value reading

$$Y(s,a,s') = R(s,a) + \gamma \max_{a'} Q_{\theta}(s',a'). \tag{5.12}$$

Methods using DQNs are considerably more data-efficient than Q-table approaches as their generalization allows to provide Q-values also for previously not-visited state-action pairs. However, their training may easily become unstable and diverge. In this sense, the work by Mnih et al. [83] can be seen as the breakthrough of deep Q-learning. Mnih et al. propose an approach that considerably stabilizes the training phase by introducing a separate *target network* parameterized by $\bar{\boldsymbol{\theta}}$ and the use of *experience replay*. Both, the target network's model and the online network model are the same besides they have their own parameter vector $\bar{\boldsymbol{\theta}}$. Regarding the target network, the vector $\boldsymbol{\theta}_{\text{target}}$ is updated only periodically with the online parameters $\boldsymbol{\theta}_{\text{online}}$ and kept fixed for all other time steps. Introducing the target network allows to provide a more stable estimate of the next maximum Q-value. Additionally, Mnih et al. propose experience replay. Here, a replay memory $\mathcal{D}$ stores experience tuples $(\boldsymbol{s}, a, r, \boldsymbol{s}')$ gathered over time while the agent is interacting with its environment. The agent then uniformly samples mini-batches from its memory $\mathcal{D}$ to update its network. The resulting loss function adapted from Equation (5.11) reads

$$L(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{s},a,r,\boldsymbol{s}') \sim \mathcal{D}} \left[\left(Q_{\theta}(s,a) - Y(s,a,s') \right)^2 \right]. \tag{5.13}$$

Following the findings presented by Mnih et al. [83], both approaches drastically improve the RL algorithm's performance. This is demonstrated by the authors on various games of the Atari game suite.

Q-learning and its neural network-based pendant deep Q-learning tend to overestimate their actions' Q-values. This effect has been first studied by Thrun and Schwarz [121] for traditional Q-learning and can lead to undesirable oscillations in the training process. Double Q-learning constitutes a solution to this problem and has been first proposed by van Hasselt [124] for tabular Q-learning. It was later generalized to the double deep Q-learning (DDQN) algorithm in [125]. The key idea is to compute the target value by

$$Y^{\mathrm{DDQN}}\left(s, a, s'\right) = R(s, a) + \gamma Q_{\bar{\theta}}\left(s', \underset{a'}{\operatorname{argmax}}\, Q_{\theta}\left(s', a'\right)\right), \tag{5.14}$$

leading to the loss function

$$L^{\mathrm{DDQN}}\left(\boldsymbol{\theta}\right) = \mathbb{E}_{\pi}\left[\left(Q_{\theta}\left(s, a\right) - Y^{\mathrm{DDQN}}\left(s, a, s'\right)\right)^2\right]. \tag{5.15}$$

Notice that in Equation (5.14) we selected the best action based on $\boldsymbol{\theta}$ but estimate the value of this action using our DQN parameterized by $\bar{\boldsymbol{\theta}}$. This constitutes an important difference to Equation (5.12) and considerably improves the training process.

Network Architecture

We propose a network architecture consisting of convolutional and fully-connected layers to map the observation input $\boldsymbol{o}$ to the action a. The network is illustrated in Figure 5.4. The observation $\boldsymbol{o}$ is created by the observation renderer which we introduced in Subsection 5.2.4. It consists of the multi-channel map representation of the agent's local physical environment, e. g. boundaries and obstacles, as well as the maintained field belief. Note that due to our local mapping approach, we avoid the effort of representing our agent's position explicitly within the belief map. This is done for instance in [118] using a one-hot representation and the introduction of additional map channels.

The observation input is fed into three convolutional layers which encode the spatial connections between the rendered map cells. Note that we need to pad the convolutional layers such that their output shape remains the same as their input shape. Additionally, early trials showed that is it beneficial to add *maxpool*-layers between the convolutional layers to decrease the dimension of the layer's input shapes along the network data flow. The widely used rectified linear unit (ReLU) is used as an activation function between all

layers. The output of the last convolutional layer is flattened such that it can be fed into the fully-connected layer.

The last fully-connected layer of our network has dimension $|\mathcal{A}|$ and provides the Q-values for all actions based on the input observation $\boldsymbol{o}$. Action selection is done via an ϵ-greedy policy reading

$$\pi(\boldsymbol{o}) = \begin{cases} \text{argmax}_{a \in \mathcal{A}} Q_\theta(\boldsymbol{o}, a), & \text{with probability } 1-\epsilon, \\ \text{random action } a \in \mathcal{A}, & \text{with probability } \epsilon. \end{cases} \tag{5.16}$$

This means that the action with the highest Q-value is selected with a chance of $1-\epsilon$. A random action $a \in \mathcal{A}$ is chosen with a probability of ϵ. During the training phase, ϵ is decreased following a decrement schedule. Starting from a large share of randomly taken actions in the beginning of the training, this reduces the share of random actions (*exploration*-phase) over time. At the same time, the trust in actions chosen by means of the learned Q-value function increases (*exploitation*). This results in exploratory behavior at the beginning of the training, then shifting the focus to exploiting the learned Q-value function.

Action selection using a *soft-max* policy constitutes an alternative to the ϵ-greedy policy, Equation (5.16). The soft-max balances exploration and exploitation dynamically based on the relative difference of the Q-values. Note that this makes the soft-max policy independent from the number of training iteration steps. The soft-max policy reads

$$\pi(a_i|\boldsymbol{o}) = \frac{e^{Q_\theta(\boldsymbol{o},a_i)/\beta}}{\sum_{a_j \in \mathcal{A}} e^{Q_\theta(\boldsymbol{o},a_j)/\beta}}, \tag{5.17}$$

where β is a tuning parameter. Increasing β favors exploratory behavior while $\beta \to 0$ results in a purely greedy policy, i. e. Equation (5.16) with $\epsilon = 0$ [118].

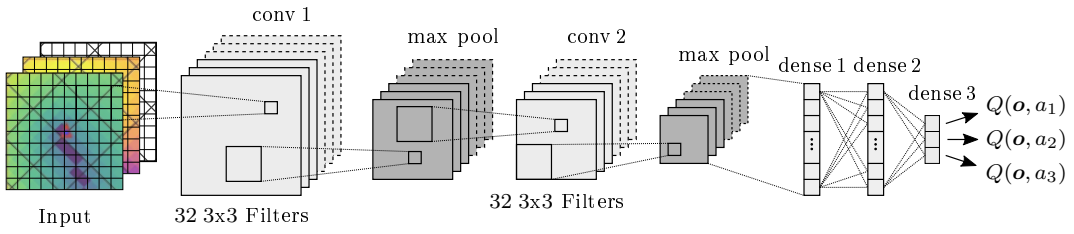


Figure 5.4: Sketch of the Neural Network architecture consisting of two convolutional layers which are trailed by max-pooling layer. Their output is flattened in fed into two fully connected hidden dense layers with consist of 256 neurons. The last layer provides outputs in form of Q-values for three actions. All layers possess ReLU activation functions. An ϵ-greedy policy is used for action selection.

Training Procedure

In the following, we briefly describe the training procedure for our proposed deep reinforcement learning IPP framework.

As a first step, the replay buffer $\mathcal{D}^i$ is initialized as it gathers experience tuples over multiple episodes. At the start of each episode, we reset the environment state $\boldsymbol{s}$. This includes randomly sampling the robot agent's initial state as well as the generation of the obstacle map and the field environment.

The episode continues until either the episode length ℓ_{episode} is reached or the episode is terminated, e. g. the mission is completed or the robot has crashed into an obstacle. Since the environment is only partially known to the agent, the observation renderer provides a local observation $\boldsymbol{o}^i$ of the robot surroundings. This observation is fed into the DQN to obtain Q-values for the available actions. The action is then selected according to the policy π and sent to the robot's low-level controller. Next, the robot takes a field measurement according to Equation (5.2) to update its field belief. We provide more details of the inference process of the field belief in Section 5.3. Reward R^i is computed by the reward generation module. Subsequently, the new experience tuple $(o^i, a^i, R^i, o'^{,i})$ is stored in the replay buffer $\mathcal{D}^i$.

A minibatch of size n_{batch} is sampled from replay buffer $\mathcal{D}^i$ to train the DQN's parameter vector $\boldsymbol{\theta}$ using the Adam optimizer [65]. Target network is updated every $n_{\text{target\_update}}$ iteration steps. Then, the network parameter vector $\bar{\boldsymbol{\theta}}$ is updated by $\boldsymbol{\theta}$. Finally, a new episode begins until the maximum number of training episodes N_{max} is reached.

5.3 Probabilistic Field Belief Modeling

In this section, we present a probabilistic field belief representation based on the Gaussian Markov random field (GMRF) formalism in [26]. This representation allows information-based exploration control with mobile robots. Note that an important criterion for this application is the constant computational cost over time.

In order to enable the incorporation of the belief models into the previously introduced IPP framework, we revise the belief algorithm presented by Kreuzer and Solowjow [68]. The original algorithm comes with the shortcoming that it does not fulfill the requirement of constant computational cost over time which is an important requirement for implementations on micro robots.

The GMRF-based belief algorithm was originally proposed in [136] and enables efficient estimation of stationary spatial processes on a discrete grid of Gaussian random variables.

Defining the field representation on a lattice raises the question of how to choose the grid discretization based on the fundamental trade-off between the accuracy of the field representation versus the available computational power. The latter aspect is of particular importance in the field of underwater robotics, as off-board computation of the field representation is not feasible due to the very limited communication bandwidth. Thus, the discretization step size has to be selected depending on the actual application scenario. Here, prior knowledge on the spatial scale of the physical process is a helpful and valid assumption. For instance, if the user aims to explore small-scale processes in a local environment, e. g. an industrial tank, a dense grid is likely to be a better choice than a coarse grid that captures global physical phenomena with an acceptable computational burden. Moreover, shape function approximation can be used to interpolate the field belief between the grid points. This allows to efficiently monitor large-scale fields where the main interest lies in the exploration of global phenomena rather than local small-scale processes. For more details on shape function approximation in GMRFs, we refer the reader to the recent works [26, 68].

5.3.1 Gaussian Markov Random Field Regression

Gaussian Markov random fields define a conditional autoregressive (CAR) process. A process is a CAR(j) process if the expectation of a process value is fully defined through the next j adjacent graph vertices. The Markov assumption enables the direct construction of a sparse precision matrix. Given a labeled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V} = \{1, ..., n\}$ and edges $\mathcal{E}$, a probabilistic graphical model $\boldsymbol{\eta}$ defines a GMRF, if the edges $\mathcal{E}$ are chosen such that there is no edge between node i and j, if $\eta_i \perp \eta_j \mid \boldsymbol{\eta}_{-ij}$, in which $-ij$ denotes the nodes adjacent to i and j, respectively [105]. The pairwise conditional independence properties of $\boldsymbol{x}$ on $\mathcal{G}$ are inherent in the subdiagonal entries of the precision matrix $\boldsymbol{\Lambda}$. We refer the reader to [105] for an in-depth discussion of GMRFs.

In order to construct the GMRF, the continuously indexed spatial field $\mathcal{F}^* \subset \mathbb{R}^d$ is discretized into a labeled undirected spatial graph with n^* vertex positions $\mathcal{S}^* = \boldsymbol{x}_1, ..., \boldsymbol{x}_{n*}$. Here, $\boldsymbol{x}_i$ denotes the i-th field vertex position. Note that in this section the scalars x or y denote the spatial position coordinates of a two-dimensional spatial position vector $\boldsymbol{x}$, while a bold $\boldsymbol{y}$ represents an environmental field observation vector. The set of field locations $\mathcal{S}^*$ is extended to $\mathcal{S}$ with vertex positions $\mathcal{S} = \boldsymbol{x}_1, ..., \boldsymbol{x}_n$, as depicted in Figure 5.5, to compensate boundary effects occurring due to the GMRF approximation. Then on $\mathcal{S}$, a GMRF $\boldsymbol{\eta}$ is constructed using the SPDE approach proposed by [75].

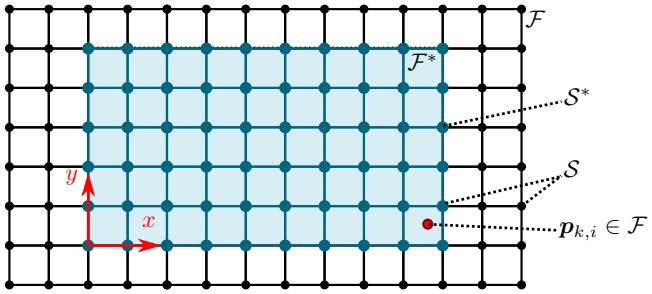


Figure 5.5: The agent's position $\boldsymbol{p}_{k,i}$ at the discrete timestep k lies in a spatial field $\mathcal{F}^*$ with coordinate values x and y. The field $\mathcal{F}^*$ can be extended to $\mathcal{F}$ and is then discretized into a regular grid $\mathcal{S}$ to enable the construction of a GMRF and to compensate for boundary effects.

Let $\boldsymbol{\eta} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$ be a GP on $\mathbb{R}^2$ defined by the Matérn covariance function defined as

$$k_{\text{Matérn}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \Big(\kappa \left\| \boldsymbol{x} - \boldsymbol{x}' \right\| \Big)^{\nu} \mathrm{K}_{\nu} \Big(\kappa \left\| \boldsymbol{x} - \boldsymbol{x}' \right\| \Big), \tag{5.18}$$

in which $\|\cdot\|$ denotes the Euclidean distance in $\mathbb{R}^d$ and K_{ν} the modified Bessel function of the second kind. The GMRF $\boldsymbol{\eta} \sim \mathcal{N}(0, \boldsymbol{\Lambda}^{-1})$ defined on a regular two-dimensional lattice approximates a Matérn GP for $\nu = 0$ if the Gaussian full conditionals read

$$\mathbb{E}\left(\eta | \boldsymbol{\eta}_{-ij}, \boldsymbol{\theta}\right) = \frac{1}{a}\left(\eta_{i-1,j} + \eta_{i+1,j} + \eta_{i,j-1} + \eta_{i,j+1}\right) = \frac{1}{a} \begin{pmatrix} \circ & \bullet & \circ \\ \bullet & \circ & \bullet \\ \circ & \bullet & \circ \end{pmatrix}, \tag{5.19}$$

$$\mathrm{Pre}\left(\eta | \boldsymbol{\eta}_{-ij}, \boldsymbol{\theta}\right) = a\tau. \tag{5.20}$$

For the case of $\nu = 1$, the Gaussian full conditionals read

$$\mathbb{E}\left(\eta | \boldsymbol{\eta}_{-ij}, \boldsymbol{\theta}\right) = \frac{1}{4 + a^2} \begin{pmatrix} & \circ & \circ & \circ & \circ & \\ \circ & \circ & \bullet & \circ & \circ \\ \circ & \bullet & \circ & \bullet & \circ \\ \circ & \circ & \bullet & \circ & \circ \\ & \circ & \circ & \circ & \circ & \end{pmatrix} 2a - \begin{pmatrix} & \circ & \circ & \circ & \circ & \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \\ & \circ & \circ & \circ & \circ & \end{pmatrix} 2 - \begin{pmatrix} & \circ & \circ & \bullet & \circ & \\ \circ & \circ & \circ & \circ & \circ \\ \bullet & \circ & \circ & \circ & \bullet \\ \circ & \circ & \circ & \circ & \circ \\ & \circ & \circ & \bullet & \circ & \end{pmatrix} 1, \tag{5.21}$$

$$\mathrm{Pre}\left(\eta | \boldsymbol{\eta}_{-ij}, \boldsymbol{\theta}\right) = \left(4 + a^2\right)\tau, \tag{5.22}$$

with $a = \kappa^2 + 4$ and $\boldsymbol{\theta} = [\tau, \kappa]^{\top} \in \mathbb{R}^2_{>0}$ being hyperparameters of the model. The additional hyperparameter τ adjusts the GMRF's signal variance independent of κ. The proof of Equations (5.20) and (5.22) for the general case of irregular grids is stated by Lindgren et al. [75]. Figure 5.6

illustrates the correspondence between the spatial lattice locations and the values in each column of $\mathbf{\Lambda}$ using the previously presented construction scheme.

When designing the GMRF precision matrix, the full conditionals for the border vertices of the GMRF grid affect the estimation result considerably. In order to actively cope with this effect, boundary conditions have to be defined. Three commonly used boundary conditions are the Dirichlet, Neumann, and torus boundary conditions and discussed in [105].

5.3.2 Sequential GMRF Regression

In this subsection, the GMRF regression algorithm proposed by Xu et al. [136] is extended to enable spatial process estimation with continuous observations. For the sake of brevity, we present a concise discussion of this extension in the following and refer for the detailed discussion to the article by Duecker et al. [30].

Let the values of the field be represented by the latent variable $z_i = z(\boldsymbol{s}_i) \in \mathbb{R}$. The latent variables are expressed using a global linear model, such that

$$z_i = \mu(\boldsymbol{s}_i, \boldsymbol{\beta}) + \eta_i \quad \forall\, 1 \leq i \leq n, \tag{5.23}$$

$$\mu\left(\boldsymbol{s}_i, \boldsymbol{\beta}\right) = \boldsymbol{m}^\top \boldsymbol{\beta}. \tag{5.24}$$

Hereby, $\boldsymbol{m} = [m_1(\boldsymbol{s}_i), ..., m_p(\boldsymbol{s}_i)]^\top \in \mathbb{R}^p$ denotes the regression function vector and the vector $\boldsymbol{\beta} = [\beta_1, ..., \beta_p]^\top$ contains the unknown regression coefficients. The field belief on the lattice is denoted as $\boldsymbol{z} = [z_1, ..., z_n]^\top$. The small-scale correlations of the field are modeled through the zero-mean GMRF $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}_{\boldsymbol{\eta}|\boldsymbol{\theta}}^{-1})$. We initialize the GMRF precision matrix $\mathbf{\Lambda}_{\boldsymbol{\eta}|\boldsymbol{\theta}}$ with the full conditionals as defined in Equation (5.20) and Equation (5.22). A zero-mean Gaussian prior is assumed on the regression coefficients $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{T}^{-1})$ to estimate the regression coefficients as a function of $\boldsymbol{z}$ and $\boldsymbol{\theta}$, where $\boldsymbol{T}^{-1}$ is initialized as a diagonal matrix with large diagonal elements. The probability distribution of the full latent field $\bar{\boldsymbol{z}} = \left[\boldsymbol{z}^\top, \boldsymbol{\beta}^\top\right]^\top \in \mathbb{R}^{n+p}$ reads

$$p(\bar{\boldsymbol{z}}, \boldsymbol{\theta}) = p(\boldsymbol{z}|\boldsymbol{\beta}, \boldsymbol{\theta})\, p(\boldsymbol{\beta}),$$

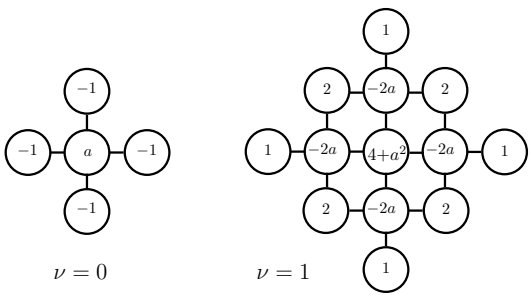


Figure 5.6: Non-zero elements of a column of the precision matrix $\mathbf{\Lambda}$ associated with one random variable and its neighbor vertices on a regular two-dimensional GMRF lattice.

$$\propto \exp\Big(-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{m}\boldsymbol{\beta})^\top \boldsymbol{\Lambda}_{\boldsymbol{\eta}|\boldsymbol{\theta}}(\boldsymbol{z}-\boldsymbol{m}\boldsymbol{\beta}) - \frac{1}{2}\boldsymbol{\beta}^\top \boldsymbol{T}\boldsymbol{\beta}\Big),$$
$$= \exp\big(-\frac{1}{2}\bar{\boldsymbol{z}}^\top \boldsymbol{\Lambda}_{\bar{\boldsymbol{z}}|\boldsymbol{\theta}}\bar{\boldsymbol{z}}\big), \tag{5.25}$$

with precision matrix $\boldsymbol{\Lambda}_{\bar{\boldsymbol{z}}|\boldsymbol{\theta}} \in \mathbb{R}^{(n+p)\times(n+p)}$ being defined as

$$\boldsymbol{\Lambda}_{\bar{\boldsymbol{z}}|\boldsymbol{\theta}} = \begin{bmatrix} \boldsymbol{\Lambda}_{\boldsymbol{\eta}|\boldsymbol{\theta}} & -\boldsymbol{\Lambda}_{\boldsymbol{\eta}|\boldsymbol{\theta}}\boldsymbol{m} \\ -\boldsymbol{m}^\top \boldsymbol{\Lambda}_{\boldsymbol{\eta}|\boldsymbol{\theta}} & \boldsymbol{m}^\top \boldsymbol{\Lambda}_{\boldsymbol{\eta}|\boldsymbol{\theta}}\boldsymbol{m} + \boldsymbol{T} \end{bmatrix}. \tag{5.26}$$

By exploiting the GMRF's *canonical form*, only the current measurements $\boldsymbol{y}_k$ are required to sequentially update the conditional precision matrix $\boldsymbol{\Lambda}_{\bar{\boldsymbol{z}}|\boldsymbol{\theta},\boldsymbol{y}_{1:k}} := \boldsymbol{\Lambda}_{k|\boldsymbol{\theta}}$ and the *canonical mean* $\boldsymbol{b}_k = \boldsymbol{\Lambda}_{k|\boldsymbol{\theta}}\boldsymbol{\mu}_{k|\boldsymbol{\theta}}$. A sequential updating algorithm is obtained by inserting the canonical mean definition into the formula for conditioning of a multivariate Gaussian distribution, such that

$$p(\bar{\boldsymbol{z}}|\boldsymbol{\theta},\boldsymbol{y}_{1:k}) = \mathcal{N}(\boldsymbol{\Lambda}_{k|\boldsymbol{\theta}}^{-1}\boldsymbol{b}_k, \boldsymbol{\Lambda}_{k|\boldsymbol{\theta}}^{-1}), \tag{5.27}$$

$$\boldsymbol{\Lambda}_{k|\boldsymbol{\theta}} = \boldsymbol{\Lambda}_{\bar{\boldsymbol{z}}|\boldsymbol{\theta}} + \frac{1}{\sigma_y^2}\boldsymbol{\Phi}_{1:k}^\top \boldsymbol{\Phi}_{1:k} = \boldsymbol{\Lambda}_{k-1|\boldsymbol{\theta}} + \frac{1}{\sigma_y^2}\boldsymbol{\Phi}_k^\top \boldsymbol{\Phi}_k, \tag{5.28}$$

$$\boldsymbol{b}_k = \frac{1}{\sigma_y^2}\boldsymbol{\Lambda}_{k|\boldsymbol{\theta}}\boldsymbol{\Lambda}_{k|\boldsymbol{\theta}}^{-1}\boldsymbol{\Phi}_{1:k}^\top \boldsymbol{y}_{1:k} = \boldsymbol{b}_{k-1} + \frac{1}{\sigma_y^2}\boldsymbol{\Phi}_k^\top \boldsymbol{y}_k, \tag{5.29}$$

with initial conditions $\boldsymbol{\Lambda}_{0|\boldsymbol{\theta}} = \boldsymbol{\Lambda}_{\bar{\boldsymbol{z}}|\boldsymbol{\theta}}$ and $\boldsymbol{b}_0 = \boldsymbol{0}$. The final sequential GMRF regression algorithm is summarized in Algorithm 1. To obtain the variance $\mathrm{diag}(\boldsymbol{\Lambda}_{k|\boldsymbol{\theta}}^{-1})$ of the full latent field, without calculating the inverse of the precision matrix, the Sherman–Morrison formula is used, Line 13. The complete derivation is outlined in Appendix A. For the sake of clarity, we omit the notation for the conditioning of the GMRF matrices on the hyperparameters $\boldsymbol{\theta}$. It is worth mentioning that adding the product $\boldsymbol{\Phi}_k^\top \boldsymbol{\Phi}_k$ to $\boldsymbol{\Lambda}_{0|\boldsymbol{\theta}}$ does not significantly increase the density of the initial precision matrix. Thus, the algorithm has a computational complexity of $O(1)$ over time.

Algorithm 1 Sequential GMRF Regression

Require: Hyperparameter vector $\boldsymbol{\theta}$, Extended field lattice $\mathcal{S}$, Regression function $\boldsymbol{m}$, Measurement variance σ_y^2, $\boldsymbol{b}_{0,0} = \boldsymbol{0}$, $\boldsymbol{\Lambda}_{0,0} := \boldsymbol{\Lambda}_{\bar{\boldsymbol{z}}}$

1: compute $\mathrm{diag}(\boldsymbol{\Sigma}_0) = \mathrm{diag}(\boldsymbol{\Lambda}_{\bar{\boldsymbol{z}}}^{-1})$
2: **for** $k \in \mathbb{Z}_{>0}$ **do**
3: **for** $1 \leq j \leq N$ **do**
4: get j-th agent location $\boldsymbol{x}_{k,j}$ and measurement $\boldsymbol{y}_{k,j}$
5: compute $\boldsymbol{\Phi}_{k,j}(\boldsymbol{x}_{k,j}, \mathcal{S})$
6: $\boldsymbol{b}_{k-1,j} = \boldsymbol{b}_{k-1,j-1} + \frac{1}{\sigma_y^2}\boldsymbol{\Phi}_{k,j}^\top \boldsymbol{y}_k$
7: $\boldsymbol{\Lambda}_{k-1,j} = \boldsymbol{\Lambda}_{k-1} + \sum_{l=1}^{N} \frac{1}{\sigma_y^2}\boldsymbol{\Phi}_{k,l}^\top \boldsymbol{\Phi}_{k,l}$
8: $\boldsymbol{h}_{k,j} = \boldsymbol{\Lambda}_{k-1,j}^{-1}\boldsymbol{\Phi}_{k,j}^\top$
9: **end for**
10: $\boldsymbol{b}_{k,0} = \boldsymbol{b}_{k-1,N}$
11: $\boldsymbol{\Lambda}_{k,0} = \boldsymbol{\Lambda}_{k-1,N}$
12: $\boldsymbol{\mu}_{\boldsymbol{k}} = \boldsymbol{\Lambda}_{k,0}^{-1}\boldsymbol{b}_{k,0}$
13: $\mathrm{diag}(\boldsymbol{\Sigma}_k) = \mathrm{diag}(\boldsymbol{\Sigma}_{k-1}) - \sum_{l=1}^{N} \frac{\boldsymbol{h}_{k,l} \circ \boldsymbol{h}_{k,l}}{\sigma_y^2 + \boldsymbol{\Phi}_{k,l}\boldsymbol{h}_{k,l}}$
14: **end for**

Computational Complexity

In the following, we provide an overview of the computational complexity of our proposed field belief algorithm in comparison to other methods. The upper bounds of the belief algorithm's computational complexity are summarized in Table 5.1.

The original empirical GMRF algorithm, as proposed by Kreuzer and Solowjow [68], already shows a drastic improvement in terms of processor and memory requirements compared to vanilla GP regression. Nonetheless, the empirical GMRF algorithm still suffers from a linearly increasing computational cost over the number of time steps k, as we assume the ongoing gathering of field measurements.

The sequential GMRF regression algorithm utilizes the canonical form of the GP, which in combination with a predefined GMRF precision matrix enables a sequential update of the belief. Therefore, the sequential Bayesian GMRF algorithm has a constant computational time with upper bound $O(Nn^{3/2})$ for the two-dimensional scenario. In general, the computational time of the GMRF increases with the number of dimensions as the bandwidth of the precision matrix increases [105].

Table 5.1: Computational complexity of the developed belief algorithms for two and three field dimensions. Number of agents N, number of discrete time steps k, and the number of field grid values n.

Belief Algorithm	**2d**	**3d**
GP Regression	$O((Nk)^3)$	$O((Nk)^3)$
Empirical GMRF Regression	$O(n^{3/2}) + O(Nk)$	$O(n^2) + O(Nk)$
Sequential GMRF Regression	$O(Nn^{3/2})$	$O(Nn^2)$

5.4 Case Studies on Multi-Robot Field Exploration

In this section, we study the functional feasibility and the performance of our proposed exploration framework. As a reference scenario, we aim to explore and monitor a scalar radiation field in a confined environment. This setting constitutes a common mission scenario expected to be addressed by μAUVs in the future. The designed scenario mimics deployment scenes such as storage pond monitoring in nuclear facilities and disaster sides such as the Fukushima-Daiichi power plant.

Our feasibility demonstration is followed by performance analysis considering various configurations of our algorithm. For this purpose, we benchmark our DRL-GMRF approach against a random walk (RW) explorer which is a common baseline for IPP solvers [127]. Finally, we test the developed software architecture on the HippoCampus μAUV to discuss the computational feasibility.

5.4.1 Study Setup

For the following studies, we consider the exploration of a scalar radiation field which consists of four radiation sources as illustrated in Figure 5.7. Given their continuous emissions, the field can be assumed to be time-invariant. For the sake of simplicity, we normalize the radiation strength to a maximum level of four. Depending on the study, one or multiple μAUV agents are deployed to explore an unknown environmental field. The agent i is described by its state $\boldsymbol{\zeta} = \left[x^i\, y^i\, \psi^i\right]^\top$ and its dynamics are governed by a self-propelled particle model with orientation and unit mass,

$$\begin{bmatrix} \dot{x}^i \\ \dot{y}^i \\ \dot{\psi}^i \end{bmatrix} = \begin{bmatrix} v\cos\left(\psi^i\right) \\ v\sin\left(\psi^i\right) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ u^i \end{bmatrix}. \tag{5.30}$$

Throughout our studies, we assume a constant vehicle velocity v. Regarding the software-side, all algorithms used in this case study are implemented in PyTorch and Python. Regarding the neural network architecture of our DRL-GMRF agent, we use for all studies the DDQN as illustrated in Figure 5.4 with hyperparameters listed in Table 5.4. For our study, we consider a confined fluid volume with planar dimensions of $10\,\mathrm{m} \times 10\,\mathrm{m}$. The GMRF belief representation is defined by its hyperparameters listed in Table 5.2. Note, previous works [68, 110] use a considerable coarser discretization for their GMRF belief, namely 12-by-12 lattice nodes. In this work, we use a much finer resolution of 40-by-40 nodes which becomes feasible due to the benefits of the sparse sequential implementation. The chosen resolution represents a spacing of 25 cm.

Regarding the exploratory mission, we consider a fixed maximum episode length of 150 s. This results in a total of 300 times steps, as we run our exploration algorithms with a step size of 0.5 s. Note that running the field exploration algorithm for longer episode lengths is indeed possible.

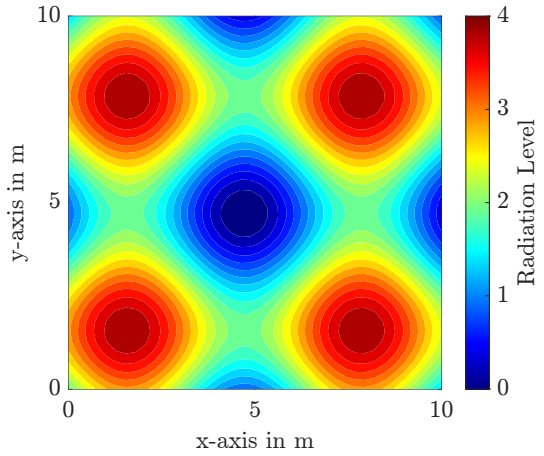


Figure 5.7: Reference field consisting of four continuously emitting radiation sources.

However, the chosen episode length is found sufficient to analyze the exploration performance of our algorithm in a confined scenario. The simulation setting are summarized ind Table 5.3

The simulation process consists of two steps. First, we train the DDQN-agent for 90,000 iterations (300 episodes with 300 iterations). Then, we evaluate the exploration performance using our developed IPP algorithm. Notice that we initialize the agents' position randomly and reset their environmental belief for each training and simulation episode. For our first feasibility study, we consider a single μAUV agent. In a second step, we extend our analysis to a full multi-robot fleet consisting of three μAUVs.

Table 5.2: Field Belief Hyperparameters

Parameter	**Symbol**	**Value**
Field Dimension	–	$10\,\mathrm{m} \times 10\,\mathrm{m}$
Belief Grid	–	40×40
Grid spacing	–	$0.25\,\mathrm{m}$
Boundary Padding	n_{pad}	2

Table 5.3: Simulation Parameters

Parameter	**Symbol**	**Value**
Episode Length	ℓ_{episode}	$150\,\mathrm{s}$
Time step	Δt	$0.5\,\mathrm{s}$
μAUV speed	v	$0.5\,\mathrm{m/s}$
Sensor noise	σ_z	0.2

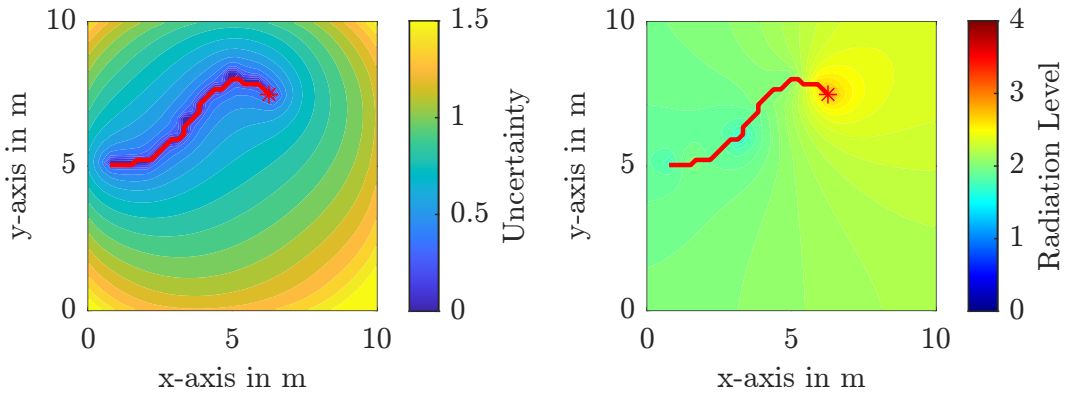
Table 5.4: Neural Network Hyperparameters

Parameter	**Symbol**	**Value**
Learning rate	α	0.001
Discount factor	γ	0.99
Initial Decrement	ϵ_0	0.5
Decrement step	ϵ_{dec}	0.001
Minibatch size	n_{batch}	1,000
Replay Buffer Size	–	100,000
Target Network Update	n_{update}	2,000
Planning Horizon	ℓ_o	$3\,\mathrm{m}$
Horizon Resolution	ξ	11×11

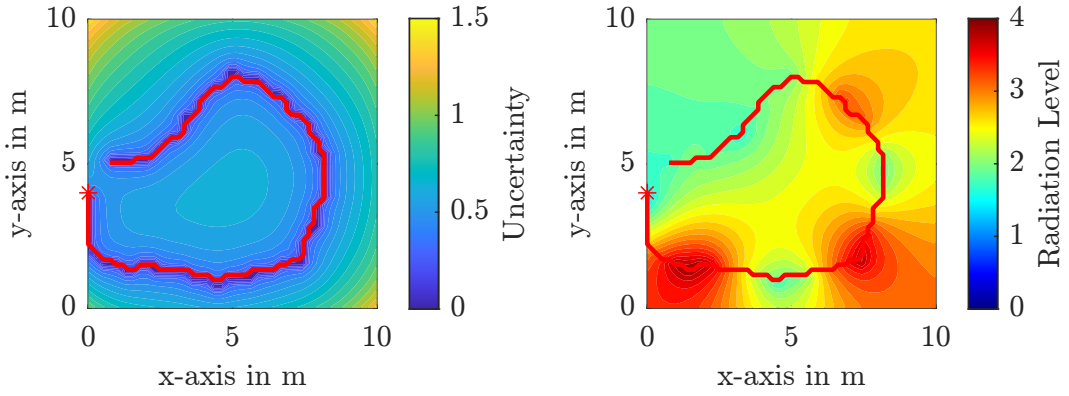
5.4.2 Feasibility Demonstration

As the first part of the following case study, we present a brief feasibility study of the proposed field exploration algorithm. Besides the quantitative analysis of performance metrics, it is very helpful in the development process of DRL algorithms to check whether the trained agent generally performs as expected. Various examples are presented in the literature where the agent identifies local rewards during training which can be exploited with such great success that the original main task is no longer targeted anymore. A prominent example for a suboptimally designed reward signal is the boat-race game discussed in [15]. Another risk of failed training arises from bugs inside the simulation environment. As a result, the DRL agent may exploit software bugs in the simulator's physic engine. A prominent example is a DRL agent that drives through walls inside a maze. This is of course undesired by the developers since it breaks the intended physics of the environment. However, the agent finds its policy to identify 'transparent' loophole walls as shortcuts to maximize its rewards.

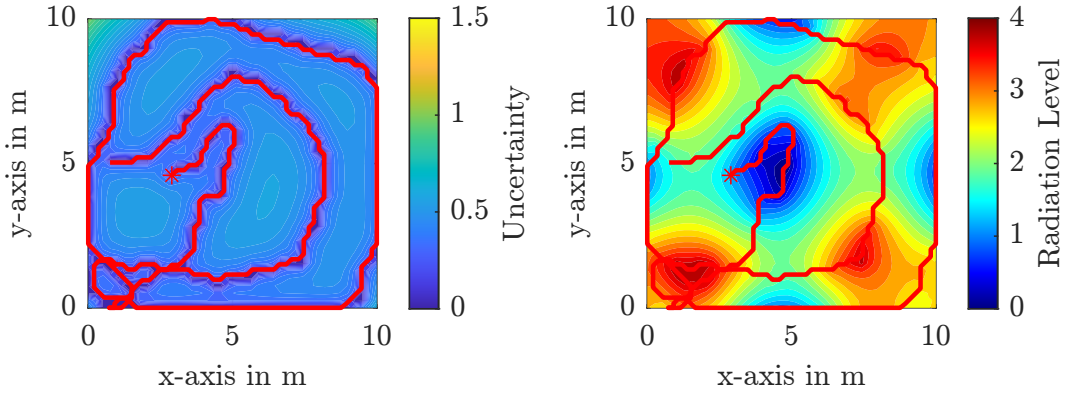
Following this motivation, we study various exploration episodes of which we exemplary present a single cycle. At the beginning of the exploration mission, the μAUV starts at position $(1\,\mathrm{m}, 5\,\mathrm{m})$, with heading direction $\psi = 0$. The vehicle follows the natural direction away from the field boundaries where the expected information gain is limited. The temporal evolution of the exploration is depicted in Figure 5.8. The depicted heatmaps represent the spatial field uncertainty (*left column*) and the corresponding estimated radiation field level (*right column*). Given the figures for time instance $t = 15\,\mathrm{s}$, we observe that the field belief inference algorithm temporarily estimates medium-intensity levels for the overall field. This is because the first segment of the driven path lies between two radiation sources, see also Figure 5.8a. The resulting medium-intensity measurements lead to an overestimate of the radiation field's minimum (*center*) and an underestimation of the radiation sources. Note that the areas of the radiation sources go along with a high field uncertainty (green/yellowish colored areas). In contrast, the field value estimates close to the vehicle are backed by low field uncertainties (dark blue areas). The underestimated radiation levels of the radiation sources are corrected as soon as the μAUV gathers measurements in their vicinity, see Figure 5.8b. Approximately at $t = 15\,\mathrm{s}$, the robot turns right before driving deep into the field's bottom right corner. In these corners, we would expect only small amounts of information to be gained given the effort of maneuvering close to the field boundaries. After 50 s, we observe that the vehicle proceeds on the outer side of the already driven track as it is repelled from the already gathered measurements. However, after finishing the outer circle, the μAUV crosses the already visited areas in the field's bottom-left corner to collect measurements from the field's center. Here, the perception horizon allows incorporating the expected future information gains which overcompensate the less rewarding area which has to be crossed first. After 150 s (300 time steps), the radiation field induced by the four radiation sources is successfully explored and the exploration episode is terminated.



(a) Timestep $t = 30$ (15 s)



(b) Timestep $t = 100$ (50 s)



(c) Timestep $t = 300$ (150 s)

Figure 5.8: Temporal evolution of exploration of radiation field induced by four sources. The estimated spatial field uncertainty is depicted on the left. Mean value estimates of the radiation field are shown in the right column.

5.4.3 Multi-Agent Environmental Field Exploration

In this study, we extend the previous analysis of a single agent to a fleet of μAUVs. Specifically, we consider the setting of $n_{\text{AUV}} = 3$ μAUV agents. Each agent runs its own instance of the DRL controller. During their mission, the μAUV agents share their current position, and their measurements, as illustrated in Figure 5.2. We assume for this scenario that the communication radius is sufficiently large that all-to-all communication is ensured. As a result, all agents possess their own instance of the shared GMRF belief representation. The agents' initial positions and orientations are randomly sampled for each episode. This random sampling also accounts for suboptimal starting position at the field's corners. The temporal evolution for a single episode is exemplary depicted in Figure 5.10.

Performance Analysis and Algorithm Comparison

Due to the stochastic nature of IPP tasks, studying single exploration episodes is not sufficient to reason about the performance of a planning methodology. Thus, a common approach is to evaluate the performance over a large number of randomly initialized trials. Among these, the absolute performance can be quantified by one or multiple performance metrics, for example, mean errors or uncertainty reduction. However, in order to draw generalizing conclusions it is beneficial to compare planning algorithms directly against each other by using existing methods as a performance baseline. This leads to the obvious goal of outperforming the baseline by the new algorithm. One of the most prominent baselines for sequential decision-making problems, such as IPP, is random walk (RW). To quantitatively compare our previously developed IPP method against RW we use the predictive variance sum (Equation (5.7)) of the complete field as a performance metric. A rapid reduction of the field's predictive variance corresponds to an efficient exploration of the environmental field.

To obtain meaningful findings within this stochastic setting, we run both simulations, for our DRL and the RW agent fleets, for 100 randomly initialized episodes. The temporal sequence of the predictive variance sum is depicted in Figure 5.9. Note that the predictive variance sum is not normally distributed among the 100 test runs. Thus, we compute the median and the inter-quartile ranges (IQRs) along the episodes' time series. The shaded areas in Figure 5.9 indicate the IQR where the bottomline and the topline indicate the lower 25 % and upper 75 %, respectively. We observe that both groups, our DRL and the RW agents, quickly reduce the uncertainty in the beginning. This is expected, since at the start all actions, hence including random ones, lead to considerable information gain. As the episode progresses, more and more spatial information is gathered leading to a continuously growing area of low uncertainty. This area should be avoided by the agents in order to keep their information gain high. While both agent groups start from the same initial uncertainty level, we observe that the DRL agents are quickly able to outperform the baseline RW agents. Furthermore, the performance of the

DRL agents varies much less than the RW agents' performance. This is indicated by the smaller shaded area. Regarding the quantitative analysis, we exemplary consider the required time spans (crossing times) to reduce the initial predictive variance sum ($\mathrm{VAR}\,[\mathcal{F}_0] = 1700$) by 50 % and by 70 %. We choose 70 % because it represents the median uncertainty reduction achieved by the baseline RW exploration strategy. The results for the median performance for both exploration algorithms are summarized in Table 5.5. Examining the results, we observe that the DRL-GMRF strategy is considerably faster in reducing the field uncertainty (24 s vs. 56 s for 50 % reduction). Note that the DRL-GMRF performance is more stable as the IQRs are much closer to the median than for the RW baseline, see Figure 5.9. Furthermore, the terminal field uncertainty reduction is larger for the DRL-GMRF scheme. We summarize that the results show for the given setting that the DRL agents considerably outperform the RW baseline agents with respect to the predictive variance sum metric.

Computational Consideration

The studied DRL-GMRF exploratory framework specifically targets the deployment onboard μAUVs. For the purpose, it is critical that the framework is lightweight enough to run at a sufficient high update rate. Note that the DRL-GMRF exploration algorithm takes the task of a high-level planner within the overall μAUV software architecture, as previously discussed in Section 3.1. Thus, it is not necessary to run the exploratory planner at a high rate since the computation of the final control commands is performed by the robot's low-level attitude controller. Furthermore, it is important to ensure a constant computational complexity over time. This is achieved by the GMRF's sequential update procedure as discussed in detail in Subsection 5.3.2. The presented simulations have been conducted on a desktop computer with Intel Xeon 8×3.6 GHz CPU, 32 GB RAM, and Nvidia Quadro P1000 GPU with 4 GB memory. However, this setting is indeed not representative for field deployment onboard computational restricted micro underwater robots. Following the prior discussion in Subsection 3.1.3, we run the exploration framework embedded onboard a RaspberryPi 4 SBC with 8 GB RAM which serves as the μAUV's companion computer. Evaluating the DRL-GMRF architecture as defined by Tables 5.4 and 5.2 for 100 episodes with 300 iterations results in an average episode processing

Table 5.5: Comparison of the required median crossing times to reduce the initial field uncertainty by 50 % and 70 %. The reduction of 70 % represents the minimum uncertainty the RW baseline is able to achieve within the episode length of $t = 150$ s. This corresponds to a remaining variance sum of $\mathrm{VAR}\,[\mathcal{F}] = 515$, cf. Figure 5.9.

Exploratory Controller	**Uncertainty Reduction**	
	50 %	70 %
DRL-GMRF	24 s	99 s
RW	56 s	150 s

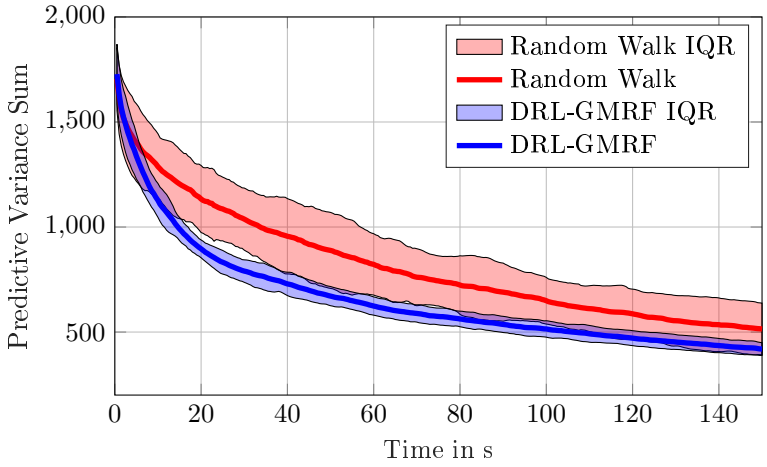
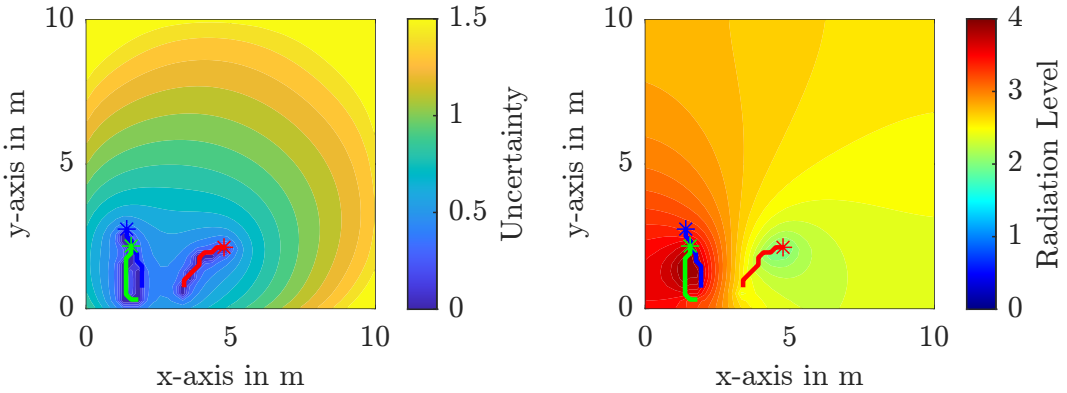


Figure 5.9: Exploration Performance of the DRL-GMRF scheme against Random Walk.

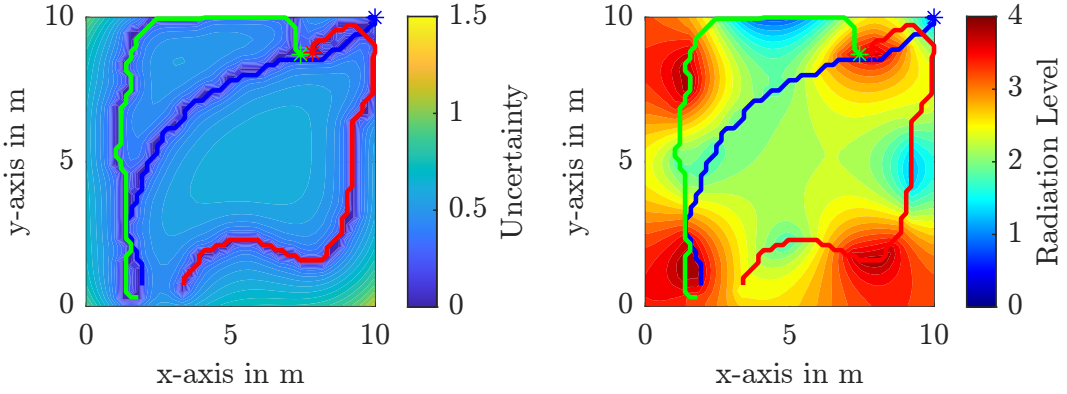
time of 40 s. This results in an execution rate of 7.5 Hz. This included the inference update of the GMRF based on new field measurements and computation of the next control action by means of the DRL controller. As a result, we can summarize that even a low-cost SBC such as a Raspberry Pi, the proposed framework achieves an update rate which is sufficiently fast for high-level planning in combination with a low-level attitude controller.

5.5 Summary and Discussion

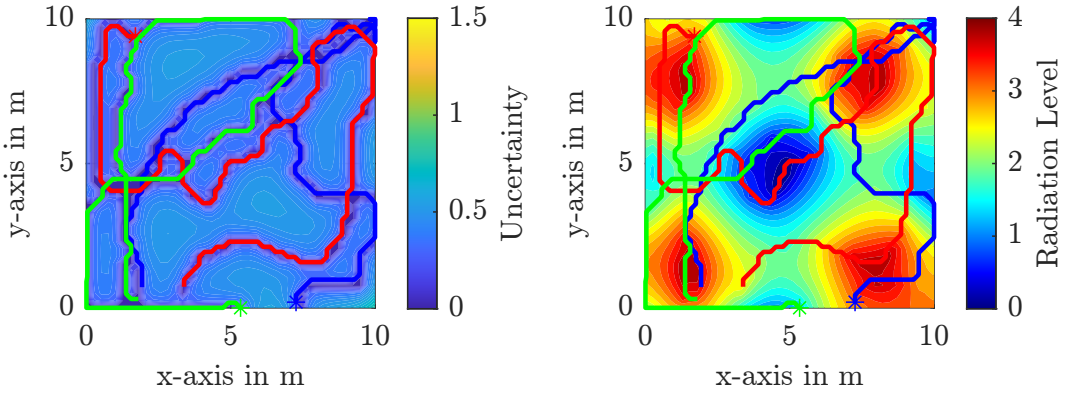
In this chapter, we studied the problem of IPP for environmental field exploration with μAUV groups. We proposed a modular multi-robot IPP framework that constitutes a combination of deep reinforcement learning and a stochastic field belief representation. The idea is to interpret the stochastic field belief in the local vicinity of the agent in an image-like fashion. These images are processed by a DDQN which has convolutional layers as an input to obtain Q-values for action selection. Our prior survey of related work (Subsection 2.4.3) revealed that current DRL exploration approaches mostly base their planning mostly on deterministic grid maps rather than stochastic representations. This motivated the incorporation of stochastic measures such as the field uncertainty into the reward signal. For this reason, a lightweight field belief representation based on Gaussian Markov random fields (GMRFs) was presented which fulfills the important criterion of having a constant computational complexity given an increasing number collected field measurements. We studied the feasibility of our proposed exploration framework and compare it against a random walk baseline in a multi-robot scenario. As a result, we observe that our DRL-GMRF approach clearly outperforms the baseline algorithm. Finally, our computational study demonstrated that our proposed DRL-GMRF exploratory framework is computationally lightweight enough to run on embedded μAUV hardware with sufficiently high update rates. The contents of this chapter have been partly published in [26, 32].



(a) Timestep $t = 10$ (5 s)



(b) Timestep $t = 70$ (35 s)



(c) Timestep $t = 150$ (75 s)

Figure 5.10: Temporal evolution of exploration of radiation field induced by four sources. The estimated spatial field uncertainty is depicted on the left. Mean value estimates of the radiation field are shown in the right column.

Chapter 6 Conclusion

6.1 Summary

Field exploration missions in confined environments demand for small-scale underwater robots with agile maneuvering capabilities. A literature survey in conjunction with a detailed problem analysis revealed research gaps in three areas, namely the absence of agile micro underwater vehicles, robust and accurate self-localization systems, and lightweight stochastic field exploration methods.

The capability gap of agile underwater robots is addressed in Chapter 3 where an embedded modular control and state estimation software architecture is proposed which is combined with a low-cost mechatronic robot design. Hardware-wise, this represents a complete revision of the original HippoCampus prototype. The proposed software architecture is highly adaptable and can easily be ported to other underwater robot platforms, e. g. the commercial BlueROV2 system. Considering the vehicle model, a detailed study of the HippoCampus vehicle dynamics revealed a close link between agile aerial quadrotor drones and the HippoCampus μAUV design. It was shown that for certain assumptions the design and propulsion concept yield the property of being differentially flat. This constitutes an important insight and greatly facilitates control design and path planning. With regard to low-level control, a nonlinear geometric control scheme was developed which can be used stand-alone or in conjunction with a high-level planning module. The HippoCampus platform was tested with regard to its control performance and maneuvering capabilities. For this purpose, a real-world experimental case study was designed and conducted. The agile control performance was successfully demonstrated by the vehicle's task to swing up and balance an inverted Furuta-pendulum. Notably, the HippoCampus platform is one of the first platforms within the recently established category of *hydrobatic* μAUVs. To the best of the author's knowledge, HippoCampus' combination of being low-cost, its high agility, and small vehicle size is currently unique within the family of small-scale underwater robots.

Self-localization of small-scale underwater robots used to be one of the most critical bottlenecks and is still hindering the development of μAUV systems. Based on the detailed technological analysis, it was found that there is no such technology available that solves the localization problem for all scenario conditions. Thus, a modular loosely coupled state estimation architecture is proposed in Chapter 4. In addition to this general architecture, three localization concepts based on different physical principles were developed. First, a visual approach using fiducial markers was introduced to address the demand for accurate reference self-localization systems. It is capable of providing precise high-rate position estimates that can be used for controller benchmarking in freshwater research tanks. The second approach is based on the carrier signal attenuation of electromagnetic waves. Prior works on this technique used off-board full-fledge spectrum analyzers to process the power spectrum density of the received carrier signals achieving update rates up to 1 kHz. Instead, the embedded approach in this dissertation uses a DVB-T USB dongle to process the signal. This allows building an small-scales localization system at a fraction of the cost of the original high-fidelity system. High localization accuracies were achieved and update rates in the range of 10-15 Hz were found sufficient for autonomous vehicle navigation. However, as a remaining challenge, it was observed that the accuracy is affected by the vehicle attitude. The two short-range concepts were supplemented by a third concept for medium-scale scenarios, e. g. marinas. This concept uses an acoustic two-way ranging localization scheme in combination with low-cost acoustic modems. Each localization approach was implemented onboard the μAUV hardware platform and tested in a physical experimental setting. Various experimental trials were conducted. The testing included outdoor testing at the Hamburg-Finkenwerder Marina, in the Hamburg-Wilhelmsburg public swimming pool, and in the research tank at the TU Hamburg's Institute of Mechanics and Ocean Engineering. Overall, all three approaches demonstrated very promising performance in these real-world settings. Most importantly, they fulfill the posed criterion of running on very restricted onboard hardware while still providing sufficient accuracy for μAUV navigation. Given the state-of-the-art, in μAUV guidance, navigation, and control this progress constitutes an important step towards higher levels of autonomy in μAUV missions.

Chapter 5 explored and contributed to the problem of autonomous environmental field exploration and monitoring with μAUVs. Based on a decentralized POMDP-formulation, a novel multi-robot informative path planning framework was proposed. The framework combines deep reinforcement learning (DRL), specifically double deep Q-learning, with a stochastic field belief representation which was realized by means of Gaussian Markov random field (GMRF). The key idea of the DRL-GMRF approach is that the locally rendered field belief representation is interpreted as an imaginary image which allows to apply powerful image processing techniques such as convolutional neural networks. A double deep Q-network including these convolutional layers and is used to provide Q-values for action selection. The highlights of this approach are two-fold. First, a combination stochastic field belief representation with a deep reinforcement learning-based informative path planning method was introduced. Second, in contrast to prior works, a sequential update formalism for the stochastic field belief representation is presented.

As a result, the proposed concept runs with constant computational complexity despite the growing number of gathered measurements. This is a considerable step forward when considering prior works using a similar belief representation for field exploration with μAUVs. The performance of the proposed scheme was analyzed as part of a two case studies that targeted the general feasibility and compared the exploration performance against a random walk baseline for a multi-robot scenario. As a result, the proposed DRL-GMRF approach clearly outperformed the baseline algorithm. Finally, the developed algorithm is implemented and tested on μAUV computing hardware to demonstrate that the proposed method is lightweight enough to run on restricted micro robot platforms. Overall, the findings demonstrate the promising potential of combining deep reinforcement learning techniques with stochastic field belief representation.

6.2 Future Directions

This dissertation introduced methods and techniques for μAUV hardware design and control as well as self-localization and a DRL-GMRF informative path planning algorithm. In order to further improve the field exploration performance, the following pointers are recommended for consideration.

Regarding the design of the HippoCampus μAUV platform, a detailed analysis of the inter-vehicle communication link followed by an improvement of the same would be desirable. Such an upgrade would increase the vehicles' ability to exchange more information over longer distances. As a consequence, the effective workspace which the fleet can cooperatively explore is increased. Furthermore, an upgrade of the onboard computational resources would allow maintaining larger and more complex field belief representations.

Considering the capabilities of μAUV self-localization, the proposed future directions are method-dependent. Regarding the visual localization scheme and its expected usage for control benchmarks render its extension to system of multiple onboard cameras a promising approach. This would allow to obtain accurate detections of the fiducial markers even for long sequences of extreme motions, e. g. driving upside-down, etc. However, additional cameras increase the computational burden. Localization based on electromagnetic ranging could be substantially improved for dynamic changes of the vehicle attitude. As discussed, this can be achieved by a better understanding of the antenna characteristics on the vehicle side. By developing more accurate antenna models the localization accuracy would be improved and today's mismatches could then be compensated adequately. Today's acoustic two-way ranging systems suffer from their comparatively low cycle time which limits the update rate of the position estimator. The recently introduced chip-scale atomic clock could be a promising attempt to switch the ranging scheme to synchronized one-way ranging. Such a scheme would come with the advantage of allowing for considerably higher update rates. Furthermore, one-way ranging facilitates scaling the fleet size to larger numbers as the present infrastructure can be used more effectively.

As pointed out in this dissertation, the combination of deep reinforcement learning and stochastic field belief representations is promising but is still in its infancy. However, such combined concepts offer great potential for challenging exploration tasks. Stochastic belief representations can be used as a common ground for various information sources. For example, information from heterogeneous sensing sources can be fused into the belief by considering the individual sensor characteristics, e. g. their measurement noise level. Extending this idea, a promising future research direction includes the introduction of heterogeneous μAUV fleets. Within these heterogeneous fleets, individual vehicles come with different capabilities. For instance, single vehicles may become *experts* by carrying a more sensitive sensor suite and sample probes from the environment. Regular vehicles that have faster marching speeds are sent-out beforehand to provide a rough estimate of the field which is then used to identify areas of high interest that are then efficiently explored in more detail by the expert-vehicles. Similarly, vehicles can be configured as mother-ships carrying an energy supply to recharge other vehicles during their mission to extend the overall mission time. Note that these approaches would considerably increase the mission complexity. Hence, planning and control methodologies have to be developed which can address this increased complexity.

Bibliography

[1] A. I. Al-Shamma'a, A. Shaw, and S. Saman. Propagation of electromagnetic waves at MHz frequencies through seawater. *IEEE Transactions on Antennas and Propagation*, 52(11):2843–2849, 2004.

[2] A. Arora, P. M. Furlong, R. Fitch, S. Sukkarieh, and T. Fong. Multi-modal active perception for information gathering in science missions. *Autonomous Robots*, 43(7):1827–1853, 2019.

[3] C. P. Bechlioulis, G. C. Karras, S. Heshmati-Alamdari, and K. J. Kyriakopoulos. Trajectory Tracking With Prescribed Performance for Underactuated Underwater Vehicles Under Model Uncertainties and External Disturbances. *IEEE Transactions on Control Systems Technology*, 25(2):429–440, 2017.

[4] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch. Dec-MCTS: Decentralized planning for multi-robot active perception. *International Journal of Robotics Research*, 38(2-3):316–337, 2019.

[5] M. Beul and S. Behnke. Fast full state trajectory generation for multirotors. In *International Conference on Unmanned Aircraft Systems, ICUAS 2017*, pages 408–416, Miami, FL, US, 2017. IEEE.

[6] S. Bhat. *Hydrobatics: Efficient and Agile Underwater Robots.* Phd-thesis, KTH Royal Institute of Technology, 2020.

[7] S. Bhat and I. Stenius. Hydrobatics: A Review of Trends, Challenges and Opportunities for Efficient and Agile Underactuated AUVs. In *IEEE/OES Autonomous Underwater Vehicle Symposium (AUV)*, pages 1–8, Porto, 2018. IEEE.

[8] J. Binney and G. S. Sukhatme. Branch and bound for informative path planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2147–2154, Saint Paul, US, 2012. IEEE.

[9] C. Böttcher, T. Knobloch, N.-P. Rühl, J. Sternheim, U. Wichert, and J. Wöhler. *Munitionsbelastung der deutschen Meeresgewässer: Bestandsaufnahme und Empfehlungen*. Bundesamt für Seeschifffahrt und Hydrographie (BSH), Hamburg, Germany, 2011 edition, 2011.

[10] A. D. Buchan, E. Solowjow, D. A. Duecker, and E. Kreuzer. Low-Cost Monocular Localization with Active Markers for Micro Autonomous Underwater Vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4181–4188, Vancouver, Canada, 2017. IEEE/RSJ.

[11] S. Carniel, J. Beldowski, and M. Edwards. Munitions in the sea. *Energetic Materials and Munitions: Life Cycle Management, Environmental Impact, and Demilitarization*, pages 139–167, 2019.

[12] W. Chen and L. Liu. Pareto Monte Carlo Tree Search for Multi-Objective Informative Planning. In *Robotics: Science and Systems (RSS)*, pages 1–10, Freiburg (Brsg), Germany, 2019.

[13] M. Chiou, R. Stolkin, G. Bieksaite, N. Hawes, K. L. Shapiro, and T. S. Harrison. Experimental analysis of a variable autonomy framework for controlling a remotely operating mobile robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3581–3588, Daejeon, Korea, 2016. IEEE/RSJ.

[14] S. Choudhury, N. Gruver, and M. J. Kochenderfer. Adaptive Informative Path Planning with Multimodal Sensing. In *Proceedings of the Thirtieth International Conference on Adaptive Automated Planning and Scheduling (ICAPS)*, pages 57–65, Nancy, France, 2020.

[15] J. Clark and D. Amodei. Faulty Reward Functions in the Wild. URL: `https://openai.com/blog/faulty-reward-functions/`, accessed on 2021-11-12.

[16] T. F. Clark. Tank Inspections Back Water Quality. *Opflow*, 43(7):18–21, 2017.

[17] R. Cui, Y. Li, and W. Yan. Mutual Information-Based Multi-AUV Path Planning for Scalar Field Sampling Using Multidimensional RRT*. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(7):993–1004, 2016.

[18] J. E. da Silva, B. Terra, R. Martins, and J. B. de Sousa. Modeling and Simulation of the LAUV Autonomous Underwater Vehicle. In *13th IEEE IFAC International Conference on Methods and Models in Automation and Robotics*, pages 1–6, Szczecin, Poland, 2007. IEEE/IFAC.

[19] A. Dikarev, A. Griffiths, S. Watson, B. Lennox, and P. R. Green. Combined multiuser acoustic communication and localisation system for microAUVs operating in confined underwater environments. *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*, 48(2):161–166, 2015.

[20] V. Djapic, W. Dong, D. Spaccini, G. Cario, A. Casavola, P. Gjanci, M. Lupia, and C. Petrioli. Cooperation of coordinated teams of Autonomous Underwater Vehicles. *IFAC-PapersOnLine*, 49(15):88–93, 2016.

[21] D. B. dos Santos Cesar, C. Gaudig, M. Fritsche, M. A. dos Reis, and F. Kirchner. An Evaluation of Artificial Fiducial Markers in Underwater Environments. In *IEEE/MTS OCEANS*, pages 1–6, Genova, Switzerland, 2015. IEEE/MTS.

[22] D. A. Duecker, N. Bauschmann, T. Hansen, E. Kreuzer, and R. Seifried. HippoCampus X – A Hydrobatic Open-source Micro AUV for Confined Environments. In *IEEE/OES Autonomous Underwater Vehicle Symposium (AUV)*, pages 1–6, St. Johns, Canada (Virtual), 2020. IEEE.

[23] D. A. Duecker, N. Bauschmann, T. Hansen, E. Kreuzer, and R. Seifried. Towards Micro Robot Hydrobatics: Vision-based Guidance, Navigation, and Control for Agile Underwater Vehicles in Confined Environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1819–1826, Las Vegas, NV, USA (Virtual), 2020. IEEE.

[24] D. A. Duecker, K. Eusemann, and E. Kreuzer. Towards an Open-Source Micro Robot Oceanarium: A Low-Cost, Modular, and Mobile Underwater Motion-Capture System. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8048–8053, Macau SAR, 2019. IEEE/RSJ.

[25] D. A. Duecker, A. R. Geist, M. Hengeler, E. Kreuzer, M. A. Pick, V. Rausch, and E. Solowjow. Embedded spherical localization for micro underwater vehicles based on attenuation of electro-magnetic carrier signals. *Sensors (Switzerland)*, 17(5):1–22, 2017.

[26] D. A. Duecker, A. R. Geist, E. Kreuzer, and E. Solowjow. Learning environmental field exploration with computationally constrained underwater robots: Gaussian processes meet stochastic optimal control. *Sensors (Switzerland)*, 19(9):1–28, 2019.

[27] D. A. Duecker, A. Hackbarth, T. Johannink, E. Kreuzer, and E. Solowjow. Micro Underwater Vehicle Hydrobatics: A Submerged Furuta Pendulum. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7498–7503, Brisbane, 2018. IEEE.

[28] D. A. Duecker, T. Hansen, and E. Kreuzer. RGB-D Camera-based Navigation for Autonomous Underwater Inspection using Low-cost Micro AUVs. In *IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*, pages 1–7, St. Johns, Canada (Virtual), 2020. IEEE.

[29] D. A. Duecker, C. Horst, and E. Kreuzer. From Aerobatics to Hydrobatics: Agile Trajectory Planning and Tracking for Micro Underwater Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8, Prague (virtual), CZ, 2021. IEEE/RSJ.

[30] D. A. Duecker, T. Johannink, E. Kreuzer, V. Rausch, and E. Solowjow. An Integrated Approach to Navigation and Control in Micro Underwater Robotics using Radio-Frequency Localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6846–6852, Montreal, Canada, 2019. IEEE.

[31] D. A. Duecker, E. Kreuzer, G. Maerker, and E. Solowjow. Parameter Identification for Micro Underwater Vehicles. In *Proceedings in Applied Mathematics and Mechanics (PAMM)*, pages 1–2, Munich, Germany, 2018. WILEY-VCH.

[32] D. A. Duecker, B. Mersch, R. C. Hochdahl, and E. Kreuzer. Embedded Stochastic Field Exploration with Micro Diving Agents using Bayesian Optimization-Guided Tree-Search and GMRFs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8, Prague (virtual), CZ, 2021. IEEE/RSJ.

[33] D. A. Duecker, F. Steinmetz, E. Kreuzer, and B. C. Renner. Micro AUV Localization for Agile Navigation with Low-cost Acoustic Modems. In *IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*, pages 1–7, St. Johns, Canada (Virtual), 2020. IEEE.

[34] C. Edge, S. S. Enan, M. Fulton, J. Hong, J. Mo, K. Barthelemy, H. Bashaw, B. Kallevig, C. Knutson, K. Orpen, and J. Sattar. Design and Experiments with LoCO AUV: A Low Cost Open-Source Autonomous Underwater Vehicle. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1761–1768, Las Vegas, NV, USA (Virtual), 2020. IEEE/RSJ.

[35] K. Ehlers, C. Isokeit, B. Meyer, U. Behrje, and E. Maehle. Using Generalised Self-Organizing Maps as Part of Underwater Localisation for Quay Wall Inspections. In *Global Oceans 2020: Singapore – U.S. Gulf Coast*, pages 1–8, Biloxi, MS, USA, 2020. IEEE.

[36] M. Fliess, J. Levine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: Introductory theory and examples. *International Journal of Control*, 61(6):1327–1361, 1995.

[37] T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control.* Wiley, 2011.

[38] H. T. Friis. Simple Transmission Formula. *Proceedings of the IRE*, 34(5):254–256, 1946.

[39] H. T. Friis. Introduction to radio and radio antennas. *IEEE Spectrum*, 8(4):55–61, 1971.

[40] K. Furuta, M. Yamakita, and S. Kobayashi. Swing Up Control of Inverted Pendulum. In *International Conference on Industrial Electronics, Control and Instrumentation*, pages 2193–2198, Kobe, Japan, 1991. IEEE.

[41] K. Furuta, M. Yamakita, and S. Kobayashi. Swing-up control of inverted pendulum using pseudo-state feedback. *Journal of Systems and Control Engineering*, 206(4):263–267, 1992.

[42] A. R. Geist, A. Hackbarth, E. Kreuzer, V. Rausch, M. Sankur, and E. Solowjow. Towards a hyperbolic acoustic one-way localization system for underwater swarm robotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4551–4556, Stockholm, Sweden, 2016. IEEE.

[43] A. Griffiths, A. Dikarev, P. R. Green, B. Lennox, X. Poteau, and S. Watson. AVEXIS - Aqua vehicle explorer for in-situ sensing. *IEEE Robotics and Automation Letters*, 1(1):282–287, 2016.

[44] A. Hackbarth, J. Hedrick, E. Kreuzer, and E. Solowjow. A Nonlinear Estimator for Mobile Source Localization Applications. In *ENOC*, Vienna, Austria, 2014.

[45] A. Hackbarth, E. Kreuzer, and T. Schröder. CFD in the Loop: Ensemble Kalman Filtering With Underwater Mobile Sensor Networks. In *Proceedings of the 33rd International Conference on Ocean, Offshore and Arctic Engineering*, pages 1–8, San Francisco, CA, USA, 2014. ASME.

[46] A. Hackbarth, E. Kreuzer, and E. Solowjow. HippoCampus: A micro underwater vehicle for swarm applications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2258–2263, Hamburg, Germany, 2015. IEEE/RSJ.

[47] H. Hanff, P. Kloss, B. Wehbe, P. Kampmann, S. Kroffke, A. Sander, M. B. Firvida, M. V. Einem, J. F. Bode, and F. Kirchner. AUVx - A Novel Miniaturized Autonomous Underwater Vehicle. In *Oceans*, pages 1–10, Aberdeen, UK, 2017. IEEE.

[48] H. Hanff, K. Schmid, P. Kloss, and S. Kroffke. μAUV2 - Development of a Minuscule Autonomous Underwater Vehicle. In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics*, pages 185–196, Lisbon, Portugal, 2016.

[49] A. J. Healey, S. M. Rock, S. Cody, D. Miles, and J. P. Brown. Toward an Improved Understanding of Thruster Dynamics for Underwater Vehicles. *IEEE Journal of Oceanic Engineering*, 20(4):354–361, 1995.

[50] M. Hehn and R. D'Andrea. A Flying Inverted Pendulum. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 763–770, Shanghai, China, 2011. IEEE.

[51] J. Heitmann, F. Steinmetz, and B. C. Renner. Self-Localization of Micro AUVs Using a Low-Power , Low-Cost Acoustic Modem. In *OCEANS 2018 MTS/IEEE*, pages 72–74, Charleston, US, 2018. IEEE.

[52] E. H. Henriksen, I. Schjølberg, and T. B. Gjersvik. Vision Based Localization for Subsea Intervention. In *Proceedings of the ASME 2017 36th International Conference on Offshore Mechanics and Arctic Engineering (OMAE)*, pages 1–9, Trondheim, Norway, 2017.

[53] S. Heshmati-Alamdari, G. C. Karras, P. Marantos, and K. J. Kyriakopoulos. A Robust Model Predictive Control Approach for Autonomous Underwater Vehicles Operating in a Constrained workspace. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6183–6188, Brisbane, Australia, 2018. IEEE.

[54] G. Hollinger and G. S. Sukhatme. Stochastic motion planning for robotic information gathering. In *Robotics: Science and Systems (RSS)*, 2013.

[55] G. A. Hollinger and G. S. Sukhatme. Sampling-based robotic information gathering algorithms. *International Journal of Robotics Research*, 33(9):1271–1287, 2014.

[56] M. Ishida and K. Shimonomura. Marker Based Camera Pose Estimation for Underwater Robots. In *IEEE/SICE International Symposium on System Integration*, pages 629–634, Fukuoka, 2012. IEEE/SICE.

[57] S. F. Jackson, S. D. Monk, and Z. Riaz. An investigation towards real time dose rate monitoring, and fuel rod detection in a First Generation Magnox Storage Pond (FGMSP). *Applied Radiation and Isotopes*, 94:254–259, 2014.

[58] M. Jadaliha, J. Jeong, Y. Xu, J. Choi, and J. Kim. Fully Bayesian Prediction Algorithms for Mobile Robotic Sensors under Uncertain Localization Using Gaussian Markov Random Fields. *Sensors (Switzerland)*, 18(9):2866, 2018.

[59] Jensen, T. Dempster, E. B. Thorstad, I. Uglem, and A. Fredheim. Escapes of fishes from Norwegian sea-cage aquaculture: Causes, consequences and prevention. *Aquaculture Environment Interactions*, 1(1):71–83, 2010.

[60] J. Journée and W. Massie. *Offshore Hydromechanics*, volume 1. TU Delft, 2000.

[61] J. Jung, Y. Lee, D. Kim, D. Lee, H. Myung, and H.-t. Choi. AUV SLAM Using Forward / Downward Looking Cameras and Artificial Landmarks. *IEEE Underwater Technology*, pages 1–3, 2017.

[62] J. Jung, J. H. Li, H. T. Choi, and H. Myung. Localization of AUVs using visual information of underwater structures and artificial landmarks. *Intelligent Service Robotics*, 10(1):67–76, 2017.

[63] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

[64] S. Kemna, J. G. Rogers, C. Nieto-Granda, S. Young, and G. S. Sukhatme. Multi-robot coordination through dynamic Voronoi partitioning for informative adaptive sampling in communication-constrained environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2124–2130, Singapore, 2017. IEEE.

[65] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, pages 1–15, San Diego, CA, US, 2015.

[66] D. J. Klein, P. K. Bettale, B. Triplett, and K. A. Morgansen. Autonomous underwater multivehicle control with limited communication: Theory and experiment. *Proceedings of the 2008 IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*, 41(1):113–118, 2008.

[67] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[68] E. Kreuzer and E. Solowjow. Learning environmental fields with micro underwater vehicles: a path integral—Gaussian Markov random field approach. *Autonomous Robots*, 42(4):761–780, 2018.

[69] D. G. Krige. A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, pages 201–215, 1952.

[70] K. Kwak, D. Park, W. K. Chung, and J. Kim. Underwater 3D Spatial Attenuation Characteristics of Electromagnetic Waves with Omnidirectional Antenna. *IEEE/ASME Transactions on Mechatronics*, 21(3):1409—-1419, 2016.

[71] K. Kwak, D. Park, W. K. Chung, and J. Kim. Manta ROV docking sequence using 3-D Omni-directional antenna ' s Signal Attenuation. In *4th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 858–859, Jeju, 2017. IEEE.

[72] T. Lee. Robust adaptive attitude tracking on SO(3) with an application to a quadrotor UAV. *IEEE Transactions on Control Systems Technology*, 21(5):1924–1930, 2013.

[73] T. Lee, M. Leoky, and N. McClamroch. Geometric Tracking Control of a Quadrotor UAV on SE(3). In *49th IEEE Conference on Decision and Control (CDC)*, pages 5420–5425, Atlanta, 2010. IEEE.

[74] E. V. Lewis. *Principles of Naval Architecture Second Revision*. Society of Naval Architects and Marine Engineers, Jersey City, NY, 1989.

[75] F. Lindgren, H. Rue, and J. Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society, Series B*, 73:423–498, 2011.

[76] A. Mallios, P. Ridao, D. Ribas, and E. Hernández. Scan matching SLAM in underwater environments. *Autonomous Robots*, 36(3):181–198, 2014.

[77] D. Malyuta, C. Brommer, D. Hentzen, T. Stastny, R. Siegwart, and R. Brockers. Long-duration fully autonomous operation of rotorcraft unmanned aerial systems for remote-sensing data acquisition. *Journal of Field Robotics*, August:1–21, 2019.

[78] R. Marchant and F. Ramos. Bayesian Optimisation for informative continuous path planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6136–6143, Hong Kong, 2014. IEEE.

[79] L. Meier, D. Honegger, and M. Pollefeys. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6235–6240, Seattle, 2015. IEEE.

[80] D. Mellinger and V. Kumar. Minimum Snap Trajectory Generation and Control for Quadrotors. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2520–2525, Shanghai, 2011. IEEE.

[81] P. H. Milne. *Underwater Acoustic Positioning Systems*. E. F. N. Spon, London, 1983.

[82] S. Mintchev, E. Donati, S. Marrazza, and C. Stefanini. Mechatronic design of a miniature underwater robot for swarm operations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2938–2943, Hong Kong, 2014. IEEE.

[83] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[84] V. Mnih, A. Puigdomènech Badia, M. Mirza, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. In *International Conference on Machine Learning*, pages 1928–1937. PMLR, 2016.

[85] P. Morere, R. Marchant, and F. Ramos. Continuous State-Action-Observation POMDPs for Trajectory Planning with Bayesian Optimisation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8779–8786, Madrid, Spain, 2018. IEEE/RSJ.

[86] M. W. Mueller, M. Hehn, and R. D. Andrea. A Computationally Efficient Motion Primitive for Quadrocopter Trajectory Generation. *IEEE Transactions on Robotics*, 31(6):1294–1310, 2015.

[87] M. Nancekievill, A. R. Jones, M. J. Joyce, B. Lennox, S. Watson, J. Katakura, K. Okumura, S. Kamada, M. Katoh, and K. Nishimura. Development of a Radiological Characterization Submersible ROV for Use at Fukushima Daiichi. *IEEE Transactions on Nuclear Science*, 65(9):2565–2572, 2018.

[88] S. Napora and D. A. Paley. Observer-Based Feedback Control for Stabilization of Collective Motion. *IEEE Transactions on Control Systems Technology*, 21(5):1846–1857, 2013.

[89] AdvancedNavigation. Subsonus. URL: `www.advancednavigation.com/products`, accessed on 2021-08-13.

[90] Navilock. Navilock NL-8001U Datasheet. URL: `https://www.navilock.com/produkte/G_62573/merkmale.html`, accessed on 2021-08-13.

[91] J. N. Newman. *Marine Hydrodynamics*. MIT Press, Cambridge, MA, 40th edition, 2018.

[92] F. A. Oliehoek and C. Amato. *A Concise Introduction to Decentralized POMDPs*. Springer, 2016.

[93] E. Olson. AprilTag : A robust and flexible visual fiducial system. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407, Shanghai, 2011. IEEE.

[94] D. A. Paley. *Cooperative Control of Collective Motion for Ocean Sampling with Autonomous Vehicles*. Phd-thesis, Princeton University, 2007.

[95] D. A. Paley. Stabilization of collective motion on a sphere. *Automatica*, 45(1):212–216, 2009.

[96] D. Park, J. Jung, K. Kwak, J. Kim, and W. K. Chung. Received Signal Strength of Electromagnetic Waves Aided Integrated Inertial Navigation System for Underwater Vehicle. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1870–1876, Madrid, Spain, 2018. IEEE/RSJ.

[97] D. Park, K. Kwak, W. K. Chung, and J. Kim. Development of Underwater Short-Range Sensor Using Electromagnetic Wave Attenuation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5125–5130, Karlsruhe, 2013. IEEE.

[98] D. Park, K. Kwak, W. K. Chung, and J. Kim. Development of Underwater Short-Range Sensor Using Electromagnetic Wave Attenuation. *IEEE Journal of Oceanic Engineering*, 41(2):318,325, 2015.

[99] D. Park, K. Kwak, J. Kim, and W. K. Chung. Underwater sensor network using received signal strength of electromagnetic waves. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1052–1057, Hamburg, 2015. IEEE/RSJ.

[100] D. Park, K. Kwak, J. Kim, and W. K. Chung. 3D Underwater Localization Scheme using EM Wave Attenuation with a Depth Sensor. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2631–2636, Stockholm, Sweden, 2016. IEEE.

[101] PX4-Community. PX4 - Open Source Autopilot for Drones. URL: `www.px4.io`, accessed on 2021-08-12.

[102] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source Robot Operating System. In *ICRA workshop on open source Software*, pages 1–2, Kobe, Japan, 2009. IEEE.

[103] B. C. Renner, J. Heitmann, and F. Steinmetz. AHOI: Inexpensive, low-power communication and localization for underwater sensor networks and μAUVs. *ACM Transactions on Sensor Networks*, 16(2):1–46, 2020.

[104] C. Renner. Packet-based ranging with a low-power, low-cost acoustic modem for micro AUVs. In *11th International ITG Conference on Systems, Communications and Coding*, pages 1–6, Hamburg, 2017.

[105] H. Rue and L. Held. *Gaussian Markov Random Fields*. Chapman and Hall, 2005.

[106] P. Seman, B. Rohal'-Ilkiv, M. Juhás, and M. Salaj. Swinging up the furuta pendulum and its stabilization via model predictive control. *Journal of Electrical Engineering*, 64(3):152–158, 2013.

[107] J. Shen, A. K. Sanyal, N. A. Chaturvedi, D. Bernstein, and H. McClamroch. Dynamics and control of a 3D pendulum. In *IEEE Conference on Decision and Control*, pages 323–328, Nassau, 2004. IEEE.

[108] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza. *Introduction to Autonomous Mobile Robots*. Intelligent robotics and autonomous agents. MIT Press, Cambridge, Massachusetts, 2nd edition, 2011.

[109] D. Silver and J. Veness. Monte-Carlo Planning in Large POMDPs. In *Advances in neural information processing systems*, pages 1–9, 2010.

[110] E. Solowjow. *Design, Dynamics, and Control of Micro Underwater Vehicle Systems for Autonomous Environmental Exploration*. Phd-thesis, Hamburg University of Technology, 2017.

[111] G. S. Spagnolo, L. Cozzella, and F. Leccese. Underwater optical wireless communications: Overview. *Sensors (Switzerland)*, 20(8), 2020.

[112] Z. N. Sunberg and M. J. Kochenderfer. Online algorithms for POMDPs with continuous state, action, and observation spaces. *Proceedings International Conference on Automated Planning and Scheduling, ICAPS*, June(Icaps):259–263, 2018.

[113] R. S. Sutton and A. G. Barto. *Reinforcement learning: An Introduction*. MIT press, 2018.

[114] N. Sydney, S. Napora, S. Beal, P. Mohl, P. Nolan, S. Sherman, A. Leishman, S. Butail, and D. A. Paley. A Micro-UUV Testbed for Bio-Inspired Motion Coordination. *Int. Symp. Unmanned Untethered Submersible Technology*, pages 1–13, 2009.

[115] N. Sydney and D. A. Paley. Multivehicle coverage control for a nonstationary spatiotemporal field. *Automatica*, 50(5):1381–1390, 2014.

[116] H.-P. Tan, R. Diamant, W. K. G. Seah, and M. Waldmeyer. A Survey of Techniques and Challenges in Underwater Localization. *Ocean Engineering*, 38(14-15):1663–1676, 2011.

[117] Q. Tao, Y. Zhou, F. Tong, A. Song, and F. Zhang. Evaluating acoustic communication performance of micro autonomous underwater vehicles in confined spaces. *Frontiers of Information Technology and Electronic Engineering*, 19(8):1013–1023, 2018.

[118] M. Theile, H. Bayerlein, R. Nai, D. Gesbert, and M. Caccamo. UAV coverage path planning under varying power constraints using deep reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1444–1449, Las Vegas, NV, USA (Virtual), 2020. IEEE/RSJ.

[119] M. Theile, H. Bayerlein, R. Nai, D. Gesbert, and M. Caccamo. UAV Path Planning using Global and Local Map Information with Deep Reinforcement Learning. In *International Conference on Advanced Robotics (ICAR)*, pages 539–546, 2021.

[120] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[121] S. Thrun and A. Schwartz. Issues in using function approximation for reinforcement learning. In *Proceedings of the Fourth Connectionist Models Summer School*, pages 255–263, Hillsdale, NJ, 1993.

[122] E. Topham and D. McMillan. Sustainable decommissioning of an offshore wind farm. *Renewable Energy*, 102:470–480, 2017.

[123] R. J. Urick. *Principles of underwater sound*. McGraw-Hill, New York, NY, 3rd edn, 1983.

[124] H. Van Hasselt. Double Q-learning. In *Advances in Neural Information Processing Systems*, pages 2613–2621, Vancouver, Canada, 2010.

[125] H. van Hasselt, A. Guez, and D. Silver. Learning Multi-Robot Decentralized Macro-Action-Based Policies via a Centralized Q-Net. In *AAAI 30th Conference on Artificial Intelligence*, pages 2094–2100, Phoenix, AZ, US, 2016.

[126] M. J. Van Nieuwstadt and R. M. Murray. Real-time trajectory generation for differentially flat systems. *International Journal of Robust and Nonlinear Control*, 8(11):995–1020, 1998.

[127] A. Viseras and R. Garcia. DeepIG: Multi-robot information gathering with deep reinforcement learning. *IEEE Robotics and Automation Letters*, 4(3):3059–3066, 2019.

[128] A. Viseras, M. Meissner, and J. Marchal. Wildfire Front Monitoring with Multiple UAVs using Deep Q-Learning. *IEEE Access*, pages 1–14, 2021.

[129] C. Wang, T. Li, M. Q. Meng, and C. De Silva. Efficient Mobile Robot Exploration with Gaussian Markov Random Fields in 3D Environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5015–5021, Brisbane, Australia, 2018. IEEE.

[130] J. Wang and E. Olson. AprilTag 2 : Efficient and robust fiducial detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2–7, Deajeon, South Korea, 2016. IEEE/RSJ.

[131] Waterlinked. Underwater GPS. URL: `https://waterlinked.com/underwater-gps/`, accessed on 2021-08-13.

[132] C. J. C. H. Watkins and P. Dayan. Machine Learning based Resource Allocation Strategy for Network Slicing in Vehicular Networks. *Machine Learning*, 8:279–292, 1992.

[133] S. Watson, D. A. Duecker, and K. Groves. Localisation of unmanned underwater vehicles (UUVs) in complex and confined environments: A review. *Sensors (Switzerland)*, 20(21):1–35, 2020.

[134] E. Westman and M. Kaess. Underwater AprilTag SLAM and calibration for high precision robot localization. Technical Report, Carnegie Mellon University, Pittsburgh, PA, 2018.

[135] Y. Xu and J. Choi. Spatial prediction with mobile sensor networks using Gaussian processes with built-in Gaussian Markov random fields. *Automatica*, 48(8):1735–1740, 2012.

[136] Y. Xu, J. Choi, S. Dass, and T. Maiti. Efficient Bayesian spatial prediction with mobile sensor networks using Gaussian Markov random fields. *Automatica*, 49(12):3520–3530, 2013.

[137] Y. Xu, J. Choi, S. Dass, and T. Maiti. *Bayesian Prediction and Adaptive Sampling Algorithms for Mobile Sensor Networks: Online Environmental Field Reconstruction in Space and Time*. Springer, Berlin, 2015.

[138] Y. Xu, J. Choi, and S. Oh. Mobile sensor network navigation using Gaussian processes with truncated observations. *IEEE Transactions on Robotics*, 27(6):1118–1131, 2011.

[139] G. Z. Yang, J. Cambias, K. Cleary, E. Daimler, J. Drake, P. E. Dupont, N. Hata, P. Kazanzides, S. Martel, R. V. Patel, V. J. Santos, and R. H. Taylor. Medical robotics-Regulatory, ethical, and legal considerations for increasing levels of autonomy. *Science Robotics*, 2(4):1–2, 2017.

[140] N. Ye, A. Somani, D. Hsu, and W. S. Lee. DESPOT: Online POMDP planning with regularization. *Journal of Artificial Intelligence Research*, 58:231–266, 2017.

[141] T. Yoo, M. Kim, S. Yoon, and D. Kim. Performance Enhancement for Conventional Tightly Coupled INS/DVL Navigation System Using Regeneration of Partial DVL Measurements. *Journal of Sensors*, 2020, 2020.

Appendix

A Sequential GMRF Inference

First, Equation (5.28) is rewritten into a sum of vector products with $\mathbf{\Phi}_{k,j}$ denoting the interpolation matrix of the j-th agent at discrete time step k, such that

$$\begin{aligned}\mathbf{\Lambda}_{k|\boldsymbol{\theta}} &= \mathbf{\Lambda}_{k-1|\boldsymbol{\theta}} + \frac{1}{\sigma_y^2}\mathbf{\Phi}_k^\top \mathbf{\Phi}_k = \mathbf{\Lambda}_{k-1|\boldsymbol{\theta}} + \sum_{j=1}^{N} \frac{1}{\sigma_y^2}\mathbf{\Phi}_{k,j}^\top \mathbf{\Phi}_{k,j}, \\ &= \underbrace{\underbrace{\mathbf{\Lambda}_{k-1|\boldsymbol{\theta}} + \frac{1}{\sigma_y^2}\mathbf{\Phi}_{k,1}^\top \mathbf{\Phi}_{k,1}}_{\mathbf{\Lambda}_{k-1,1|\boldsymbol{\theta}}} + \frac{1}{\sigma_y^2}\mathbf{\Phi}_{k,2}^\top \mathbf{\Phi}_{k,2} + \ldots + \frac{1}{\sigma_y^2}\mathbf{\Phi}_{k,N}^\top \mathbf{\Phi}_{k,N}}_{\mathbf{\Lambda}_{k-1,N|\boldsymbol{\theta}} = \mathbf{\Lambda}_{k|\boldsymbol{\theta}}}.\end{aligned} \tag{A.1}$$

By analyzes of the structure of Equation (A.1) an update rule is obtained that allows the sequential incorporation of new measurements as

$$\mathbf{\Sigma}_{k-1,j|\boldsymbol{\theta}} = \mathbf{\Lambda}_{k-1,j|\boldsymbol{\theta}}^{-1} = \left(\mathbf{\Lambda}_{k-1,j-1|\boldsymbol{\theta}} + \frac{1}{\sigma_y^2}\mathbf{\Phi}_{k,j}^\top \mathbf{\Phi}_{k,j}\right)^{-1}. \tag{A.2}$$

Applying the Sherman-Morrison formula on (A.2) results in

$$\mathbf{\Sigma}_{k-1,j|\boldsymbol{\theta}} = \mathbf{\Sigma}_{k-1,j-1|\boldsymbol{\theta}} - \frac{\mathbf{\Sigma}_{k-1,j-1|\boldsymbol{\theta}}\mathbf{\Phi}_{k,j}^\top \mathbf{\Phi}_{k,j}\mathbf{\Sigma}_{k-1,j-1|\boldsymbol{\theta}}}{\sigma_y^2 + \mathbf{\Phi}_{k,j}\mathbf{\Sigma}_{k-1,j-1|\boldsymbol{\theta}}\mathbf{\Phi}_{k,j}^\top}. \tag{A.3}$$

With Equation (A.1) and Equation (A.3), the conditional covariance matrix is obtained as

$$\mathbf{\Sigma}_{k|\boldsymbol{\theta}} = \mathbf{\Sigma}_{k-1|\boldsymbol{\theta}} - \sum_{j=1}^{N} \frac{\mathbf{\Sigma}_{k-1,j-1|\boldsymbol{\theta}}\mathbf{\Phi}_{k,j}^\top \mathbf{\Phi}_{k,j}\mathbf{\Sigma}_{k-1,j-1|\boldsymbol{\theta}}}{\sigma_y^2 + \mathbf{\Phi}_{k,j}\mathbf{\Sigma}_{k-1,j-1|\boldsymbol{\theta}}\mathbf{\Phi}_{k,j}^\top} = \mathbf{\Sigma}_{k-1|\boldsymbol{\theta}} - \sum_{j=1}^{N} \frac{h_{k,j}h_{k,j}^\top}{\sigma_y^2 + \mathbf{\Phi}_{k,j}h_{k,j}}. \tag{A.4}$$

Thus, the sequential update rule for the conditional variance can be written as

$$\operatorname{diag}(\boldsymbol{\Sigma}_{k|\boldsymbol{\theta}}) = \operatorname{diag}(\boldsymbol{\Sigma}_{k-1|\boldsymbol{\theta}}) - \sum_{j=1}^{N} \frac{\boldsymbol{h}_{k,j} \circ \boldsymbol{h}_{k,j}}{\sigma_y^2 + \boldsymbol{\Phi}_{k,j}\boldsymbol{h}_{k,j}}, \text{ with} \tag{A.5}$$

$$\boldsymbol{h}_{k,j} = \boldsymbol{\Sigma}_{k-1,j|\boldsymbol{\theta}}\boldsymbol{\Phi}_{k,j}^{\top} = \boldsymbol{\Lambda}_{k-1,j|\boldsymbol{\theta}}^{-1}\boldsymbol{\Phi}_{k,j}^{\top}, \tag{A.6}$$

$$\boldsymbol{\Lambda}_{k-1,j|\boldsymbol{\theta}} = \boldsymbol{\Lambda}_{k-1|\boldsymbol{\theta}} + \sum_{j=1}^{N} \frac{1}{\sigma_y^2}\boldsymbol{\Phi}_{k,j}^{\top}\boldsymbol{\Phi}_{k,j}. \tag{A.7}$$