Design, Dynamics, and Control of Micro Underwater Vehicle Systems for Autonomous Environmental Exploration

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

von
Eugen Solowjow
aus Stepnogorsk

2018

1. Gutachter: Prof. Dr.-Ing. habil. Prof. E.h. Edwin Kreuzer
2. Gutachter: Prof. Dr. rer. nat. Jianwei Zhang
3. Gutachter: Prof. Dr.-Ing. Günter Ackermann

Tag der mündlichen Prüfung: 4. Januar 2018

Design, Dynamics, and Control of Micro Underwater Vehicle Systems for Autonomous Environmental Exploration

Eugen Solowjow

Acknowledgements

The journey that lead to this dissertation has been truly fulfilling and rewarding. I would like to thank everybody who provided support and encouragement. I am deeply grateful to my advisor Prof. Edwin Kreuzer who always supported me and profoundly shaped my academic development. His guidance extended well beyond my thesis research and will hopefully continue for a long time to come. I would also like to thank Prof. Jianwei Zhang and Prof. Günter Ackermann for being on my dissertation committee and Prof. Otto von Estorff for serving as the chair of the committee.

Being part of the Institute of Mechanics and Ocean Engineering provided me with colleagues and friends. I thank Michael Steidl who encouraged me to pursue a Ph.D. at MuM. Axel Hackbarth inspired me for the research direction that turned into this dissertation. I am grateful to Marc-André Pick for countless discussions about research and life. I could always count on Daniel Dücker's support and advice. Daily life at MuM was always very pleasant thanks to my colleagues Ludwig Krumm, Leo Dostal, Christian Radisch, Margret Andrlik, Bärbel Kretzschmar, Norbert Borngräber-Sander, Wolfgang Brennecke, Riza Demir, Merlin Morlock, Svenja Otto, and Robert Seifried.

I am very grateful to all the excellent students I had the honour to supervise and to work with. Representative for more than twenty students, I would like to thank Andreas René Geist, Tobias Johannink and in particular Viktor Rausch for their dedication to research and on the basketball court.

Over the years I had the pleasure to collaborate and interact with many excellent researchers from whom I learned a lot. Many of these collaborations grew into friendships spanning continents. I am very thankful to Alyssa Novelia, Andreas Hansen, Michael Sankur, Hsien-Chung Lin, Austin Buchan, and Wallace Bessa. My friends Arthur Seibel, Viktor Krasmik, and Alex Schimanowski were always available for entertaining coffee

breaks, for which I am very thankful. I would also like to thank Hannes Hatecke for his support, cheerfulness, and for our decade-long friendship.

I cannot thank enough my family who provided the foundation of my support and without whom none of this would have been possible. My parents, Nelli and Alexander, instilled the curiosity and work ethic in me. My uncle Alexander taught me the importance of learning and education. My brother Friedrich provided the very important affection and support.

Contents

List of Figures

List of Tables

List of Important Abbreviations and Symbols

Abbreviations

ADC	analog-to-digital converter
AUV	autonomous underwater vehicle
CRLB	Cramer-Rao lower bound
EM	electro-magnetic
ESC	electronic speed controller
EKF	extended Kalman filter
FSK	frequency shift keying
GRF	Gaussian random fid
GMRF	Gaussian Markov random field
GNSS	Global Navigation Satellite Systems
I^2C	inter-integrated circuit
IMU	inertial measurement unit
LiPo	lithium polymer
MEMS	microelectromechanical system
μAUV	micro autonomous underwater vehicle
PF	particle filter
PI-GMRF	path integral Gaussian Markov random field
POMDP	partially observable Markov decision process
RF	radio frequency
RMS	root mean square
RSS	received signal strength
TDOA	time difference of arrival
WP	waypoint

Functions and Scalars

b	adjustable tuning parameter, cost
c	velocity
d	distance
e	error
f	frequency
k	control gain, timestep
l	line segment
m	mass
p	roll velocity
q	pitch velocity, control cost
r	yaw velocity
s	signal
t	time
u	surge velocity
v	sway velocity
w	heave velocity
x, y, z	Cartesian coordinates
A	reference area
F	force
M	moment
N	prediction horizon length
O	origin
P	sampling points
R	spherical range
S	EM wave power
T	circular period, sampling time
V	value function
α	attenuation constant
ϵ	quaternion element
θ	pitch angle
κ	scaling factor
λ	constant
ρ	fluid density, tuning parameter, hyperbola
σ	standard deviation
τ	time difference of arrival

ϕ	roll angle
ψ	yaw angle
Γ	EM offset factor

Matrices and Vectors

$\boldsymbol{d}$	distance vector
$\boldsymbol{e}$	unit vector
$\boldsymbol{g}$	control vector
$\boldsymbol{p}$	receiver position vector
$\boldsymbol{q}$	orientation vector
$\boldsymbol{r}$	position vector
$\boldsymbol{s}$	reference points
$\boldsymbol{w}$	noise vector, GMRF value vector
$\boldsymbol{z}$	field measurements
$\boldsymbol{A}$	observation matrix
$\boldsymbol{C}$	added Coriolis matrix, measurement covariance
$\boldsymbol{D}$	hydrodynamic damping matrix
$\boldsymbol{F}$	Fisher information matrix
$\boldsymbol{G}$	control transition matrix
$\boldsymbol{I}$	unit matrix
$\boldsymbol{J}$	inertia tensor, Jacobian
$\boldsymbol{K}$	Kalman gain
$\boldsymbol{M}$	mass matrix
$\boldsymbol{Q}$	covariance matrix, precision matrix
$\boldsymbol{R}$	rotation matrix, weight matrix
$\boldsymbol{S}$	transformation matrix, innovation covariance matrix
$\boldsymbol{U}$	control matrix
$\boldsymbol{\epsilon}$	Gaussian control noise
$\boldsymbol{\eta}$	inertial state vector
$\boldsymbol{\kappa}$	innovation vector
$\boldsymbol{\lambda}$	flat output
$\boldsymbol{\mu}$	field mean
$\boldsymbol{\nu}$	relative state vector
$\boldsymbol{\xi}$	state vector
$\boldsymbol{\tau}$	covariance evaluation

$\boldsymbol{\upsilon}$	translational velocity vector, measurement noise
$\boldsymbol{\chi}$	flow field
$\boldsymbol{\omega}$	angular velocity vector
$\boldsymbol{\Theta}$	Euler angles
$\boldsymbol{\Sigma}$	variance

Indices

des	desired
fl	fluid
A	added mass
D	derivative feedback
P	proportional feedback
υ	translational velocity
ϕ	roll angle
ω	angular velocity

Operators and Other Symbols

IFFT	inverse fast Fourier transform
FFT	fast Fourier transform
Var[$\cdot$]	variance
∇	Nabla operator
∂	partial derivative
Δ	difference
J	cost functional
$p(\cdot)$	probability
$\mathbb{E}[\cdot]$	expectation
$\mathcal{F}$	field object
$\mathcal{N}$	normal distribution
$\sim$	distributed as
$\propto$	proportional to
$()^\top$	transposed

Chapter 1 Introduction

Exploration, mapping, and monitoring with autonomous agents, notably in marine environments, has been a prominent topic in the last decade and is expected to gain more importance in the future. In particular, understanding spatio-temporal dynamic processes and fields such as flow conditions, temperature, salinity, or oxygen concentration is at the core of many scientific disciplines and is becoming increasingly important for engineering marine applications.

Recent technological advances in the fields of perception, design, and control have given rise to the use of autonomous underwater vehicles (AUVs) for sampling and analyzing spatio-temporal fields. While today, mostly single, large, and highly specialized AUVs are employed, the future generations of AUVs will act as networked groups operating in dynamic environments. Compared with vehicles which perform solo missions, AUV fleets offer greater operational capabilities and are more efficient. This is particularly the case for adaptive sampling of environmental fields because a multi-agent system can provide a higher resolution in space and time and allows to adapt the strategy of single agents based on the information gathered by the whole group.

Advances in the area of micro aerial robotic systems led to a miniaturization and cost decrease of hardware, such as motor controllers and sensor suits. This enables the development of novel micro AUVs (μAUVs) suitable for multi-vehicle operations. An underwater vehicle is usually termed μAUV if it has a characteristic size of less than 50 cm [79]. Micro AUVs are increasingly becoming interesting for operations in industrial settings. Equipped with appropriate sensors, they allow for autonomous exploration, mapping, and monitoring of spatio-temporal fields in confined environments, as illustrated in Figure 1.1. Furthermore, μAUVs require positioning systems to estimate their own positions, which consists of localization beacons installed prior to operations.

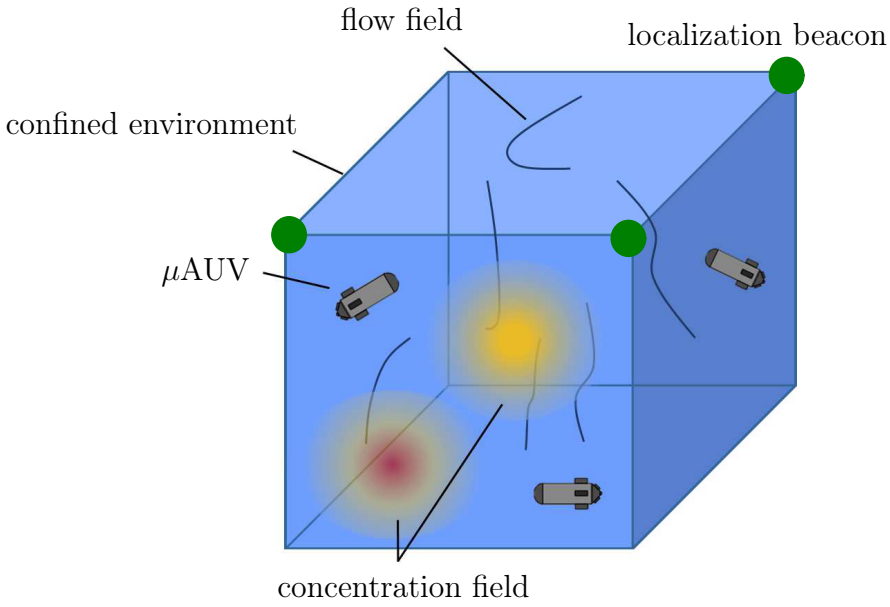


Figure 1.1: Application scenario for a fleet of μAUVs. A confined environment contains a liquid with an underlying concentration field of interest. The μAUVs collect measurements to infer the state of the concentration field while the flow field disturbs the motion of the μAUVs. Localization beacons installed prior to operation allow the micro AUVs to estimate their positions.

Environmental fields of interest include temperature, pH-values, or certain chemical concentrations that are important for the processes. Possible applications are monitoring of storage ponds at nuclear power plants [24] and information gathering in process tanks that are used in chemical engineering as well as in electricity generation and water production [79]. A μAUV based monitoring system could enable improved in-situ analyses and feedback of liquid-based processes offering great potential for increased safety and cost decrease. Further sites for using μAUVs are port basins, where they could replace laborious vessel-based operations for measuring certain concentration levels. The small size and autonomy makes μAUVs attractive for space exploration as well. Data gathering in water on planets and moons, such as Jupiter's moon Europa, are promising future applications of this technology [29].

Deploying μAUVs as a fleet is challenging because efficient vehicle group management by human remote-control is at best extremely difficult, and often just impossible. Therefore, distributed automatic control algorithms have to be deployed to navigate multi-AUV systems instead of human remote-control. Fundamental questions on how a μAUV fleet can be utilized to adaptively sample and autonomously explore spatio-temporal fields are currently in the domain of active research. Addressing these problems requires a formal mathematical framework which considers coordinated vehicle controls and information

gathered from environment through local sensing. The problem differs from control of aerial or ground vehicle groups because of the constraints imposed by aqueous environments. This includes communication, dynamics, and position uncertainty underwater. Exploration and monitoring in confined underwater environments requires μAUVs with sufficient computing and sensing capabilities. Unlike with micro aerial vehicles, the technology and commercialization of μAUV platforms did not yet advance to a point where vehicles would be readily available for purchase. Current full scale AUV technologies offer only limited benefits for μAUVs because crucial sensing devices, such as Doppler Velocity Log sensors, are based on physical principles that do not allow down scaling. Fundamental research on μAUV systems to make industrial applications possible began only recently. This includes mechatronic design but also position and attitude control systems. For a fleet of μAUVs to autonomously explore underwater environments without human intervention, self-localization is a key challenge. Autonomous operations require the agents to know their positions. Based on state-of-the-art technology, this appears to be the most critical bottleneck for real-world applications of μAUVs.

This dissertation addresses challenges associated with exploration and monitoring of confined environments with μAUVs. The presented work contributes to the current main problems consisting of μAUV systems design, self-localization, and tractable information-theoretic control approaches for autonomous behavior.

1.1 Survey of Related Work

This section provides a literature overview on topics related to this dissertation. The survey focuses on exploration and monitoring with AUV groups, μAUV systems, and self-localization.

1.1.1 Exploration and Monitoring with AUV Groups

Exploration of spatio-temporal fields with AUV groups can be linked to information theoretic control [4], where information collected by agents is directly coupled with their motion. This is also called active sensing. The agents observe the environment to perform local control actions and to infer information about the underlying environmental fields.

There has been substantial work in the area of collective motion in general and for exploration purposes in particular. Control approaches for collective explorative motion can be categorized along different dimensions. The approaches are either centralized or decentralized, they follow different strategies on ways to use collected information to compute

control actions, they tackle the inherent suboptimality in the problem differently. Furthermore, the included environmental models differ.

Centralized algorithms for controlling collective motion, for example with model predictive control [12, 25], rely on a central unit that receives all information and provides coordination to all vehicles. These approaches are usually difficult to scale and require computational resources and communication bandwidths which are often not available in underwater use. Hence, a great emphasis lies on research concerned with coordinated control based on local information without an explicit global coordinating unit. Among those, decentralized feedback control laws based on simple dynamical models are of particular interest for two reasons. They show great scalability, which is a prerequisite for engineering large scale systems and they offer the potential for designing systems which resemble the rich behavior observed in biological collectives such as fish schools [67, 5]. However, decentralized approaches are usually more difficult to design and analyze compared to their centralized counterparts.

Exploration and monitoring tasks that have been of interest in recent studies include environmental mapping, gradient climbing, level curve tracking, coverage control of possibly large areas, and extremum seeking. Information gathering problems have by design an optimization flavor because there is a choice for where to take next measurements. Solution approaches for the above tasks usually fall into one of the following four optimization strategies. (1) Myopic gradient based optimization. Here, the next best action is derived without considering a planning horizon [13]. This is usually efficient but prone to local optima. (2) Receding horizon techniques [70]. This allows for optimality within the control horizon, but cannot provide any performance guarantees beyond the control horizon. (3) Dynamic programming. If the exploration problem is phrased as a (partially observable) Markov decision process, it can be tackled with methods such as dynamic programming [55]. However, without domain and state discretization, such approaches are not tractable. (4) Sampling-based methods. These methods have been very successful during the last years [38, 8], but do not guarantee global optimality.

Field exploration approaches can furthermore be classified in two types; (1) control methods which rely on environmental models and (2) the ones which do not. Extremum seeking [73] and coverage-control [39, 75] are among the most common applications that do not require environmental models. Various other control methods such as concentration field mapping require the incorporation of environmental models within the control law. Optimal coordination of a group for the sake of active estimation of a spatio-temporal field is a provably computationaly infeasible problem [23], i. e. any algorithm scales exponentially in time. All approaches for navigating AUVs for information gathering can only be suboptimal, either by relaxing the optimality condition, restricting the optimization problem

to a subset of possible paths [46], or parameterizing the formations and fields by a finite set of parameters [10].

While environmental fields can be naturally expressed with physics-based models [25], such approaches require solving partial differential equations (e. g. the Navier-Stokes equations) within the control formulation on-line. This is computationally demanding and requires information that might not be available to the AUV group, such as the boundary conditions of the environmental fields.

Very recently, techniques based on stochastic field modeling gained attention. These techniques seem to be suitable for computationally constrained AUV teams because they are computationally less involved than physics-based models and they do not require the type of a priori knowledge that physics-based models do. An example of a stochastic regression technique for estimating environmental fields is Gaussian process regression. Gaussian process regression, also called Kriging method, was originally developed in geostatistics [35]. The unknown field is assumed to be a Gaussian random field (GRF), which can be interpreted as a distribution over a function space. The GRF is completely defined by its mean and covariance function. The goal of Gaussian process regression is to compute a posterior field distribution at arbitrary locations given the measurements. Given a prior, which is often a distribution over all possible realizations, observations update the distribution of functions and render some of them more likely than others. Krause [34] provides many important theoretical results on learning environmental fields which are modelled as GRFs. A key result of his work is that near-optimal sensor placement can be achieved for known stationary covariance functions with submodular information-theoretic metrics using an a-priori design. However, this does not hold true for sequential sampling where agents have to obey dynamics or are disturbed by unknown flow fields.

The feasibility of using Gaussian process regression to learn an environmental scalar field is demonstrated in [85]. In their study Zhang and Sukhatme deployed static buoys and a robotic surface boat to sample and reconstruct the temperature field of a lake surface. Xu et al. [82] consider a mobile sensor network for learning spatio-temporal GRFs via Gaussian process regression based on truncated observations. In their subsequent monograph, Xu et al. [84] provide an extensive collection of techniques for modeling environmental fields with GRFs and myopic adaptive exploration approaches. Hollinger and Sukhatme [28] introduce sampling-based navigation strategies for information gathering which have the advantages of being suitable for continuous domains and scale well with the number of agents. They focus on planning directly in the space of the information quality metric and develop a sampling based exploration algorithm. Recent work by Ma et al. [50] proposes a path planner that exploits GRFs to find most informative observation waypoints. Their algorithm solves a Markov decision process, but does not consider AUV dynamics which

is usually not necessary for path planning in oceans, but becomes important in confined underwater environments. Marino and Antonelli [51] present a sampling strategy for AUVs that is based on GRFs. The proposed AUV control strategy divides the spatial domain with Voronoi tessellations, and for each tessellation a GRF is constructed. The controller follows a myopic strategy to decide on future measurement points.

Recent efforts incorporate Gaussian Markov random fields (GMRFs) to approximate GRFs [69]. They possess the Markov property in space and time and scale better with increasing number of measurements than conventional GRFs. This property is advantageous for resource constrained, distributed mobile robots. Xu et al. [83] present a framework where a scalar field is modeled as a GMRF with uncertain parameters. Research on representing spatiotemporal fields by GRFs is expected to become more popular because of the computational advantages over physics-based models. Boundary conditions are not required and the availability of uncertainty measures promises great potential for navigating AUV-groups in monitoring and exploration missions. Currently, ongoing work on GMRF representations considers nonstationarity of the covariance function. This allows more accurate modeling because environmental fields have in general nonconstant decorrelation scales [16].

1.1.2 Micro Underwater Vehicles

A key step for transitioning theoretical results to real-world applications lies in hardware implementation and experimental validation with μAUVs. Disturbance rejection, sensor noise, process noise, communication limits, and model uncertainties are should be studied in theory and experiments. So far, experimental validations with μAUVs were mostly performed in the context of formation control. This type of control usually requires information about the relative positions of the vehicles to each other. Monitoring and exploring, however, require global position information, which is more difficult to obtain. The corresponding self-localization problem is challenging in confined underwater environments. This will be further explained in the next subsection.

Since μAUVs are not readily available for purchase yet, each research project requires the design of a platform, which is often customized for target applications. One of the first experimental test beds for μAUV cooperative formation control is reported by Klein et al. [33]. The authors developed and deployed fin-actuated μAUVs and provided results for formation control based on the Kuramoto model. Napora and Paley [56] present the implementation and experimental results of a synchronization algorithm for an AUV group, which is supposed to perform stabilized parallel and circular motions. The AUVs in the study are modified commercially available remote controlled vehicles equipped

with custom electronics. The size of these vehicles exceeds one meter. A μAUV specifically designed for swarm experiments is introduced in [54]. Swarm algorithms distinguish themselves through their simplicity, and the computing hardware is therefore usually limited. Other μAUV designs include a spherical μAUV for applications in wireless sensor networks [47], the AVEXIS platforms for monitoring storage ponds at nuclear facilities [24] and the MONSUN vehicle [60]. In this dissertation, the HippoCampus platform [26] is covered. It exploits recent advancements in the area of micro aerial vehicle systems.

1.1.3 Self-Localization in Confined Underwater Environments

Self-localization approaches for μAUVs pose great technological challenges. State-of-the-art systems do not provide satisfactory and reliable performance for μAUV operations yet.

Outdoor robots usually rely on Global Navigation Satellite Systems (GNSS) for localization, whereas indoor robots, such as aerial vehicles, often utilize camera systems for localization. Submerged robots, however, have difficulties using aforementioned technologies. It is impossible to use GNSS for localizing submerged robots because electro-magnetic signals at frequencies above 1 GHz are quickly attenuated in water. Vision-based systems are also not feasible because image recognition is difficult underwater. Furthermore, in order to obtain global positions, image capturing and processing has to be performed off-board, and the positions have to be sent to the robots. Off-board localization introduces latencies and degrades closed-loop controller performance. Since underwater communication is limited, sending positions to the robots can be unreliable. Therefore, the ability to perform on-board self localization is highly desirable for μAUVs. Furthermore, a localization framework that is not affected by the number of fleet members is required for large μAUV fleets. This has to be achieved via passive localization approaches. In passive localization, the receiver observes an ambient reference signal without actively transmitting.

Passive self-localization systems rely on a signal that is transmitted by beacons. The beacons are installed prior to operations, and their positions are known to the receivers. The receiver position is obtained in relation to the beacons by either determining the range to the beacons (spherical positioning), or by determining the difference between the ranges (hyperbolic positioning). Acoustic signals can be used as ambient reference signals for passive localization in confined underwater environments, such as filled tanks. However, acoustic signals multi-path propagation and reflections pose serious challenges to acoustic approaches in confined underwater environments as opposed to openwaters, for example oceans or lakes. Reverberation time of acoustic signals in confined underwater environments can be much longer than the original signal. This is a problem because the

signal has to be uniquely detected for localization signal processing. Another challenge for deploying acoustic systems lies in the limited capabilities of μAUVs in terms of clock synchronization. Without proper clock synchronization, time of flights of acoustic signals cannot be determined accurately. An acoustic spherical positioning system with channel switching that does not require clock synchronization is suggested in [11]. However, only simulations are presented without degrading effects.

Recently, a series of papers [63, 64, 65, 66, 36] introduced a novel approach for determining the range in underwater localization based on the attenuation of electro-magnetic carrier waves. Furthermore, the authors show the feasibility of the system for accurate localization in confined underwater environments. In [63], Park et al. derive the relationship between range and electro-magnetic wave attenuation from the Maxwell equations and the Friis transmission formula. The relationship is used in [64] to demonstrate self-localization in the horizontal plane based on the attenuation of electro-magnetic carrier waves. Thorough analyses for issues occurring when the system is deployed in three-dimensional space are provided in [66, 36]. The benefits of the electro-magnetic approach is that multi-path propagation, reflections, and reverberation are much weaker for electro-magnetic waves than for acoustic waves in water. The localization accuracies that are reported in [63, 64, 65, 66, 36] are within a millimeter range at a 1000 Hz update rate. All results reported in [63, 64, 65, 66, 36] require a full-fledged spectrum analyzer for computing an ultra-high frequency (UHF) power density spectrum. Unfortunately, such a spectrum analyzer is not deployable on-board for μAUVs, because it is too bulky, heavy, and costly. Due to inferior UHF signal processing capabilities, an embedded system for μAUVs will most likely have a degraded localization accuracy and update rate. However, even if the accuracy reduces to a centimeter range and the update rate to 10 Hz, such an embedded localization system would fulfill localization specifications for most autonomous operations in confined underwater environments. It would be an improvement over the state-of-the-art that would render μAUV operations in confined underwater environments possible.

1.2 Contributions and Outline of Dissertation

This dissertation contributes to three areas associated with exploration and monitoring of confined underwater environments with μAUV systems. First, a novel μAUV system called HippoCampus is presented. Second, two solution approaches for the underwater self-localization problem in confined underwater environments are introduced. Third, adaptive and autonomous environmental field exploration and monitoring is addressed. In the following, a summary of key sections of this dissertation is provided.

Micro Underwater Vehicle HippoCampus

The HippoCampus vehicle is a low-cost, open-sourced μAUV designed for applications in confined environments. It is highly manoeuvrable due to its quadrotor design. In Chapter 2, the HippoCampus system is presented in detail. Its mechatronic architecture and a thorough dynamical analysis are covered including low-level feedback controls for trajectory and waypoint tracking. Furthermore, the section introduces a path following controller for μAUVs, which is based on the receding horizon sliding control framework.

Contents of this chapter are partly published in

- 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.
- T. Johannink, E. Kreuzer, and E. Solowjow. Sealing of Machine Parts and Modules Manufactured with Desktop 3D Printers (in German). *Konstruktion*, 1/2: 63–67, 2017.

Acoustic Hyperbolic and Electro-Magnetic Spherical Localization for μAUVs

Self-localization systems for μAUVs in confined underwater environments represent very challenging problems due to their strict requirements in size and cost. Chapter 3 introduces two self-localization approaches that are suitable for μAUVs.

The first approach is a hyperbolic acoustic system for underwater self-localization. It is a passive localization system with one-way signal transmission. Time differences of arrival (TDOAs) between the emitted acoustic signals are estimated by the receiver via cross-correlation. These TDOAs are fed to a Bayesian filter to estimate the global position of the receiver. Experimental results in air and water demonstrate that acoustic one-way localization is possible with off-the-shelf hardware.

As the second approach, an embedded system for spherical electro-magnetic localization is presented. It extends recent results for estimating ranges based on attenuation of electromagnetic waves. Previous work relied on a full-fledged signal analyzer to compute the spectrum of a high frequency signal. This section demonstrates that the required signal processing can be performed on a compact Digital Video Broadcasting - Terrestrial dongle.

Contents of this chapter are partly published in

- 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.

- D.-A. Duecker, A.R. Geist, M. Hengeler, E. Kreuzer, M.-A. Pick, V. Rausch, E. Solowjow. Embedded Spherical Localization for Micro Underwater Vehicles based on Attenuation of Electro-Magnetic Carrier Signals. *Sensors*, 17(5): 959-981, 2017.

Path Integral - Gaussian Markov Random Field Controller

Chapter 4 presents an information theoretic control approach for exploration and monitoring of underwater environments with μAUVs. The controller is based on core ideas from path integral control and inference with Gaussian Markov Random Fields (GMRFs). Both domains are combined in a receding horizon scheme to the path integral (PI) -GMRF controller for environmental monitoring. The control problem is formulated within the stochastic optimal control domain, and a solution is stated as a path integral. In order to close the control theoretic loop, a belief representation of the environment is maintained that is expressed with GMRFs. The PI-GMRF controller is developed for resource constrained computing hardware and is optimized towards online feasibility.

Contents of this chapter are published in

- E. Kreuzer, E. Solowjow. Learning Environmental Fields with Micro Underwater Vehicles: A Path Integral – Gaussian Markov Random Field Approach. *Autonomous Robots*, 42(4): 761-780, 2018.

Chapter 2 Micro Underwater Vehicle HippoCampus

This chapter presents the HippoCampus μAUV, a robotic platform designed for exploration and monitoring in confined underwater environments. The vehicle development is motivated by Chapter 1. The design of HippoCampus follows a mechatronic design methodology, for which sensing, computing, and actuation are closely interlinked. Agility is critical for operating in confined environments, which is reflected in the design. The equations of motion of HippoCampus are derived and dynamic properties are analyzed for low-level control concepts. Based on the differential flatness property of the HippoCampus dynamics, control laws for trajectory and waypoint tracking are introduced. A receding horizon control method for path following is presented in the last section of this chapter.

2.1 System Architecture

This section presents the mechatronic design of the HippoCampus μAUV. HippoCampus can be deployed in confined underwater environments and in open waters such as lakes or port basins. The HippoCampus μAUV is designed for high vehicular autonomy and possesses sufficient sensing and computing resources. High maneuverability is a crucial attribute for space limited underwater sites, and actuation has to reflect that. The system follows design requirements of being low cost, open source and uses off-the-shelf components. This allows rapid reproduction of HippoCampus, which is important for a system that is intended for multi-vehicle operations. The initial application environments are water tanks with water depths of up to 10 m. The current design presented in this section is based on theoretical considerations, experimental results, and lessons learned by going through multiple design iterations. Figure 2.1 portrays different versions of HippoCampus as CAD models and photographs. The top row shows the first design iterations while the bottom row shows the current version.

2.1.1 Mechanical Structure and Components

HippoCampus consists of two cylindrical tubes made out of acrylic glass which are connected to a 3D-printed base unit. It is enclosed by two covers, which house sensors and antennas. The default configuration includes a pressure and a temperature sensor. Front and rear covers can be equipped with additional sensors. HippoCampus has two antennas. One is intended for radio communication and the other for RF localization. The acrylic enclosure hosts electronic components, a battery, and the computing boards. All 3D-printed parts are produced with the stereolithography method which allows printing nonporous parts that are waterproof without any additional processing. The more commonly used fused deposition modeling method results in leakage, due to micro pores. The structure provides rigidity and was tested at a relative pressure of 1.2 bar (equivalent to 12 m water depth). The current design should withstand relative pressures of up to 5 bar (equivalent to 50 m water depth), although this was not tested yet. Propulsion is achieved with custom 3D-printed thrusters which are attached to the base unit. Thrusters consist of integrated brushless DC motors with counter-rotating propellers. The four-thruster design is inspired by multi-rotor micro aerial vehicles and offers distinct advantages. The vehicle has great maneuverability and sufficient actuator bandwidth. Furthermore, turning rates are independent of forward motion, which is an advantage to conventional underwater vehicle designs with a single propeller and rudders. Also, no mechanical linkages and no dynamic sealing are required in the system, which vastly simplifies manufacturing and assembly. The disadvantage of the design is worse energy efficiency because of additional motors. HippoCampus does not have a dive tank and is by design neutrally net buoyant, whereby the center of volume approximately coincides with the center of mass. The total weight of the vehicle is ca. 1,300 g, and it has a length of ca. 380 mm.

2.1.2 Computational Resources

The on-board computational resources are comprised of two single-board computers that offer all required functionality for autonomous operations. The opensource Pixhawk controller board [52] is the main processing unit for low-level computations. The Pixhawk board is popular for mobile robotic applications because it is compact and widely available. It consists of a 168 MHz Cortex M4F CPU with 256 KB RAM and 2 MB flash memory. The Pixhawk unit coordinates communication, sensing, attitude estimation, and low-level controls. Data can be logged to a microSD-card. The second single-board computer is a Raspberry Pi Zero that runs at 1 GHz with 512 MB of RAM. The Raspberry Pi Zero drives the Digital Video Broadcasting - Terrestrial dongle for RF self-localization and

Figure 2.1: Different versions of HippoCampus μAUV. Top left: An early concept drawing. Top right: The first prototype. Bottom left: CAD model of the final version. Bottom right: Photo of the final version.

performs the computationally intensive signal processing associated with localization. It can in principle also be used for additional high-level control algorithms, that are not real-time critical.

2.1.3 Sensors and Electronics

The sensor suite consists of the on-board Pixhawk sensors and includes a 16 bit three-axis gyroscope as well as a combined 14 bit three-axis accelerometer and magnetometer for vehicle attitude estimation. A pressure sensor for depth control and a temperature sensor for environmental monitoring tasks are installed. Additional sensors can be added through I^2C, SPI or ADC interfaces. Off-the-shelf electronic speed controllers (ESCs) rated at 15 Amps drive the brushless DC motors. The vehicle is powered by a lithium polymer (LiPo) battery, which has a capacity of 2,200 mAh. The operational time is approximately 60 minutes. The communication link to the ground station and to other vehicles is established through radio at 433 MHz with an adjustable output power of up to 100 mW and frequency shift keying (FSK) modulation. Due to attenuation of electromagnetic waves in water, only short range communication of ca. 6 m can be achieved with radio at 433 MHz. Since the system is developed for small- and mid-sized underwater environments, this communication range is sufficient. A reliable communication link

to HippoCampus can be maintained at any location in a $4\,\text{m} \times 3\,\text{m} \times 1.5\,\text{m}$ tank with an output power of 10 mW and a baud-rate of 56,700 Bd. Reducing the baud-rate and increasing the output power increases the communication range for operations in larger tanks. A Digital Video Broadcasting - Terrestrial dongle can be used to process RF signals on-board for self-localization. It is driven via USB 2.0 by for example a Raspberry Pi Zero board. Figure 2.2 provides a high-level overview of the sensors and electronic components of the vehicle.

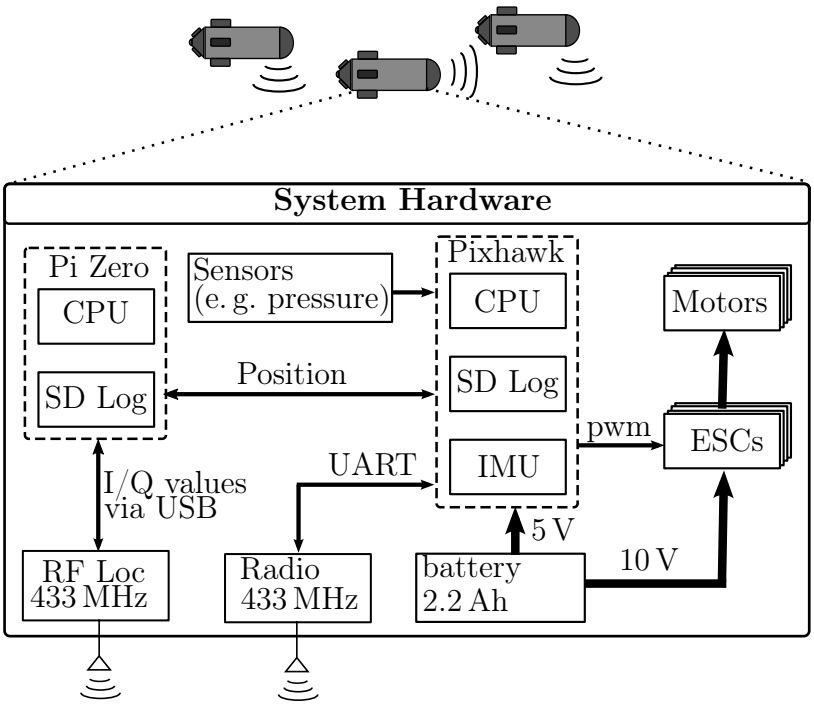


Figure 2.2: Electronic components of HippoCampus.

2.1.4 Software Architecture

The software architecture of HippoCampus consists of several abstraction layers and extends the PX4 project [52] to μAUVs. The PX4 stack is a node-based multi-threaded open source framework developed along the Pixhawk controller board for deeply embedded mobile robotics applications [52]. The HippoCampus firmware includes customized software modules specific to the HippoCampus platform and to underwater robots in general. However, at the same time, it utilizes common elements among mobile robots such as inertial state estimation. It is crucial to utilize existing functionality in robot software development, because modern mobile robotics firmware can easily contain hundreds of thousands of code lines. The Pixhawk controller board runs NuttX, a real-time operating system that provides a POSIX-style environment. The low-level drivers are implemented

in NuttX. On top of NuttX, the PX4 middleware provides device drivers and an object request broker (uORB) for asynchronous communication between the individual threads. The Pixhawk board communicates with the single board computer via a serial link.

2.2 Dynamics

This section presents the equations of motion for the HippoCampus vehicle. The equations of motion allow to analyze the design and to develop low-level controls, for example for trajectory tracking.

Let $\mathcal{I} := \{O_\mathcal{I},\ x_\mathcal{I},\ y_\mathcal{I},\ z_\mathcal{I}\}$ be an inertial coordinate system and $\mathcal{B} := \{O_\mathcal{B},\ x_\mathcal{B},\ y_\mathcal{B},\ z_\mathcal{B}\}$ the body fixed coordinate system. The set of unit vectors $(\boldsymbol{e}_x^\mathcal{B},\ \boldsymbol{e}_y^\mathcal{B},\ \boldsymbol{e}_z^\mathcal{B})$ represents the axes of $\mathcal{B}$. The origin of $\mathcal{B}$ coincides with the center of gravity of HippoCampus. The roll axis $x_\mathcal{B}$ is along the longitudinal direction of the vehicle, the pitch axis $y_\mathcal{B}$ points in starboard direction, and $z_\mathcal{B}$ is oriented downward as illustrated in Fig. 2.3.

The state of HippoCampus is described by $[\boldsymbol{\nu}^\top\ \boldsymbol{\eta}^\top]^\top$. Thereby, $\boldsymbol{\eta} = [\boldsymbol{r}^\top\ \ \boldsymbol{q}^\top \,|\, \boldsymbol{\Theta}_{\mathcal{IB}}^\top]^\top$ describes the vehicle position and orientation. The vehicle position in the inertial coordinate system is $\boldsymbol{r} = [x\ y\ z]^\top$. The orientation can be described by either a unit quaternion $\boldsymbol{q} = [\epsilon_0\ \epsilon_1\ \epsilon_2\ \epsilon_3]^\top$ or by z-y-x Euler angles $\boldsymbol{\Theta}_{\mathcal{IB}} = [\phi\ \theta\ \psi]^\top$. While the quaternion representation avoids Gimbal lock, Euler angles allow to analyze single degrees of freedom (e. g. the roll motion) more conveniently. Both rotation representations will be used throughout the remainder of this chapter. The vector $\boldsymbol{\nu} = [\boldsymbol{v}^\top\ \boldsymbol{\omega}^\top]^\top$ describes the translational velocities $\boldsymbol{v} = [u\ v\ w]^\top$ and the angular velocities $\boldsymbol{\omega} = [p\ q\ r]^\top$ in the body fixed coordinate system $\mathcal{B}$.

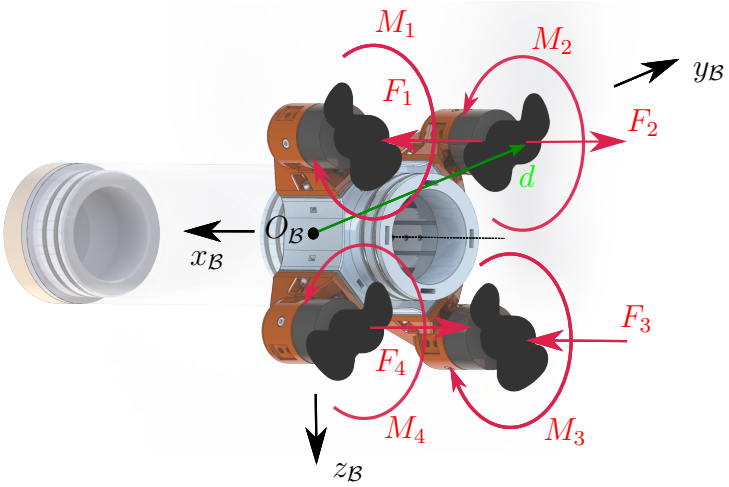


Figure 2.3: Forces and moments due to propulsion. All propellers rotate clockwise.

HippoCampus has four counter-rotating propellers. The i-th propeller's position relative to the center of mass is defined by the position vector $\boldsymbol{d}_i^{\mathcal{B}}$. The i-th propeller spins with the rotational speed ω_i and produces the thrust $\kappa_\mathrm{t}\omega_i^2$ in $x_\mathcal{B}$-direction and a moment $\kappa_\mathrm{m}\omega_i^2$ around the $x_\mathcal{B}$-axis. The scaling factors κ_t and κ_m depend on propeller geometry, the density of the fluid, and heuristic parameters.

The motor dynamics are significantly faster than the rigid body dynamics of the vehicle. We assume therefore that the desired thrust and the desired body torques can be instantaneously achieved. The controllable input $\boldsymbol{u}^{\mathcal{B}}$ to the vehicle system can be written as

$$\boldsymbol{u}^{\mathcal{B}} = \begin{bmatrix} u_1^{\mathcal{B}} \\ 0 \\ 0 \\ u_2^{\mathcal{B}} \\ u_3^{\mathcal{B}} \\ u_4^{\mathcal{B}} \end{bmatrix} = \sum_{i=1}^{4} \begin{bmatrix} \kappa_\mathrm{t}\omega_i^2 \\ 0 \\ 0 \\ \kappa_\mathrm{m}\omega_i^2 \\ -\boldsymbol{d}_i\, \boldsymbol{e}_z^{\mathcal{B}}\, \kappa_\mathrm{t}\omega_i^2 \\ \boldsymbol{d}_i\, \boldsymbol{e}_y^{\mathcal{B}}\, \kappa_\mathrm{t}\omega_i^2 \end{bmatrix} = \sum_{i=1}^{4} \begin{bmatrix} \boldsymbol{F}_i\, \boldsymbol{e}_x^{\mathcal{B}} \\ 0 \\ 0 \\ \boldsymbol{M}_i\, \boldsymbol{e}_x^{\mathcal{B}} \\ -(\boldsymbol{d}_i \times \boldsymbol{F}_i)\, \boldsymbol{e}_z^{\mathcal{B}} \\ (\boldsymbol{d}_i \times \boldsymbol{F}_i)\, \boldsymbol{e}_y^{\mathcal{B}} \end{bmatrix} . \tag{2.1}$$

The components of $\boldsymbol{u}^{\mathcal{B}}$ are the the collective thrust $u_1^{\mathcal{B}}$ and the body torques $u_2^{\mathcal{B}}$, $u_3^{\mathcal{B}}$, and $u_4^{\mathcal{B}}$ associated with the degrees of freedom roll, pitch, and yaw, respectively.

The equations of motion for HippoCampus read

$$\dot{\boldsymbol{\eta}} = \boldsymbol{S}(\boldsymbol{\eta})\boldsymbol{\nu}\,, \tag{2.2}$$

$$(\boldsymbol{M}_\mathrm{A} + m\boldsymbol{I})\dot{\boldsymbol{v}} = -\boldsymbol{D}_\mathrm{v}\boldsymbol{v} + \boldsymbol{v} \times \boldsymbol{M}_\mathrm{A}\boldsymbol{\omega} + u_1\boldsymbol{e}_x^{\mathcal{B}}\,, \tag{2.3}$$

$$(\boldsymbol{J}_\mathrm{A} + \boldsymbol{J})\dot{\boldsymbol{\omega}} = -\boldsymbol{D}_\omega\boldsymbol{\omega} - \boldsymbol{\omega} \times (\boldsymbol{J}_\mathrm{A} + \boldsymbol{J})\boldsymbol{\omega} + \boldsymbol{v} \times \boldsymbol{M}_\mathrm{A}\boldsymbol{\omega} + [u_2^{\mathcal{B}}\ u_3^{\mathcal{B}}\ u_4^{\mathcal{B}}]^\top\,, \tag{2.4}$$

where $\boldsymbol{S}$ is the transformation matrix between body fixed and inertial reference frames. The transformation matrix $\boldsymbol{S}$ can be partitioned with a rotation matrix $\boldsymbol{R}_\mathcal{B}^\mathcal{I}$ that transforms translational velocities and a matrix $\boldsymbol{T}$ that transforms angular velocities. They can either be expressed in terms of Euler angles $\boldsymbol{\Theta}_{\mathcal{IB}}$ or in terms of quaternions $\boldsymbol{q}$:

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{R}_\mathcal{B}^\mathcal{I}(\boldsymbol{\Theta}_{\mathcal{IB}}) \mid \boldsymbol{R}_\mathcal{B}^\mathcal{I}(\boldsymbol{q}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}_\Theta(\boldsymbol{\Theta}_{\mathcal{IB}}) \mid \boldsymbol{T}_\mathrm{q}(\boldsymbol{q}) \end{bmatrix} , \tag{2.5}$$

with the rotation matrices

$$\boldsymbol{R}_\mathcal{B}^\mathcal{I}(\boldsymbol{\Theta}_{\mathcal{IB}}) = \begin{bmatrix} \mathrm{c}\psi\,\mathrm{c}\theta & -\mathrm{s}\psi\,\mathrm{c}\phi + \mathrm{c}\psi\,\mathrm{s}\theta\,\mathrm{s}\phi & \mathrm{s}\psi\,\mathrm{s}\phi + \mathrm{c}\psi\,\mathrm{c}\phi\,\mathrm{s}\theta \\ \mathrm{s}\psi\,\mathrm{c}\theta & \mathrm{c}\psi\,\mathrm{c}\phi + \mathrm{s}\psi\,\mathrm{s}\theta\,\mathrm{s}\phi & -\mathrm{c}\psi\,\mathrm{s}\phi + \mathrm{s}\psi\,\mathrm{c}\phi\,\mathrm{s}\theta \\ -\mathrm{s}\theta & \mathrm{c}\theta\,\mathrm{s}\theta & \mathrm{c}\theta\,\mathrm{s}\theta \end{bmatrix} \tag{2.6}$$

or

$$\boldsymbol{R}_q(\boldsymbol{q}) = \begin{bmatrix} 1-2(\epsilon_2^2+\epsilon_3^2) & 2(\epsilon_1\epsilon_2 - \epsilon_3\epsilon_0) & 2(\epsilon_1\epsilon_3 + \epsilon_2\epsilon_0) \\ 2(\epsilon_1\epsilon_2 + \epsilon_3\epsilon_0) & 1-2(\epsilon_1^2+\epsilon_3^2) & 2(\epsilon_2\epsilon_3 - \epsilon_1\epsilon_0) \\ 2(\epsilon_1\epsilon_3 - \epsilon_2\epsilon_0) & 2(\epsilon_2\epsilon_3 + \epsilon_1\epsilon_0) & 1-2(\epsilon_1^2+\epsilon_2^2) \end{bmatrix} , \tag{2.7}$$

as well as

$$\boldsymbol{T}_{\Theta}(\boldsymbol{\Theta}_{\mathcal{IB}}) = \begin{bmatrix} 1 & \mathrm{s}\phi\,\mathrm{t}\theta & \mathrm{c}\phi\,\mathrm{t}\theta \\ 0 & \mathrm{c}\phi & -\mathrm{s}\phi \\ 0 & \frac{\mathrm{s}\phi}{\mathrm{c}\theta} & \frac{\mathrm{c}\phi}{\mathrm{c}\theta} \end{bmatrix}, \tag{2.8}$$

or

$$\boldsymbol{T}_{\mathrm{q}}(\boldsymbol{q}) = \frac{1}{2} \begin{bmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \epsilon_0 & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \epsilon_0 & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \epsilon_0 \end{bmatrix}. \tag{2.9}$$

The abbreviations $\mathrm{s}(\cdot)$, $\mathrm{c}(\cdot)$, and $\mathrm{t}(\cdot)$ represent the trigonometric functions $\sin(\cdot)$, $\cos(\cdot)$, and $\tan(\cdot)$, respectively.

In (2.3) and (2.4), m is the vehicle mass, $\boldsymbol{J}$ is the rigid body inertia tensor with respect to the body fixed frame, $\boldsymbol{M}_{\mathrm{A}}$ and $\boldsymbol{J}_{\mathrm{A}}$ are the hydrodynamic added mass matrices for translation and rotation. Damping is caused by the surrounding fluid, and the matrices $\boldsymbol{D}_{\mathrm{v}}$ and $\boldsymbol{D}_{\omega}$ are the hydrodynamic damping matrices for translation and rotation, respectively. The cross coupling term $\boldsymbol{v} \times \boldsymbol{M}_{\mathrm{A}}\boldsymbol{\omega}$ represents the Coriolis effects of the hydrodynamic added mass. In order to achieve high manoeuvrability, the vehicle is designed such that its hydrostatic equilibrium is maintained for any orientation. Also, the hydrostatic loads and the weight cancel each other by design. Furthermore, the center of gravity coincides with the center of volume and all moments due to buoyancy and weight vanish.

There are different approaches of varying complexity and degree of fidelity for modeling the hydrodynamic added mass and damping. Exact parameter values can be identified either in experiments or with CFD simulations. For very simple geometries, such as cylinders, analytical values can be obtained. The hydrodynamic modeling of HippoCampus is based on the assumptions that are introduced by Fossen [14]. If underwater vehicles are not wave affected, the hydrodynamic added mass can be assumed to be frequency independent. The added mass structure can be assumed diagonal in structure if the vehicle has approximately three planes of symmetry such as HippoCampus. In practice, these approximations deliver very good results because the contributions from off-diagonal elements of the added mass matrix are usually small and hence neglectable. Submerged underwater vehicles below the wave affected zone are not subject to hydrodynamic potential damping, but only to viscous damping [14, p. 112]. Since HippoCampus operates at low velocities of approximately 1 m/s, linearized viscous damping is assumed, with diagonal matrix structures for $\boldsymbol{D}_{\mathrm{v}}$ and $\boldsymbol{D}_{\omega}$. The approximation of damping element d_{ii} is obtained with

$$d_{ii} = \rho_{\mathrm{fl}} C_{\mathrm{D}} A_{\mathrm{ref}}\,, \tag{2.10}$$

where ρ_{fl} is the density of the surrounding fluid, C_{D} is the drag coefficient and A_{ref} is the reference area.

HippoCampus is an underactuated system, because four thrust inputs drive six degrees of freedom. It is apparent from Fig. 2.1 that sway and heave are the underactuated degrees of freedom. HippoCampus is a "flying vehicle" whose lateral velocities are in general small and the encountered flow direction due to self-propulsion is along the longitudinal vehicle axis.

2.3 Geometric Trajectory Control

This section introduces a trajectory tracking controller that allows HippoCampus to track prescribed trajectories in space. The controller is also capable of waypoint tracking. Tracking trajectories and waypoints is a basic operational building block for autonomous behavior of mobile robots. The introduced control system is modular and can serve as the inner loop of high-level controllers that generate desired trajectories or waypoints.

Geometric tracking control has become prominent for quadrotor micro aerial vehicles and has seen varios applications in recent work [42, 53, 49]. Geometric control shows enhanced performance compared with linear control systems and nonlinear control approaches that operate on Euclidean spaces. In general, geometric control theory is not based on the assumption that system states evolve on Euclidean spaces. Instead, it directly considers the influence of the state space geometry [41]. In the case of trajectory tracking for mobile robots, geometric control approaches operate directly on the special orthogonal group instead of angle parametrizations such as the Euler angle representation.

In the following, control strategies presented in [42, 53] are extended to the HippoCampus platform. The control system is designed for the limited sensing capabilities of μAUVs. Velocity measurements can only be obtained through differentiation of position information, which is extremely noisy. Overcoming this limitation for real-world performance and practicality is an important objective of the presented control design procedure.

2.3.1 Differential Flatness

The differential flatness property of a system allows to express all system states and inputs as algebraic functions of flat outputs. The design of HippoCampus renders differentially flat dynamics if two assumptions are imposed. First, neglect the hydrodynamic added mass cross-coupling term $\boldsymbol{v} \times \boldsymbol{M}_{\text{A}}\boldsymbol{\omega}$ in the equation of motion for translation (2.3). This is justified by the second order nature of the added mass cross-coupling term. Second,

set $\boldsymbol{M}_\mathrm{A} = m_\mathrm{A}\boldsymbol{I}$ as well as $\boldsymbol{D}_\mathrm{v} = d_\mathrm{v}\boldsymbol{I}$. The HippoCampus design is optimized towards approximately fulfilling the above assumptions on translational added mass $\boldsymbol{M}_\mathrm{A}$ and damping $\boldsymbol{D}_\mathrm{v}$. However, a trade-off arises in practice because a small added mass in surge direction benefits vehicle agility and energy efficiency. Therefore, a compromise had to be found for the final HippoCampus design. Interestingly, the cross-coupling term $\boldsymbol{v} \times \boldsymbol{M}_\mathrm{A}\boldsymbol{\omega}$, the added inertia $\boldsymbol{J}_\mathrm{A}$, and the damping $\boldsymbol{D}_\omega$ in the equation of motion for rotation (2.4) do not influence the differential flatness property as will be shown. While the assumptions may appear restrictive, adjusting the vehicle design to approximately fulfill them greatly benefits controller design. It can be shown that

$$\boldsymbol{\lambda} = [x\ y\ z\ \phi]^\top = [\boldsymbol{r}^\top\ \phi]^\top \tag{2.11}$$

are flat outputs of the system dynamics that are subject to the two assumptions. This is a nontrivial and desirable result. It implies that smooth curves in $\mathbb{R}^3$ with an additional specified roll orientation are a sufficient reference information for a trajectory tracking controller without the need to specify any of the other 12 vehicle states.

In order to show the differential flatness property, all states and inputs of the HippoCampus vehicle are expressed as algebraic functions of the flat outputs (2.11) and their time derivatives.

Position coordinates $\boldsymbol{x, y, z}$ and roll angle $\boldsymbol{\phi}$: The position coordinates x, y, z and roll angle ϕ are by definition flat outputs, see (2.11).

Rotation matrix $\boldsymbol{R}_\mathcal{I}^\mathcal{B}$: Stating the rotation matrix $\boldsymbol{R}_\mathcal{I}^\mathcal{B}$ in terms of the flat outputs x, y, z, ϕ is equivalent to showing that the Euler angles can be expressed in terms of the flat outputs. First, the equation of motion for translation is rewritten in terms of global coordinates under the assumption that the hydrodynamic cross-coupling term can be neglected:

$$\boldsymbol{R}_\mathcal{I}^\mathcal{B}(\boldsymbol{M}_\mathrm{A} + m\boldsymbol{I})(\boldsymbol{R}_\mathcal{I}^\mathcal{B})^\top\ddot{\boldsymbol{r}} = \boldsymbol{R}_\mathcal{I}^\mathcal{B}\boldsymbol{D}_\mathrm{v}(\boldsymbol{R}_\mathcal{I}^\mathcal{B})^\top\dot{\boldsymbol{r}} + u_1\boldsymbol{e}_x^\mathcal{B}\,. \tag{2.12}$$

The key inside to note is that, due to the assumptions $\boldsymbol{M}_\mathrm{A} = m_\mathrm{A}\boldsymbol{I}$ and $\boldsymbol{D}_\mathrm{v} = d_\mathrm{v}\boldsymbol{I}$, (2.12) can be simplified to

$$(m_\mathrm{A} + m)\ddot{\boldsymbol{r}} = d_\mathrm{v}\dot{\boldsymbol{r}} + u_1\boldsymbol{e}_x^\mathcal{B} \tag{2.13}$$

which does not include any transformation matrices.

Observe from (2.13) that $\boldsymbol{e}_x^\mathcal{B}$ and $\boldsymbol{c} := (m_\mathrm{A}+m)\ddot{\boldsymbol{r}} - d_\mathrm{v}\dot{\boldsymbol{r}}$ always point in the same direction. Hence, the orientation of the vehicle's longitudinal axis can be expressed in terms of the derivatives of the flat output $\boldsymbol{r}$ as

$$\boldsymbol{e}_x^\mathcal{B} = \frac{\boldsymbol{c}}{\|\boldsymbol{c}\|}\,. \tag{2.14}$$

The unit vector $\boldsymbol{e}_y^{\mathcal{B}}$ can be uniquely determined from the roll angle, which in turn is a flat output. The third unit vector $\boldsymbol{e}_z^{\mathcal{B}}$ is obtained by

$$\boldsymbol{e}_z^{\mathcal{B}} = \frac{\boldsymbol{e}_x^{\mathcal{B}} \times \boldsymbol{e}_y^{\mathcal{B}}}{\|\boldsymbol{e}_x^{\mathcal{B}} \times \boldsymbol{e}_y^{\mathcal{B}}\|} \,. \tag{2.15}$$

Given the body fixed unit vectors $\boldsymbol{e}_x^{\mathcal{B}},\ \boldsymbol{e}_y^{\mathcal{B}},\ \boldsymbol{e}_z^{\mathcal{B}}$, the orientation of the vehicle can be uniquely determined and so the rotation matrix $\boldsymbol{R}_{\mathcal{I}}^{\mathcal{B}}$.

Velocity $\boldsymbol{v}$: The velocity vector $\boldsymbol{v}$ is obtained through the kinematic relationship

$$\boldsymbol{v} = \boldsymbol{R}_{\mathcal{I}}^{\mathcal{B}} \dot{\boldsymbol{r}} \,. \tag{2.16}$$

Angular velocity $\boldsymbol{\omega}$: The angular velocity $\boldsymbol{\omega}$ can be obtained from the rotation matrix and its derivative:

$$\boldsymbol{\omega} = (\dot{\boldsymbol{R}}\boldsymbol{R}^{\top})^{\vee} \,, \tag{2.17}$$

where we omitted the indication of the coordinate system for clarity. The symbol $(\cdot)^{\vee}$ denotes the vee operator, which for $SO(3)$ maps a skew-symmetric traceless matrix into $\mathbb{R}^3$.

Collective thrust input u_1: The collective thrust follows from (2.13) as

$$u_1 = \|(m_{\mathrm{A}} + m)\ddot{\boldsymbol{r}} - d_{\mathrm{v}}\dot{\boldsymbol{r}}\| \,. \tag{2.18}$$

Torque inputs u_2, u_3, u_4: The torque inputs can be obtained from the rotational part of the equation of motion (2.4). However, the angular acceleration has to be expressed in terms of time derivatives of flat outputs first. From (2.17), we obtain

$$\dot{\boldsymbol{\omega}} = (\ddot{\boldsymbol{R}}\boldsymbol{R}^{\top} - (\dot{\boldsymbol{R}}\boldsymbol{R}^{\top})^2)^{\vee} \tag{2.19}$$

through differentiation and basic matrix algebra. Again, the coordinate system indication is omitted for clarity. With the expression for angular acceleration (2.19), (2.4) can be solved for $\boldsymbol{u}_{2:4}^{\mathcal{B}}$.

All states and inputs have been expressed in terms of the flat output $\boldsymbol{\lambda}$.

2.3.2 Control Formulation

This subsection introduces the HippoCampus control system for tracking a prescribed trajectory or for tracking waypoints based on [42, 53]. The control system uses the differential flatness property, which was analyzed in the previous subsection.

For a given desired trajectory in the space of flat outputs $\boldsymbol{\lambda}_{\text{des}} = [\boldsymbol{r}_{\text{des}}^{\top} \ \phi_{\text{des}}]^{\top}$, we can compute the desired angular velocity $\boldsymbol{\omega}_{\text{des}}$ and the desired orientation $\boldsymbol{R}_{\text{des}}$ because both can be expressed as algebraic functions of $\boldsymbol{\lambda}_{\text{des}}$ according to the results of the previous subsection.

Define tracking errors for position $\boldsymbol{r}$, orientation $\boldsymbol{R}$, and angular velocity $\boldsymbol{\omega}$ as follows:

$$\boldsymbol{e}_r = \boldsymbol{r} - \boldsymbol{r}_{\text{des}} \,, \tag{2.20}$$

$$\boldsymbol{e}_R = \frac{1}{2}(\boldsymbol{R}_{\text{des}}^{\top}\boldsymbol{R} - \boldsymbol{R}^{\top}\boldsymbol{R}_{\text{des}})^{\vee} \,, \tag{2.21}$$

$$\boldsymbol{e}_\omega = \boldsymbol{\omega} - \boldsymbol{\omega}_{\text{des}} \,. \tag{2.22}$$

If these tracking errors vanish HippoCampus is precisely tracking the prescribed trajectory. We propose to use feedback terms that include the tracking errors and feedback linearization terms for known system dynamics. Moreover, feedforward terms are included for states that cannot be easily measured. The control inputs to the system u_1, u_2, u_3 and u_4 are defined accordingly in what follows.

The collective thrust u_1 is chosen as

$$u_1 = (-k_r \boldsymbol{e}_r + (m_{\text{A}} + m)\ddot{\boldsymbol{r}}_{\text{des}} + d_{\text{v}}\dot{\boldsymbol{r}}_{\text{des}})\, \boldsymbol{e}_x^{\mathcal{B}} \,. \tag{2.23}$$

The thrust control law (2.23) generates such that the vehicle's center of gravity follows the desired trajectory $\boldsymbol{r}_{\text{des}}$. It provides two feedforward terms that consider $\dot{\boldsymbol{r}}_{\text{des}}$ and $\ddot{\boldsymbol{r}}_{\text{des}}$ and a feedback term for the position error $\boldsymbol{e}_r$. Since the collective thrust can only act along the longitudinal axis of the vehicle, the control law (2.23) decreases the collective thrust if the vehicle orientation and the required force for tracking $\boldsymbol{r}_{\text{des}}$ do not align. This frees actuator bandwidth to correct the orientation via an attitude control law.

The attitude control law ensures that the vehicle tracks the desired orientation $\boldsymbol{R}_{\text{des}}$. The body torque inputs to the system are computed as

$$\begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = -k_R \boldsymbol{e}_R - k_\omega \boldsymbol{e}_\omega + \boldsymbol{\omega} \times (\boldsymbol{J} + \boldsymbol{J}_{\text{A}})\boldsymbol{\omega} - m_{\text{A}} \boldsymbol{v} \times \boldsymbol{\omega} - \boldsymbol{D}_\omega \boldsymbol{\omega} \,. \tag{2.24}$$

The attitude control law (2.24) consists of feedback terms on the orientation error and on the angular velocity error as well as feedback linearization terms that cancel out the Coriolis and the damping terms. The feedforward cross-coupling term $m_{\text{A}}\boldsymbol{v} \times \boldsymbol{\omega}$ requires velocity measurements. In practice, if no reliable velocity measurements are available, the term should be omitted. Due to its second order nature, it contributes little to the overall performance.

The described control structure can be used for waypoint tracking as well. Set $\boldsymbol{r}_{\text{des}} = \boldsymbol{r}_{\text{WP}}$, with $\boldsymbol{r}_{\text{WP}}$ being the waypoint location. Since $\boldsymbol{r}_{\text{WP}}$ is constant, $\dot{\boldsymbol{r}}_{\text{des}}$ and $\ddot{\boldsymbol{r}}_{\text{des}}$ vanish in (2.23). The desired angular velocity $\boldsymbol{\omega}_{\text{des}}$ in (2.24) cannot be computed from $\boldsymbol{r}_{\text{WP}}$ anymore and is therefore set to zero. The resulting control law simplifies to a cascaded control scheme where the setpoint for the angular velocity is set by the orientation control loop.

2.3.3 Experimental Example

The geometric control laws are applied for tracking a circular trajectory underwater. Figure 2.4 demonstrates the experimental setup.

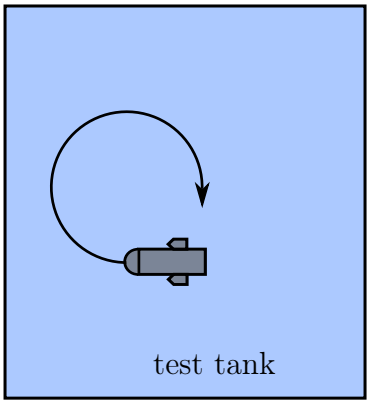


Figure 2.4: Experimental setup for tracking circular trajectory.

The desired circular trajectory can be formulated with the flat outputs in closed form as

$$\boldsymbol{r}_{\text{circle}} = \begin{bmatrix} a\,\cos(2\pi\,\frac{t}{T_{\text{circle}}}) \\ a\,\sin(2\pi\,\frac{t}{T_{\text{circle}}}) \\ z_{\text{des}}\,, \end{bmatrix}, \qquad \phi_{\text{des}} = 0\,, \tag{2.25}$$

where a is the radius of the circle, T_{circle} is the circular period, and z_{des} is the desired water depth. All other desired states can be derived from (2.25) by using the relations (2.12) – (2.19). A version of the control laws (2.23) and (2.24) is implemented on HippoCampus, albeit the absolute position information is only available for the depth dimension. This introduces translational symmetry of the circle. The desired depth is set to $z_{\text{des}} = 0.75\,\text{m}$ and the circular period to $T_{\text{circle}} = 8\,\text{s}$. Instead of the radius a, a maximum desired thrust is prescribed to ensure that the vehicle does not collide with the tank walls.

Experimental results are shown in Fig. 2.5 and Fig. 2.6 for two consecutive revolutions. Data is obtained from the on-board sensors after extended Kalman filtering. In Fig. 2.5, the yaw angle linearly increases as expected from tracking a circular trajectory. The roll angle is held at zero by the controller and the depth is constantly at 0.75 m, as desired.

The angular rates associated with the signals are shown in Fig. 2.6. Both the pitch and roll rate oscillate around zero while the yaw rate has an offset, which is a consequence from the continuously increasing yaw angle.

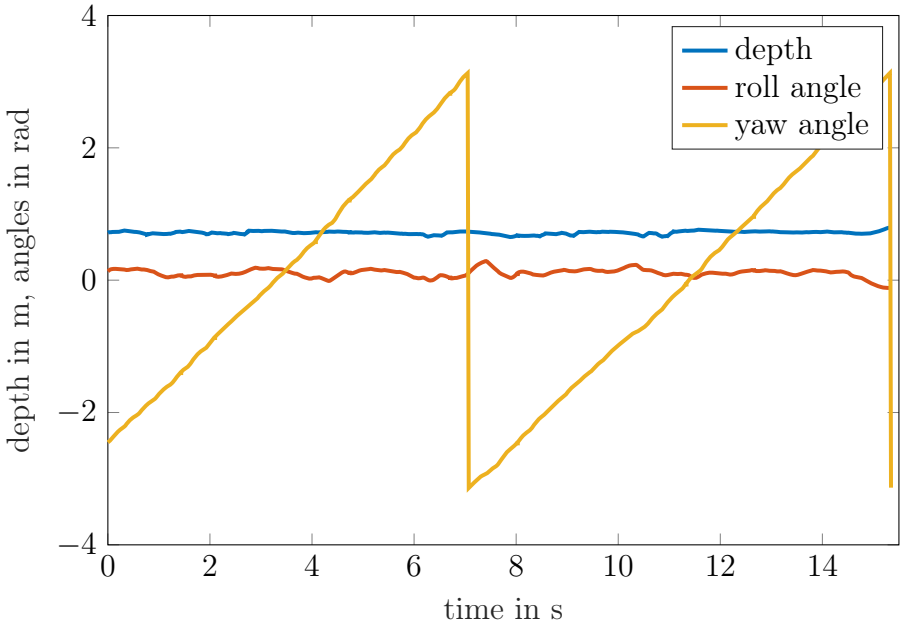


Figure 2.5: Roll angle, yaw angle, and depth for two consecutive circular trajectories obtained from on-board sensors after extended Kalman filtering.

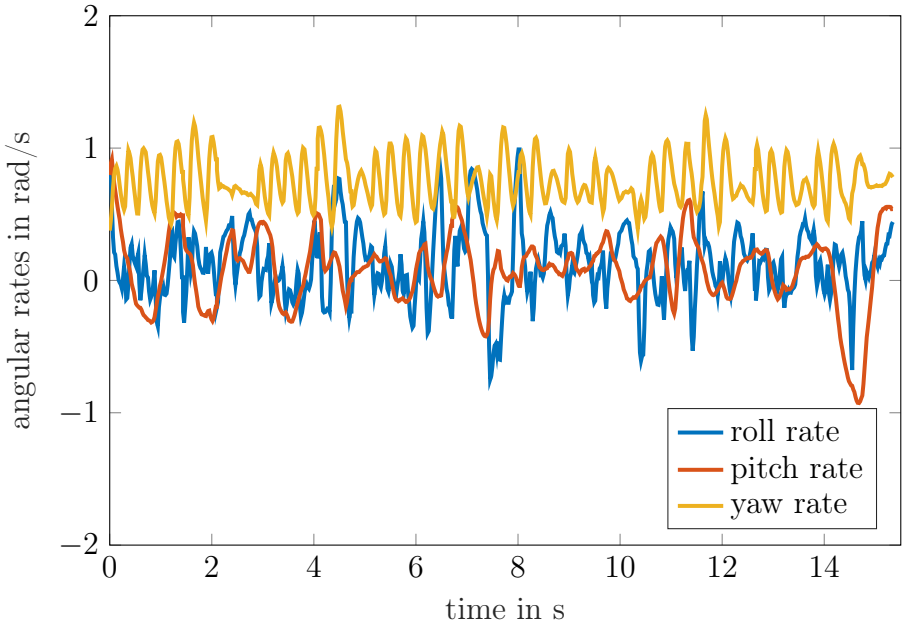


Figure 2.6: Angular velocities for two consecutive circular trajectories obtained from on-board sensors after extended Kalman filtering.

2.4 Receding Horizon Sliding Control for Path Following

The feedback controller of the previous section is concerned with trajectory tracking of the underactuated HippoCampus vehicle. This section introduces a receding horizon control strategy for path following. The approach is based on the recently proposed receding horizon sliding control (RHSC) technique introduced by [27]. The main objective is to obtain sufficient path following performance despite uncertain hydrodynamic parameters, which are often not exactly known for μAUVs. Unlike trajectory tracking problems, path following does not include time constraints on the vehicle position, but only spatial restrictions. Path following problems can be conveniently defined by a set of waypoints that are connected by spatial curves. The simplest curve connections are line segments. However, they introduce a non-smooth path with edges at the waypoints and a parametric continuity C^0. The μAUV is supposed to follow the path and keep its cross-track error to the path as small as possible.

The presented RHSC for path following is applicable to non-smooth path geometries and explicitly exploits the piecewise linear structure of the path. It shows sufficient performance in simulations, despite actuator constraints, model uncertainties, and localization noise.

2.4.1 Overview of Path Following Methods

Path following can be achieved either with explicit feedback-based control or model predictive control (MPC). The line of sight (LOS) method [15] is a widely applied feedback-based control framework for path following. A guidance system generates heading references for pitch and yaw angles, which are tracked by a feedback controller. The LOS method is based on a geometric relationship where a LOS vector from the vehicle position is projected onto the path. Despite the fact that LOS methods have been studied extensively in the last decade and have been successfully applied to 3-DOF underactuated surface vessels [3, 44], contributions which consider the full AUV dynamics are rare. The LOS path following problem in space is often decomposed in a vertical and a horizontal part, where a 3-DOF system is assumed in each plane [45]. In [2], full AUV dynamics are considered, which are similar to the ones in this chapter, to guide an AUV along a single line path. However, no rapid directional changes of the path are considered in this prior work. In confined environments, rapid directional changes are crucial for μAUV path following.

While feedback-based control laws are very efficient, MPC based methods allow the explicit incorporation of physical constraints and actuator limits. Furthermore, MPC's "look-ahead-in-time" behavior can compensate for time delays and future errors, which is beneficial for an agile vehicle behavior. Different MPC implementations have been reported for marine vessels, mostly for surface vessels with planar dynamics [58] or for AUVs with decoupled dynamics [74].

A control approach which combines the robustness of nonlinear feedback control [37] with the qualities of MPC can yield an agile path following behavior, suitable for confined environments where rapid directional changes are required. The RHSC design procedure is inspired by the sliding control method [72]. The RHSC control approach is capable of effectively managing constraints and future information. This is achieved by solving a finite-time constrained optimal control problem at every time-instant and implementing the first element of the resulting control plan before repeating the same procedure at the next time-step with the horizon shifted by one time-step. The main difference between classical MPC and RHSC lies in the formulation of the cost function. MPC typically minimizes a cost function that weights off tracking performance and control action. On the other hand, RHSC drives the system to a state-space manifold where desired error dynamics hold.

The reason for choosing RHSC over classical MPC is the expected improvement of the closed-loop behavior subject to system uncertainties. In most complex applications, uncertainties are ignored in the MPC design phase due to computational complexity issues. It relies on the capability of the nominal controller to manage the disturbances instead, i. e. it uses a property often referred to as the inherent robustness of the controller [1]. As pointed out in [18], MPC usually achieves higher levels of inherent robustness when the input penalty is increased. However, high input penalties also yield a sluggish controller that does not fully exploit the actuator potentials. RHSC bypasses this tradeoff by introducing desired error dynamics. The error dynamics can be tuned to yield a robust controller without restricting control action. Hence, RHSC yields inherent robustness and agile actuator performance, while typically being easy to tune.

2.4.2 Control Methodology

The presented control strategy is a model predictive based method and hence requires a model of the system. This subsection revisits the μAUV and defines the control objective. Subsequently, auxiliary control laws for roll angle and speed error regulation are given. Finally, the core path-following logic is synthesized based on the RHSC method.

The full nonlinear model (2.2) is embedded within the controller. The rotations are parametrized with quaternions to avoid gimbal lock during optimization. For the remainder of this section, the nonlinear equations of motion (2.2) are compactly written as

$$\dot{\boldsymbol{\xi}} = \bar{\boldsymbol{f}}_c(\boldsymbol{\xi}) + \bar{\boldsymbol{u}}\,, \tag{2.26}$$

where $\boldsymbol{\xi} = [\boldsymbol{\nu}^\top \ \boldsymbol{\eta}^\top]^\top = [x \ y \ z \ \epsilon_0 \ \epsilon_1 \ \epsilon_2 \ \epsilon_3 \ u \ v \ w \ p \ q \ r]^\top$ is the full state vector and $\bar{\boldsymbol{u}} = [\boldsymbol{0}^\top \ u_1 \ 0 \ 0 \ u_2 \ u_3 \ u_4]^\top$ contains the input variables.

Control Objective and Cross-Track Error Parameterization

Assume that a set of N_{WP} waypoints $\text{WP}_{1:N_{\text{WP}}}$ is provided for navigation purposes. A path is defined by connecting two successive waypoints with line segments l, as illustrated in Fig. 2.7 for the waypoints WP_i and WP_{i+1}. The cross-track error e is the orthogonal distance between one of those line segments and the center of mass of the μAUV. Since the path is non-smooth and the μAUV has underactuated dynamics, it is important to clarify the path following objective. We can either seek exact passing through all waypoints (Dubins path) or minimize the cross-track error while approaching the waypoints as closely as possible, referred to as approximating paths [43]. In the following the objective of approximating paths is discussed. For most operations in monitoring and exploration an overall small cross-track error is more important than exactly passing through waypoints. Hence, for a given set of waypoints, the μAUV control objective is to minimize the cross-track error while maintaining a prescribed surge velocity. Note the difference to the control objective of the previous section, where a μAUV was directly guided towards the next waypoint without considering the cross-track error to a path.

The cross-track error in 2-D is usually computed along a line segment's normal vector. In 2-D each line segment has a unique normal vector. In 3-D however, each line segment's orthogonal space is two-dimensional. Hence, each line segment has infinitely many normal vectors. However, the tangent space of any one-dimensional surface is one-dimensional as well and it can be determined from the column space of the surface Jacobian. Hence, we can express the cross-track error of the i-th line segment e_i by first computing the line segment's Jacobian

$$\boldsymbol{J}(\boldsymbol{x}_{\text{path},i}) = \frac{\partial l_i}{\partial \boldsymbol{x}}(\boldsymbol{x}_{\text{path},i}) \tag{2.27}$$

and second by exploiting that the cross-track error e_i is orthogonal to $\boldsymbol{J}_i(\boldsymbol{x}_{\text{path},i})$ and to the difference $\boldsymbol{r} - \boldsymbol{x}_{\text{path},i}$, where $\boldsymbol{x}_{\text{path},i}$ is an arbitrary point on the line segment l_i, for example, a waypoint:

$$e_i = \frac{\|(\boldsymbol{r} - \boldsymbol{x}_{\text{path},i}) \times \boldsymbol{J}_i(\boldsymbol{x}_{\text{path},i})\|}{\|\boldsymbol{J}_i(\boldsymbol{x}_{\text{path},i})\|}\,. \tag{2.28}$$

Since the Jacobian $\boldsymbol{J}_i(\boldsymbol{x}_{\text{path},i})$ is constant along each line segment, the cross-track error can be evaluated very efficiently within an optimization.

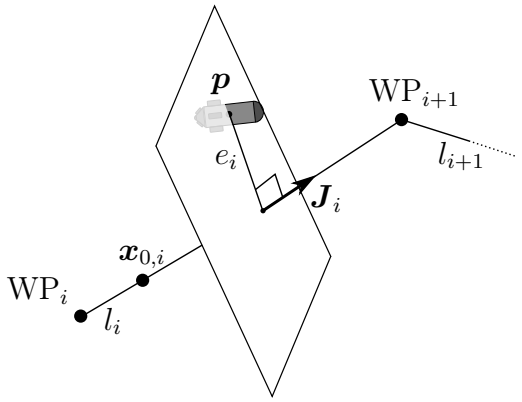


Figure 2.7: Parametrization of cross-track error.

Surge Velocity and Roll Regulation

In addition to a desired path, we assume there is a prescribed surge velocity for following the path. Furthermore, the desired roll angle is set to zero when possible. The μAUV control system is augmented with proportional (P) and proportional-derivative (PD) feedback control loops for the surge velocity control and the roll angle regulation, respectively:

$$u_1 = k_{\text{P,u}}(u - u_{\text{des}})\,, \tag{2.29}$$

$$u_2 = k_{\text{P},\phi}\phi + k_{\text{D},\phi}q\,. \tag{2.30}$$

The parameters $k_{\text{P,u}}$, $k_{\text{P},\phi}$, $k_{\text{D},\phi}$ are the control gains. In (2.29), u_{des} represents the desired surge velocity. The roll angle ϕ in (2.30) can be recovered from the quaternion representation using the four quadrant inverse tangent function

$$\phi = \operatorname{atan}\left(2(\epsilon_2\epsilon_3 + \epsilon_1\epsilon_0)), 1 - 2(\epsilon_1^2 + \epsilon_2^2)\right)\,. \tag{2.31}$$

A modified system model follows by substituting (2.29), (2.30) in (2.26) and can be written as

$$\dot{\boldsymbol{\xi}} = \boldsymbol{f}_{\text{c}}(\boldsymbol{\xi}, \boldsymbol{u})\,. \tag{2.32}$$

The remaining control inputs for navigating the μAUV are the yaw and pitch moments, i. e. $\boldsymbol{u} = [u_3\ u_4]^\top$. The receding horizon sliding control strategy computes these inputs.

Yaw and Pitch Control

The RHSC method requires a discrete-time representation of the system dynamics, which we obtain from (2.32) through Euler discretization:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x}_k) + \boldsymbol{u}_k \,. \tag{2.33}$$

The time step is indicated by subscripts, representing the system's signals at times kT_{s} and $(k+1)T_{\mathrm{s}}$, respectively, where T_{s} is the sampling time.

Following the design procedure from [27], the first step in synthesizing the RHSC is to define a sliding variable. Notice that the relative degree of the cross track error e_i from (2.28) is equal to two. Hence, we apply a first order difference operator to $e_{k,i}$ in order to obtain a sliding variable

$$s_k = e_{k+1,i} - \rho e_{k,i} \,. \tag{2.34}$$

The tuning parameter ρ should be chosen such that $0 \leq \rho < 1$. Intuitively, the control goal is now reduced to driving s_k to zero and exploiting the fact that the desired error dynamics are achieved when $s_k = 0$. The index i corresponds to the line segment that is closest to the μAUV position at timestep k.

Formally, RHSC minimizes s_k over an N-step prediction horizon. For convenience, we define

$$\boldsymbol{S}_{k+1} = \begin{bmatrix} s_{k+1} & s_{k+2} & \dots & s_{k+N} \end{bmatrix}^\top . \tag{2.35}$$

Likewise, the state and control vector sequences are abbreviated as follows:

$$\boldsymbol{\Xi}_k = \begin{bmatrix} \boldsymbol{\xi}_k & \boldsymbol{\xi}_{k+1} & \cdots & \boldsymbol{\xi}_{k+N+1} \end{bmatrix}^\top , \tag{2.36}$$

$$\boldsymbol{U}_k = \begin{bmatrix} \boldsymbol{u}_k & \boldsymbol{u}_{k+1} & \cdots & \boldsymbol{u}_{k+N} \end{bmatrix}^\top . \tag{2.37}$$

The explicit form of the minimization problem that the RHSC solves at every subsequent sampling-instant reads as follows:

$$\begin{aligned} \min_{\boldsymbol{\Xi}_k, \boldsymbol{U}_k} \quad & \|\boldsymbol{S}_{k+1}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{\xi}_{j+1} = \boldsymbol{f}_{\mathrm{d}}(\boldsymbol{\xi}_i, \boldsymbol{u}_j),\ j = k, \dots, k+N \,, \\ & \boldsymbol{u}_{\mathrm{lb}} \leq \boldsymbol{u}_j \leq \boldsymbol{u}_{\mathrm{ub}},\ j = k, \dots, k+N \,, \\ & \boldsymbol{\xi}_k = \boldsymbol{\xi}(kT_s) \,. \end{aligned} \tag{2.38}$$

The problem is subject to the system dynamics and input constraints. Lower and upper bounds on the control actions are denoted by $\boldsymbol{u}_{\mathrm{lb}}$ and $\boldsymbol{u}_{\mathrm{ub}}$. The problem is initialized with the current state that we assume to be measurable, albeit being noisy. After solving (2.38), the first element of the solution sequence is applied to the AUV. The problem is repeatedly solved in each sampling-instant.

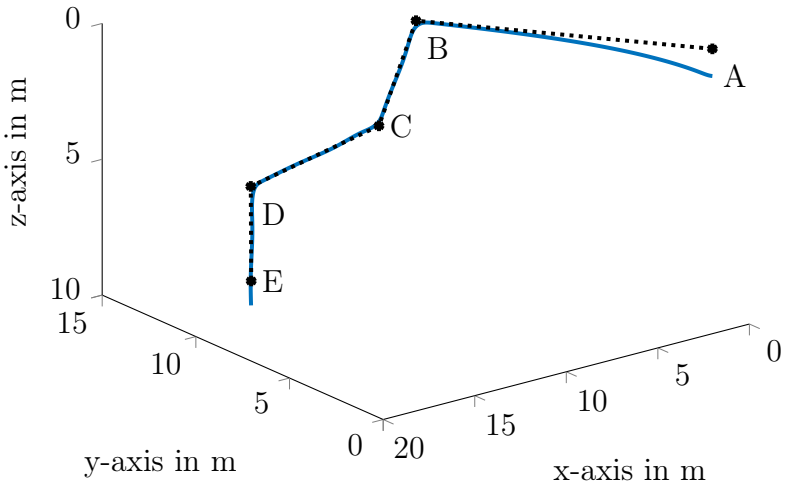
2.4.3 Simulations

The control system performance is demonstrated in numerical simulations. The sampling time of the RHSC is set to $T_\mathrm{s} = 100\,\mathrm{ms}$ and the prediction horizon length is $N = 8$ steps. The path is defined in $\mathbb{R}^3$ by waypoints connected with line segments. The RHSC is analyzed in two different scenarios. In the first, *ideal* scenario, the controller has access to the exact model parameters. In the second, *realistic* scenario, the controller model differs significantly from the simulated μAUV. The hydrodynamic added Coriolis matrix $\boldsymbol{C}_\mathrm{A}$ vanishes and the hydrodynamic damping $\boldsymbol{D}_\mathrm{A}$ has an error of 50%. Furthermore, the position measurement is corrupted by zero-mean white Gaussian measurement noise with a standard deviation of 0.2 m which is a realistic assumption in confined environments [19]. Also, no actuator constraints apply in the ideal case, while we impose hard actuator limits in the realistic case. A sample path is defined by five waypoints A, B, C, D, E whose coordinates are summarized in Table 2.1.

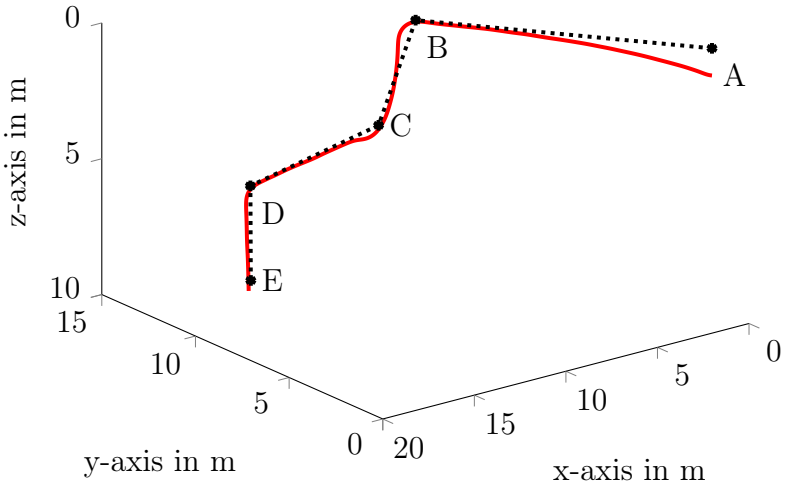
Table 2.1: Waypoint coordinates for simulation scenarios.

Waypoint	Coordinates
A	$(1, 1, 0)$
B	$(9, 9, 0)$
C	$(10, 10, 4)$
D	$(17, 10, 5)$
E	$(18, 9, 8)$

The μAUV starts at rest at the initial position $\boldsymbol{r}_0 = [1\ 0\ 0]^\top$. The obtained path following results are presented in Figs. 2.8 – 2.11 . Figure 2.8 illustrates the path following performances for the ideal (a) and the realistic case (b). The deviations from the path are smaller for the ideal case, as expected. Also, in the realistic scenario, the μAUV converges to the desired path. This is confirmed by Fig. 2.9 where the cross-track error as a function of time is shown for both scenarios. The cross-track error does not exceed 0.4 m in the realistic case. This magnitude of the cross-track error is usually sufficient for monitoring and exploration in terms of associating meauserement locations with measurements. However, if obstacles are a problem, waypoint planning needs to be conservative to consider possible path deviations. Note that spiking of the cross-track error cannot be avoided because the path is defined by points connected with line segments and the cross-track error is computed with regard to line segments. This is a direct consequence of the approximating path following approach, for which the vehicle trajectory does not pass through the waypoints because it is non-smooth.



(a) Ideal scenario.



(b) Realistic scenario.

Figure 2.8: Simulated position of μAUV's center of mass and the desired path defined by waypoints in 3-D. The top subfigure illustrates path following results without measurement noise, without actuator saturation, and with exact knowledge of the dynamical model for the controller design. The bottom subfigure shows results of the realistic case for which measurement noise, actuator saturation, and wrong assumptions of hydrodynamic added mass and of damping were included. The initial vehicle position at rest is $\boldsymbol{r}_0 = [1\ 0\ 0]^\top$.

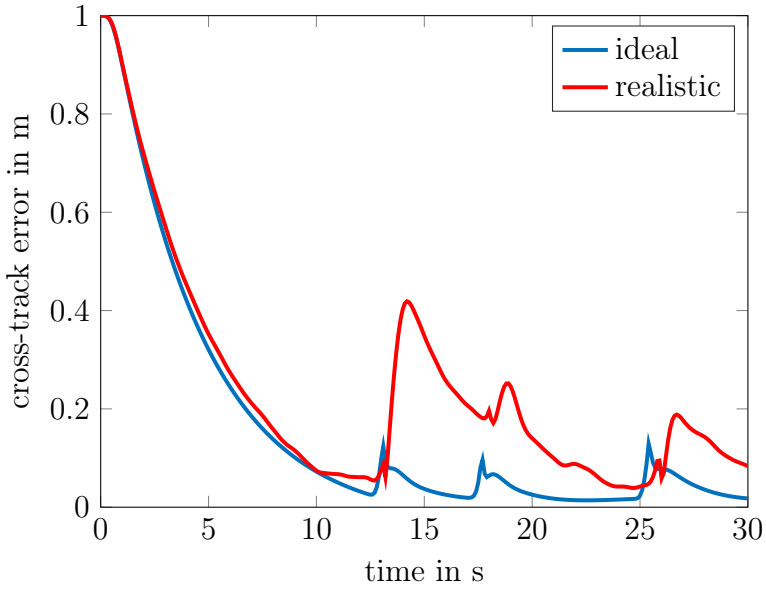
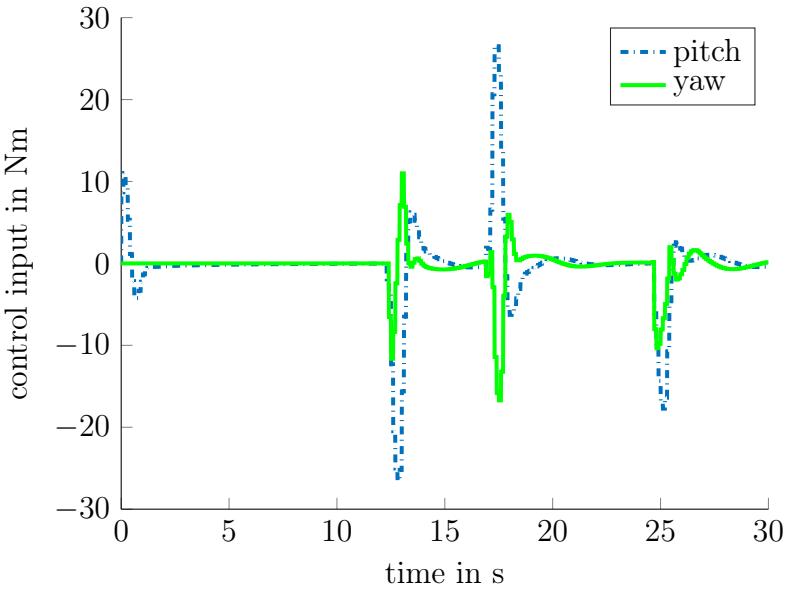


Figure 2.9: Cross-track error for ideal and realistic simulation scenarios.

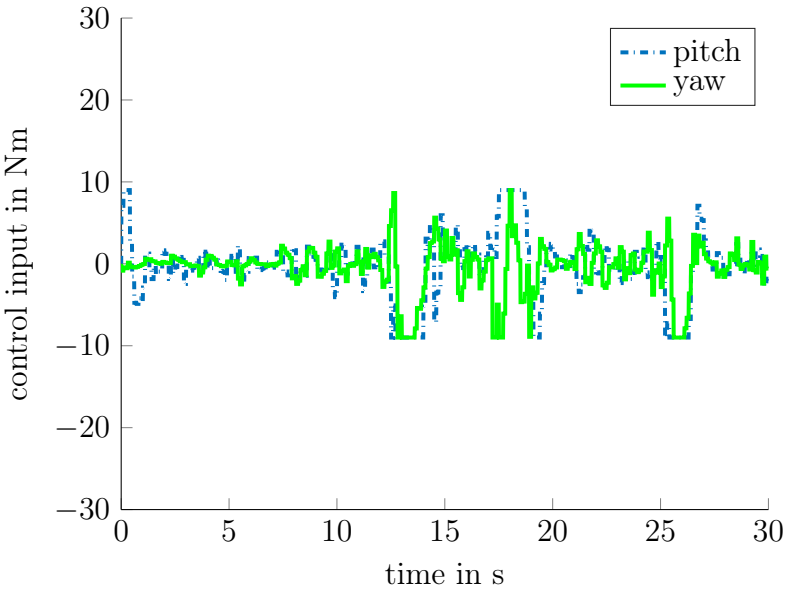
Figure 2.10 shows the yaw and pitch moments. In the ideal case, the actuator signals spike in proximity to the waypoints in order to enable a quick direction change of the μAUV to converge to a new line segment. In the realistic case, actuator limitations are imposed and the moments are applied to the μAUV saturate. It is evident that the actuator signals exhibit noise due to noise in the position signal. The roll angle is shown in Fig. 2.11. With realistic conditions, the roll angle stabilization is worse than in the ideal case. The presented results were obtained on a Quadcore-CPU with 2.66 GHz and 8 GB RAM, where the simulation including the RHSC routine runs twice as fast as real-time. Hence, the RHSC is computationally feasible for μAUVs.

2.5 Summary

This chapter presented the mechatronic design, dynamics, and controls development of the HippoCampus μAUV. HippoCampus is designed for operations in confined environments. The dynamics of HippoCampus have been analyzed, and it was shown that, under certain assumptions, the design renders differential flatness. Based on the dynamical analysis, a nonlinear geometric trajectory tracking control law has been proposed. The control law does not make the simplification that rotation states evolve in Euclidean space. In an experiment HippoCampus tracked a circular trajectory. A second control law has been introduced for path following, which is based on the RHSC approach [27]. Simulations were conducted to analyze the performance in terms of uncertainties for non-smooth paths, resulting in robust path following under realistic conditions.



(a) Ideal scenario.



(b) Realistic scenario.

Figure 2.10: Heave and yaw moments applied to μAUV in path following. The top subfigure illustrates the actuator commands for the ideal case where no actuator saturation applies. The bottom subfigure shows actuator moments in the realistic case where actuator saturation occurs. The noise in the actuator signal is due to the position measurement noise.

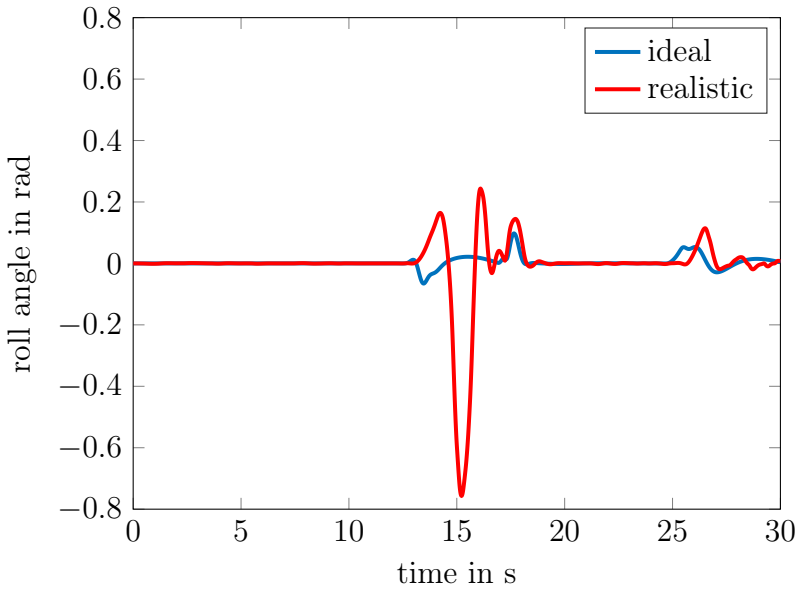


Figure 2.11: Roll angle for ideal and realistic path following scenarios.

Chapter 3 Localization in Confined Underwater Environments

Localization is the key capability of mobile robotic systems for performing tasks such as exploration and monitoring. This chapter presents self-localization methods for μAUVs in confined underwater environments. The unique requirements for such systems are provided. Moreover, the principles of spherical and hyperbolic localization are explained. Submerged μAUVs cannot use most of the self-localization techniques that are commonly used for other mobile robots, such as GPS and vision based systems. This chapter introduces two underwater localization frameworks that fulfill the low cost and compact form factor requirements. The first approach is a hyperbolic acoustic method based on differences of signal time of flights. The second approach is a spherical localization method based on the attenuation of electro-magnetic waves. Both approaches are evaluated in experimental proof of concepts.

3.1 Spherical and Hyperbolic Localization

This section covers specifications of underwater localization for μAUVs and introduces suitable methods. Desirable specifications of μAUV localization systems should include, besides being low cost and compact, on-board processing capabilities and passivity. On-board processing ensures that the position estimation is computed at the μAUV itself. With on-board localization, communicating externally computed position information through water is not required. This is advantageous because communication in liquids always introduces latencies and information losses. Passivity of a localization system allows to scale self-localization to large fleets. It implies that the signal processing is not affected by the number of robots performing localization. This can be achieved with ambient one-way reference signals that are transmitted to the μAUVs. Thereby, anchored reference beacons with known positions are required to obtain passivity and on-board

processing for localization. Spherical and hyperbolic localization principles are the most common approaches for beacon based localization. The spherical approach uses the ranges between beacons and the receivers for localization, whereas the hyperbolic approach uses the differences of ranges instead of the ranges themselves. Obtaining range differences is easier than obtaining ranges themselves, however results are less accurate. The following subsections details the two aforementioned approaches.

3.1.1 Spherical Localization

Spherical self-localization can be decomposed into two sub-problems: (1) Range estimation from receiver to beacons. (2) Fusion of estimated ranges to obtain the receiver position.

For range estimation, the receiver determines ranges to the beacons by measuring either the time of flight or the strength of the signal that is transmitted by the beacons. The time of flight can only be determined if clocks between receiver and beacons are synchronized. There are two common ways of clock synchronization: clocks can use continuous communication to stay synchronized or they are synchronized prior to deployment without further interaction. In underwater settings, communication-based clock synchronization is difficult to implement because underwater communication is unreliable. Initial one-time synchronization prior to deployment is often also not feasible because the free-run drift of off-the-shelf real-time clocks (RTCs) is usually around 2 ppm, which corresponds to a positioning drift error of 3 mm per second. This is too high for operations in confined environments. In principle, customized synchronous-clock systems with higher performance and with the free-run drift being sufficiently small can be developed. For μAUVs, this is not feasible either because such clocks do not meet typical size and cost requirements. Range estimation based on reference signal strength, however, does not require clock synchronisation.

The problem of spherical self-localization can be formulated as follows. Consider N beacons located at known positions, with the i-th beacon being at position $\boldsymbol{r}_i$. The receiver has the unknown position $\boldsymbol{p}$. The range between the receiver and the i-th beacon can be expressed by

$$R_i = \|\boldsymbol{p} - \boldsymbol{r}_i\| \,, \tag{3.1}$$

which is the equation for a sphere that is centered at the location of the i-th beacon $\boldsymbol{r}_i$. In the horizontal plane, the spheres reduce to circles. The intersection of all spheres is the receiver position of interest $\boldsymbol{p}$. Consequently, the receiver's position can be derived from geometrical relationships if the beacon locations are known and the ranges R_i can be estimated, as illustrated in Fig. 3.1. However, the spheres in general do not intersect

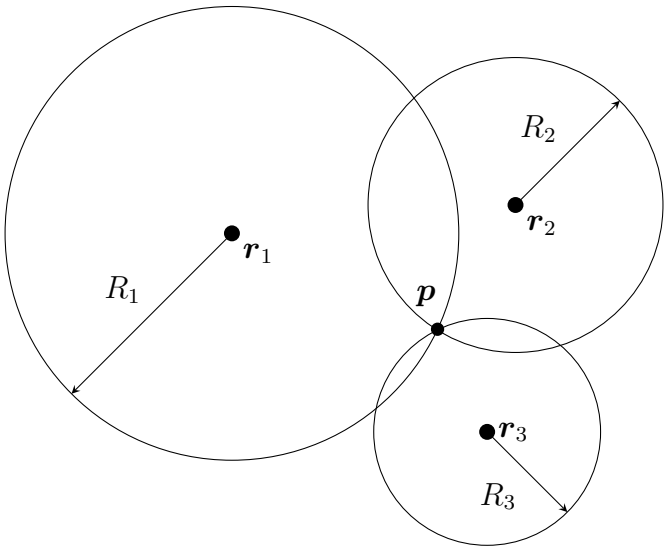


Figure 3.1: Spherical localization for determining the receiver position $\boldsymbol{p}$ at the intersection of circles (in 2-D) or spheres (in 3-D) with radii R_i. The positions of the beacons are denoted by $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ and $\boldsymbol{r}_3$.

due to uncertainty in the measurement model and noise. The intersection point has to be estimated from measurements by applying curve fitting techniques or Bayesian filtering methods. The latter are prefered because they compute continuous updates. Typical Bayesian filtering methods in the context of localization are extended Kalman filters (EKFs) and particle filters (PFs). In order for a receiver to self-localize in n dimensions, it requires range estimations to at least $n+1$ beacons. It can be shown that spherical positioning systems offer higher accuracy than hyperbolic systems in terms of Cramér-Rao bounds [9].

3.1.2 Hyperbolic Localization

Similar to spherical localization, hyperbolic localization approaches deploy anchored beacons as well, which emit ambient signals. However, unlike in spherical localization, the range between a single beacon and the receiver is not determined. Instead, range differences are estimated through measurements. Consider beacons i and j located at positions $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$, respectively. The receiver is located at the unknown position $\boldsymbol{p}$. The range difference $\Delta R_{i,j} = R_j - R_i$ can be expressed with the receiver position $\boldsymbol{p}$ and the known beacon positions $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$:

$$\Delta R_{i,j} = \|\boldsymbol{p} - \boldsymbol{r}_j\| - \|\boldsymbol{p} - \boldsymbol{r}_i\| \ . \tag{3.2}$$

Figure 3.2 illustrates the geometric solutions of (3.2) for three beacons and two range differences $\Delta R_{1,2}$ and $\Delta R_{1,3}$. The hyperbolas $\rho_{1,2}$ and $\rho_{1,3}$ are determined from range differences corresponding to pairs of beacons. The intersection of the hyperbolas is the receiver position. Similar to spherical localization, the hyperbolas in general do not intersect due to noise, and uncertainty in model and an estimation procedure for fitting or fusion is required.

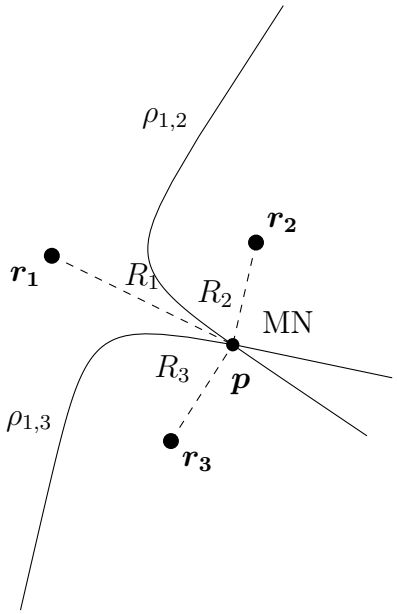


Figure 3.2: Hyperbolic localization for determining receiver's unknown position $\boldsymbol{p}$ at the intersection of hyperbolas $\rho_{1,2}$ and $\rho_{1,3}$. The hyperbolas are solutions of (3.2) and depend on pairwise differences of ranges R_1, R_2, and R_3. The beacon positions are denoted by $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ and $\boldsymbol{r}_3$.

3.2 Hyperbolic Acoustic One-Way Localization

This section presents an acoustic hyperbolic self-localization system. Anchored beacons emit acoustic signals, and the receiver measures differences in time delays of arrival (TDOAs) in those acoustic signals. The TDOA $\tau_{i,j}$ associated with two beacons i and j can be related to the range difference $\Delta R_{i,j}$ through the known speed of sound c_{sound}, and (3.2) is restated as

$$\bar{\tau}_{i,j} = \frac{\|\boldsymbol{p} - \boldsymbol{r}_i\| - \|\boldsymbol{p} - \boldsymbol{r}_j\|}{c_{\text{sound}}} \,. \tag{3.3}$$

This setup allows for the receiver to determine its position $\boldsymbol{p}$.

3.2.1 Methodology

The one-way hyperbolic localization framework includes two parts. First, the TDOAs for acoustic sequences (left-hand side of (3.3)) are determined via cross-correlation. Second, an unambiguous solution of the nonlinear hyperbolic equation (3.3) is estimated with an EKF, which is the estimation of the receiver's position. In general, the performance of hyperbolic localization quickly degrades outside the convex hull spanned by the beacons [9, 40]. In order to avoid this, localization is only performed within the convex hull.

The anchored beacons successively and repeatedly transmit acoustic signals of duration t_{signal}. The beacons do not send signals simultaneously. This is achieved with communication between the beacons and a coordinating unit. Hence the receiver observes only one signal at a time. The receiver does not require any prior knowledge about the signal other than its duration. Time delays of arrival are directly computed by cross-correlating two received signals. This is an advantage compared to systems which cross-correlate a received signal with an a priori saved signal because signals are modified through the transmission channel, degrading the cross-correlation result. Another advantage of one-way hyperbolic localization is the fact that synchronization between beacons and receiver is not required. The beacons' sending cycle and the receiver's recording loop (including computational overhead) have to repeat within a periodic duration of t_{signal}, which is within the timing tolerance of off-the-shelf RTCs at drift rates of 20 ppm. The receiver also requires knowledge about the sending sequence of the beacons. Since the sending sequence is established prior to operations, this information can be included on the receiver side.

Figure 3.3 illustrates the overall system for four beacons in a circular sending sequence. The beacons and the receiver are linked by the acoustic transmission channel, air and/or water. Every recording cycle, a new TDOA $\tau_{i,j}$ is determined and an EKF update is performed on the receiver side. The TDOAs are computed by cross-correlating the received signal from beacon i with the received signal from beacon j. Consider signal $s(t)$ sent by beacon i and a time shifted version of the signal $s(t + \Delta t)$ sent by beacon j. Both signals propagate through a noise corrupted channel and are sampled by the receiver. The resulting discrete-time signals read $s_i(n)$ and $s_j(n+m)$, where n and m are integers. The cross-correlation function

$$r_{i,j}(m) = \sum_{n=-\infty}^{\infty} s_i(n)\, s_j(n+m) \tag{3.4}$$

is a measure of similarity. It is maximized for the shift m, where both signals have the largest similarity. This value is equivalent to the TDOA between beacon i and j:

$$\tau_{i,j} = \arg\max_m r_{i,j}(m). \tag{3.5}$$

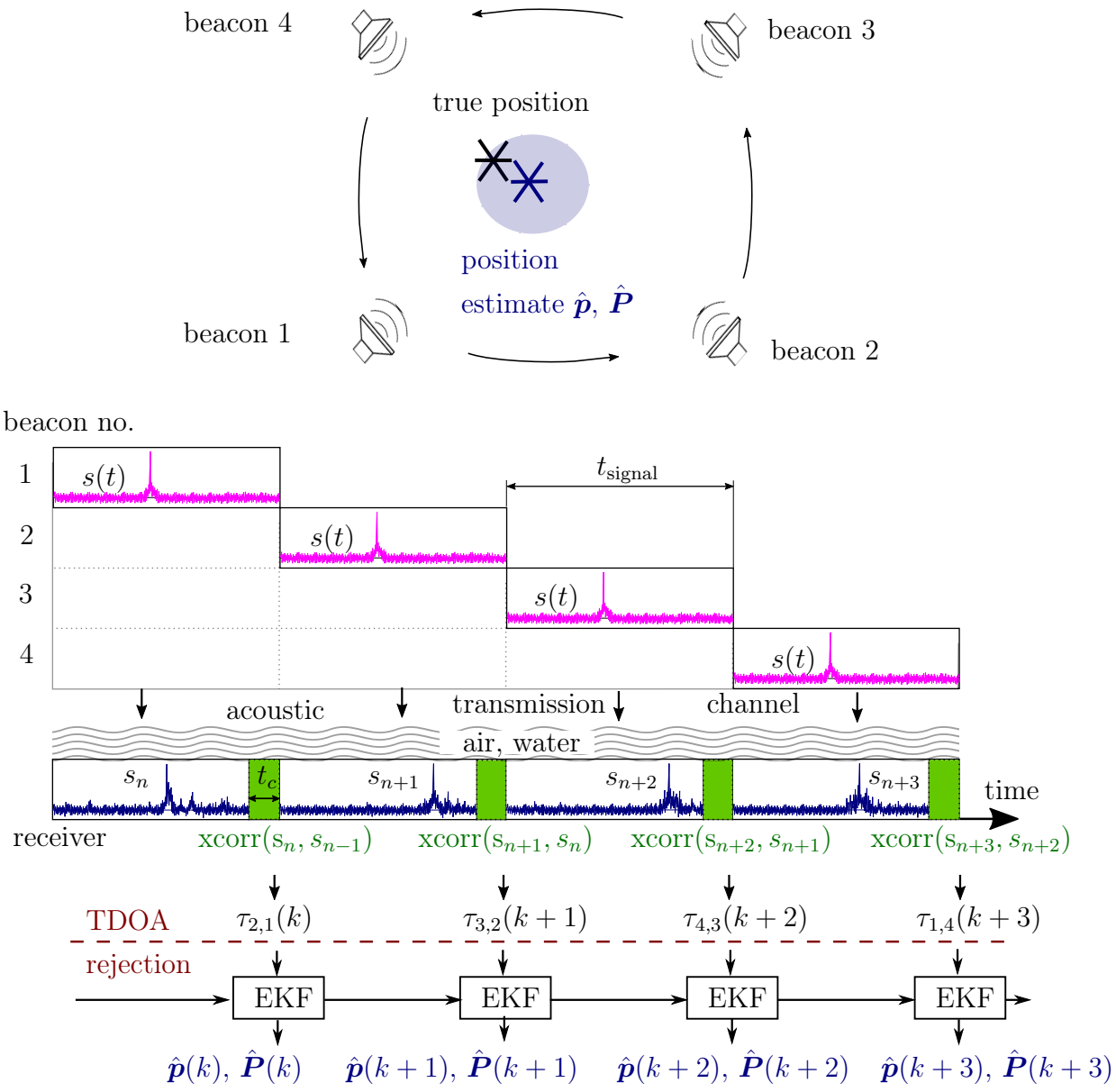


Figure 3.3: State estimation cycle illustrated for four beacons sending consecutively in a circular sequence. Each beacon sends a signal $s(t)$. After passing the propagation channel, the signal is sampled by the receiver. Received signals (in blue) are shifted because of differing time of flights from beacons to receiver. The received signal is cross-correlated with the previously received signal, and the peak is determined to estimate the TDOA. The TDOA passes the rejection algorithm and is fed to the EKF, which computes a new position estimate.

The system determines confidence intervals based on previous TDOAs. With this data, estimated TDOAs are rejected if they do not fall within a confidence interval.

The TDOA measurements τ are processed by an EKF algorithm to estimate the position state vector $\boldsymbol{p}$. They are directly included into the observation vector

$$\boldsymbol{\tau} = \left[\tau_{2,1} \ \cdots \ \tau_{i,j} \ \cdots \ \tau_{1,N}\right]^\top \sim \mathcal{N}(\bar{\boldsymbol{\tau}}, \boldsymbol{C}) \,, \tag{3.6}$$

where $\boldsymbol{C}$ is the measurement covariance matrix and $\mathcal{N}$ the normal distribution. This approach allows for tightly coupled sensor fusion with an IMU or a pressure sensor. By using a Bayesian approach, the loss of one or more TDOAs still results in a position update along a hyperbola. This allows for optimal utilization of the available information. It also improves the system robustness because glitches in the acoustic measurement system can be treated in a systematic way.

For the following EKF equations superscripts $(-)$ and $(+)$ denote a value gained in the filter prediction and in the filter update step, respectively.

The receiver's state dynamics are set to a random walk. Thus, the process equation at time step k reads

$$\boldsymbol{p}(k) = \boldsymbol{p}(k-1) + \boldsymbol{w}(k) \,. \tag{3.7}$$

The process noise vector $\boldsymbol{w}$ is assumed to be zero-mean Gaussian white noise with covariance matrix $\boldsymbol{Q}$. The non-linear measurement function $\bar{\boldsymbol{\tau}}(\boldsymbol{p})$ (3.3) defines a vector of TDOA mean values as functions of the receiver position. The EKF requires the Jacobian $\boldsymbol{J}_\mu(k)$ of $\bar{\boldsymbol{\tau}}(\boldsymbol{p})$, which consists of the derivatives $\boldsymbol{h}_{i,j}^\top(k)$ of (3.3):

$$\boldsymbol{h}_{i,j}^\top(k) = \nabla_{\boldsymbol{p}} \, \bar{\tau}_{i,j}(\boldsymbol{p}(k)) = \frac{1}{c_{\text{sound}}} \cdot \left[\frac{\boldsymbol{p} - \boldsymbol{r}_i}{|\boldsymbol{p} - \boldsymbol{r}_i|} - \frac{\boldsymbol{p} - \boldsymbol{r}_j}{|\boldsymbol{p} - \boldsymbol{r}_j|} \right]^\top_{\boldsymbol{p}=\hat{\boldsymbol{p}}^{(-)}(k)} \,, \tag{3.8}$$

$$\boldsymbol{J}_\mu(k) = \left[\boldsymbol{h}_{2,1}^\top(k) \ \cdots \ \boldsymbol{h}_{i,j}^\top(k) \ \cdots \ \boldsymbol{h}_{1,N}^\top(k) \right]^\top . \tag{3.9}$$

The predicted position state is

$$\hat{\boldsymbol{p}}^{(-)}(k) = \hat{\boldsymbol{p}}^{(+)}(k-1) + \boldsymbol{w}(k) \tag{3.10}$$

and its covariance reads

$$\hat{\boldsymbol{P}}^{(-)}(k) = \hat{\boldsymbol{P}}^{(+)}(k-1) + \boldsymbol{Q}. \tag{3.11}$$

The innovation and the innovation covariance are computed as

$$\boldsymbol{\kappa}(k) = \boldsymbol{\tau}(k) - \bar{\boldsymbol{\tau}}(\hat{\boldsymbol{p}}^{(-)}(k)) \tag{3.12}$$

and

$$\boldsymbol{S}(k) = \boldsymbol{J}_\mu(k) \hat{\boldsymbol{P}}^{(-)}(k) \boldsymbol{J}_\mu^\top(k) + \boldsymbol{C}(k) \tag{3.13}$$

respectively.

The Kalman gain

$$\boldsymbol{K}(k) = \hat{\boldsymbol{P}}^{(-)}(k)\,\boldsymbol{J}_{\mu}(k)\,\boldsymbol{S}^{-1}(k) \tag{3.14}$$

allows to compute the state update

$$\hat{\boldsymbol{p}}^{(+)}(k) = \hat{\boldsymbol{p}}^{(-)}(k) + \boldsymbol{K}(k)\boldsymbol{\kappa}(k) \tag{3.15}$$

and the covariance update

$$\hat{\boldsymbol{P}}^{(+)}(k) = (\boldsymbol{I} - \boldsymbol{K}(k)\,\boldsymbol{J}(k))\,\hat{\boldsymbol{P}}^{(-)}(k)\,. \tag{3.16}$$

3.2.2 Experimental Validation

This section shows experimental validation of the presented hyperbolic acoustic localization method. The setup consists of a receiver and four acoustic beacons as depicted in Fig. 3.3. A MEMS microphone (ADMP401) serves as the receiver in this setup. It provides an analog voltage to an A/D converter front-end circuitry, which consists of an active amplifier and a second-order high-pass filter. Phase-shifts introduced by band-pass filtering do not influence the localization algorithm because they apply to all received signals and cancel out when computing TDOAs. The software runs on a microcontroller board with 72 MHz Cortex M4 CPU, 64 KB RAM, 256 KB flash memory. The built-in A/D converter samples the receiver signals as 16 bit words with up to 400 kHz. The hardware is compact enough to fit into a μAUV.

The embedded software samples the acoustic signal, whereby one loop has a duration of t_{signal}. After sampling, the cross-correlation is computed for two consecutively received signals and its maximum is determined. For example, if the receiver has exactly the same distance to two beacons, the cross-correlation will indicate a signal shift of $m = 0\,\text{ms}$. This is equivalent to a vanishing TDOA. Precise loop timing is only required for always two consecutive loops, which is feasible with off-the-shelf RTCs. This is an advantage over other existing systems. However, latencies in the signal processing steps before cross-correlation should be very small because the signals do not contain time-stamps. Latencies of more than a few microseconds propagate into large position errors. Low latencies can be achieved with service interrupt routines in combination with a circular buffer and direct memory access.

The cross-correlation $r_{i,j}$ of two received consecutive signals s_i and s_j can be computed with fast Fourier transforms (FFTs) and inverse fast Fourier transforms (IFFTs) by

$$r_{i,j} = \text{IFFT}(\text{conj}(\text{FFT}(s_i) \cdot \text{FFT}(s_j))). \tag{3.17}$$

This is a memory and computationally intensive operation. We can accelerate the computation by exploiting the fact that the sequences s_i and s_j are real-valued. For this case the two FFTs can be computed with the cost of one FFT.

The robustness of the localization system depends on the chosen acoustic reference signal $s(t)$. In the system presented in this section, a time domain realization of wideband colored noise centered at 10 kHz is chosen. This signal offers a wide bandwidth and possesses a unique time shift, unlike sinusoidal signals. Chirp signals are another possible signal type. Second, the duration t_{signal} of the emitted signal must be taken into account to ensure robustness. The robustness scales with the number of sampling points P because more sampling points result in a more defined cross-correlation, and therefore better TDOA results. The amount of sampling points P can be linked to the time duration of the signal t_{signal} by $P = f\, t_{\text{signal}}$, where f is the sampling frequency. However, the signal duration t_{signal} depends on the application. For stationary applications, larger t_{signal} values can be chosen than for applications with fast moving receivers. The receiver's sampling frequency f is determined by the required spatial resolution. For a speed of sound c_{sound} and a sampling frequency f, the maximum round-off error is $e_{\text{r}} = \frac{c_{\text{sound}}}{f}$, which for $f = 50\,\text{kHz}$ corresponds to $e_{\text{r,air}} = 7\,\text{mm}$ in air and to $e_{\text{r,water}} = 35\,\text{mm}$ in water.

Experimental Results

A MEMS receiver is fixed to a two dimensional gantry, which is mounted on top of a test tank. Trajectories with small length scales $\mathcal{O}(0.1\,\text{m})$ are chosen to verify the accuracy of the localization system and the associated challenges. The speed of sound is the main physical parameter for position error magnitude, and it remains constant for all scales. In general, trajectories on larger length scales will yield similar absolute errors, resulting in smaller relative errors.

Experimental Setup

Four off-the-shelf, full range speakers are used in the experiments. Two of them are installed in two corners of the test tank. The other two speakers are installed so that the four speakers span a square of side length $l = 1.84\,\text{m}$. Experiments are performed in water and air. While the system was primarily developed for water application, it can also be used in other applications for GPS-denied localization outside of water. For experiments in air, the speakers are installed above the water surface, and for underwater experiments, the speakers are submerged 30 cm below the water surface with a water depth of 1 m. A MEMS receiver is routed by the gantry along a rectangular trajectory within the convex

hull spanned by the speakers. A colored noise time series of duration $t_{\text{signal}} = 75\,\text{ms}$ is generated from a JONSWAP (JOint North Sea WAve Project) spectrum with peak frequency at 10 kHz and is emitted by the speakers. Hence, every 75 ms, a new TDOA is available for the EKF to perform an update.

Experiments in Air

The sampling rate of the A/D converter is set to 47 kHz. The velocity of the receiver is chosen such that 2,700 TDOAs are captured in 180 s. Figure 3.4 illustrates the position estimation in the horizontal plane and Fig. 3.5 illustrates the corresponding TDOAs after TDOA rejection. The estimated trajectory is a rectangle, and most errors do not exceed 5 cm. Figure 3.5 exemplifies the required TDOA accuracy in order to track such a small scale trajectory. Most TDOAs vary between $-1\,\text{ms}$ and 1 ms and can be determined accurately by the system. Between 110 s and 120 s, many ill-measured TDOAs $\tau_{2,1}$ and $\tau_{1,4}$ are rejected. This explains the gap in the estimated trajectory between $x = 1\,\text{m}$ and $x = 0.8\,\text{m}$, $y = 1.1\,\text{m}$.

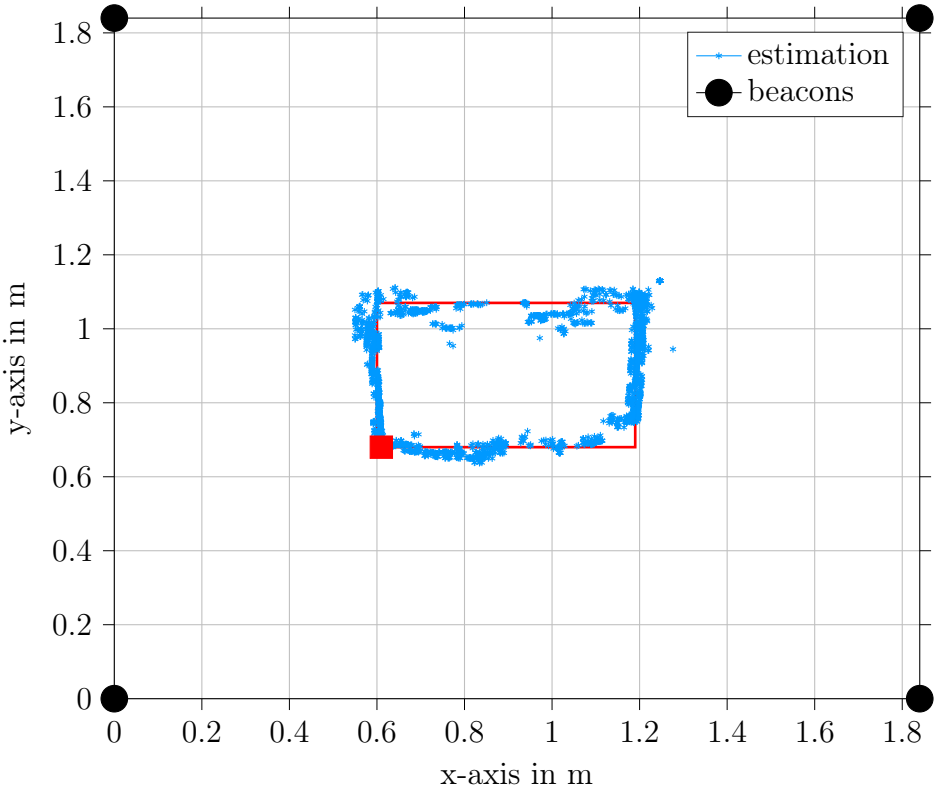


Figure 3.4: Position estimates in air (blue markers) are computed by an EKF from TDOAs, which are determined from acoustic measurements. The receiver moves counterclockwise (red line) with starting and end points depicted by the red square.

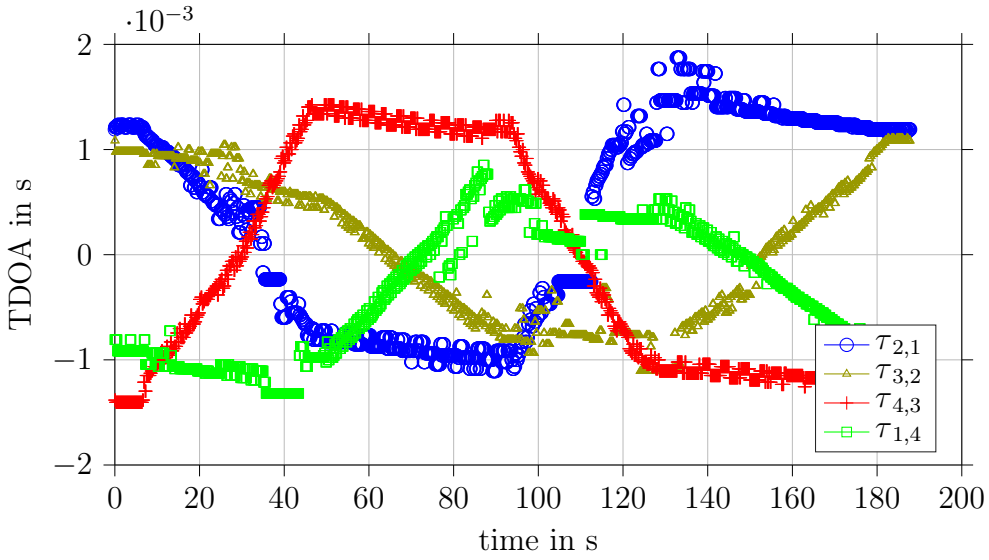


Figure 3.5: Captured TDOAs between the four consecutive speakers. The TDOAs are inputs to the EKF for position estimation.

Experiments in Water

For the experiments in water, the sampling rate of the A/D converter is increased to 117 kHz to account for the higher speed of sound in water. Fewer TDOAs are recorded, because the duration of the experiments is shorter. Figure 3.6 illustrates the estimated position of the receiver. The underwater experimental results show that the receiver position estimate tracks its actual position, within 10 cm error bounds in most cases. This is an acceptable performance for many applications in monitoring and exploration. This experiment shows that errors are larger in water than in air. The main factors contributing to the inferior performance of the in-water experiments is most likely reverberation in the confined tank. This effect can be minimized by using a larger tank, or by deploying blind dereverberation signal processing techniques [57].

3.3 EM-based Spherical Localization

Spherical localization requires range estimations between receiver and beacons. The attenuation of electro-magnetic waves in liquids can be used to determine the range. This section extends the localization framework introduced in [63, 64, 65, 66] to the μAUV domain. Beacons at known positions emit electro-magnetic waves of constant known power, and at unique frequencies. The receiver measures the superposed signal and determines the signal strengths at different frequencies.

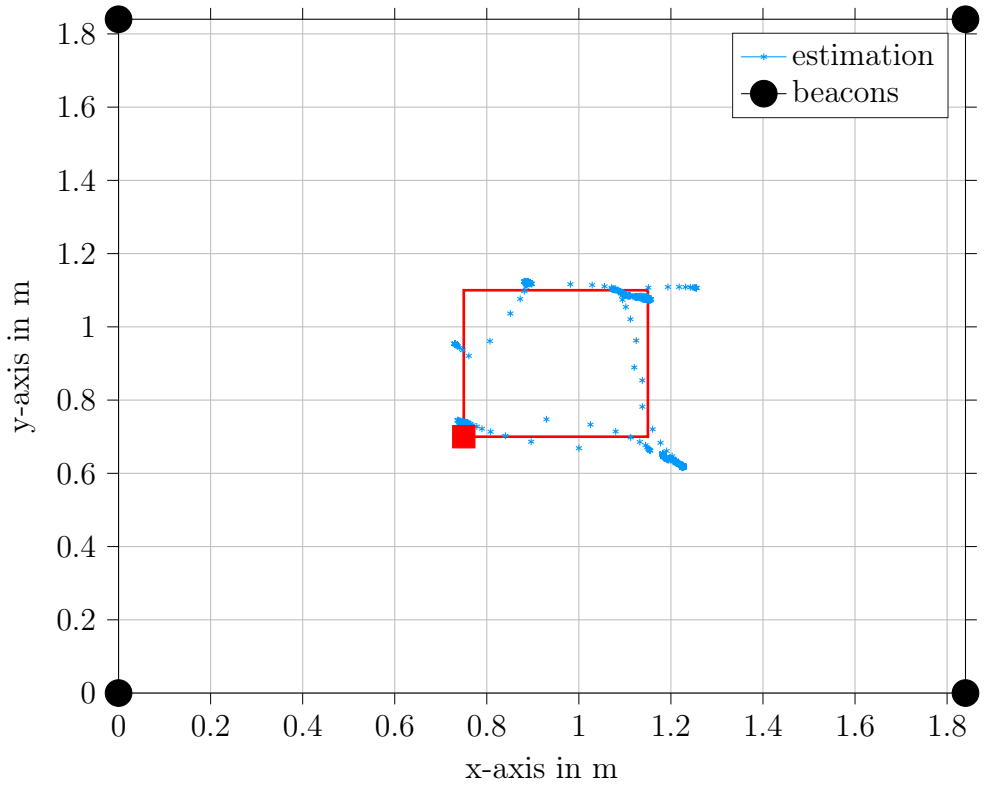


Figure 3.6: Position estimates in water (blue markers) are computed by an EKF from TDOAs, which are determined from acoustic measurements in a test tank at 30 cm water depth. The receiver moves counterclockwise (red line) with starting and end points depicted by the red square. For larger trajectories, the absolute error is expected to stay in the sub-decimeter range.

3.3.1 Underwater Range Sensor Model

The underwater range sensor model (URSM) describes the path loss attenuation as a function of distance [63]. The beacons transmit an EM signal of constant known power and the RSS is measured at the receiver position. Based on the RSS, the URSM provides a range estimation between receiver and beacons. In air, the attenuation rate of EM waves is small over distance. This leads to large fluctuations in the RSS due to reverberation and interference. In contrast, EM waves in water attenuate quickly with distance. Since disturbing effects such as reverberation or interference are less pronounced underwater, the principle of attenuating EM waves can be reliably used to estimate the distance from an EM transmitter to a receiver underwater.

In order to map RSS values to range estimates, several techniques have been proposed in literature. In air, the sensor model relies on the Friis transmission formula, which is commonly used to calculate ranges for a measured received signal strength. However,

this model needs to be modified to consider medium parameters such as temperature and conductivity. Park et al. [63, 66] developed an EM wave attenuation model for underwater environments. This model is a modified version of the Friis transmission formula which takes the attenuation constant α of the plane wave equation into account. The difference between the EM wave power on the receiver side S_{R} and on the transmitter side S_{T} is defined as RSS. The RSS as a function of the range R_i between the receiver unit and the i-th beacon reads

$$\mathrm{RSS}_i = S_{\mathrm{R},i} - S_{\mathrm{T},i} = -20\,\log_{10} R_i - 20\,R_i\,\alpha_i\,\log_{10}\mathrm{e} + \Gamma_i\,, \tag{3.18}$$

where Γ_i is an offset factor representing antenna and environmental influences. RSS, $S_{\mathrm{R},i}$ and $S_{\mathrm{T},i}$ are given in dBm The parameters α_i and Γ_i can be calculated explicitly if the polarization loss factor, the transmitting and receiving antenna gains, and the attenuation factor are known. In [63], these parameters are derived for an underwater test tank environment, and the resulting model is validated with experimental data. In this chapter, the model parameters are computed from spatially distributed RSS measurements by fitting (3.18) with non-linear least-squares.

3.3.2 Signal Identification using Channel Allocation

Localization within a horizontal plane requires at least three beacons. An important aspect in RSS-based spherical localization is that the receiver has to assign the RSS values to the respective emitting beacons. In principle, a receiver can identify the signal source by time scheduling [17]. However, this requires clock synchronization, which is usually a challenging task for underwater applications as discussed before.

An alternative approach for beacon identification is channel assignment, which is deployed in [64, 65, 66]. Here, each beacon sends an EM signal at a unique frequency. The resulting superposed signal is then measured by the receiver. By applying an FFT on the receiver side, each RSS can be allocated to the respective beacon. This technique increases the update rate significantly because beacons can transmit their signals simultaneously and do not need to wait for a scheduled time. Moreover, the FFT provides access to all RSS values in each cycle. This is shown in Fig. 3.7 for two beacons with different emitting frequencies. The prior knowledge about the beacons' frequencies allows the system to allocate the determined RSS values to the beacons. By using the URSM, the range between each beacon and the receiver can be determined.

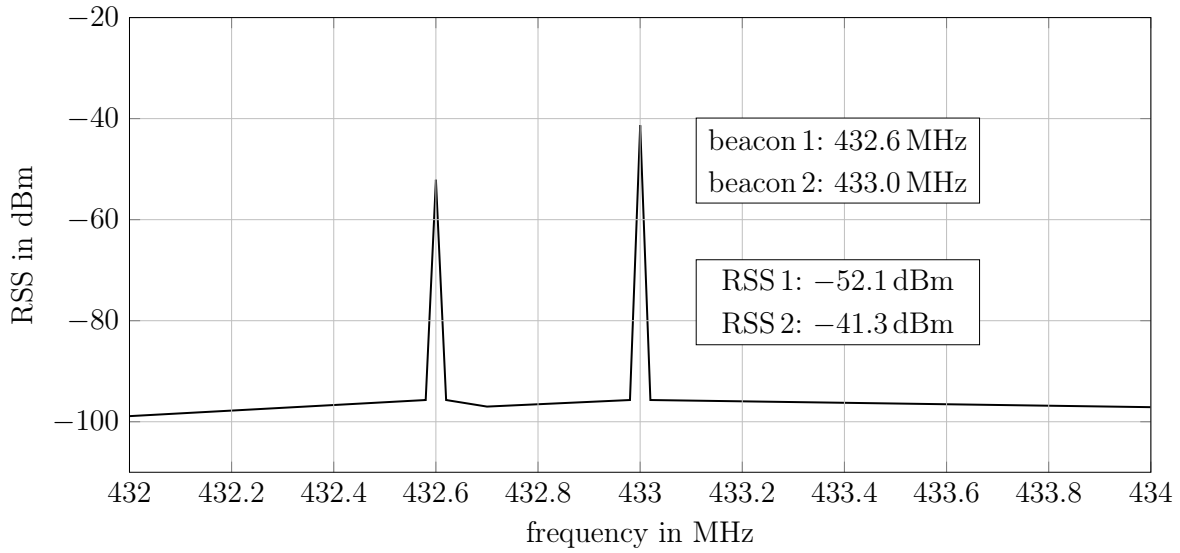


Figure 3.7: Identification of two beacons transmitting at 432.6 MHz and 433.0 MHz, respectively. The receiver unit identifies the beacons and corresponding RSS values by determining the frequencies and peak values.

3.3.3 Cramér-Rao Lower Bound

The mobile receiver unit computes its ranges to the beacons by using the measurements of the RSS values on different channels. The ranges are then fused to estimate the receiver's position, that is, with a Bayesian filter such as EKF or PF.

As a measure for the achievable accuracy of the localization system, the Cramér-Rao lower bound (CRLB) can be computed to express a lower bound on the variance of the estimated receiver position. The CRLB is the inverse Fisher information matrix $\boldsymbol{F}^{-1}$. In order to compute the CRLB, define a mean vector of received signal strengths

$$\bar{\boldsymbol{\tau}}_{\mathrm{RSS}} = [\mathrm{RSS}_1 \ \mathrm{RSS}_2 \ \cdots \ \mathrm{RSS}_N]^\top . \tag{3.19}$$

We assume the noisy signal strength measurements to be zero-mean Gaussian with covariance matrix $\boldsymbol{C} = \mathrm{diag}(\sigma_{\mathrm{RSS},i}(R_i))$. They are combined in the vector

$$\boldsymbol{\tau} \sim \mathcal{N}\left(\bar{\boldsymbol{\tau}}_{\mathrm{RSS}}, \boldsymbol{C}\right) . \tag{3.20}$$

The Fisher information matrix $\boldsymbol{F} \in \mathbb{R}^{n,m}$ has the elements

$$F_{m,n} = \frac{\partial \bar{\boldsymbol{\tau}}_{\mathrm{RSS}}^\top}{\partial p_m} \boldsymbol{C}^{-1} \frac{\partial \bar{\boldsymbol{\tau}}_{\mathrm{RSS}}}{\partial p_n} + \mathrm{tr}\left(\boldsymbol{C}^{-1} \frac{\partial \boldsymbol{C}}{\partial p_n} \boldsymbol{C}^{-1} \frac{\partial \boldsymbol{C}}{\partial p_m} \right) . \tag{3.21}$$

The two dimensional localization case yields $p_m = x$ and $p_n = y$.

The CRLB for a localization scenario with four emitting beacons are illustrated qualitatively in Fig. 3.8. Thereby, the covariance $\sigma_{\mathrm{RSS},i}(R_i)$ is modelled as a linearly increasing

function of the range to the i-th beacon. The CRLB indicate that the lowest position estimation covariance lies within the convex hull spanned by the beacons. Outside the convex hull, the variance of the localization estimate increases.

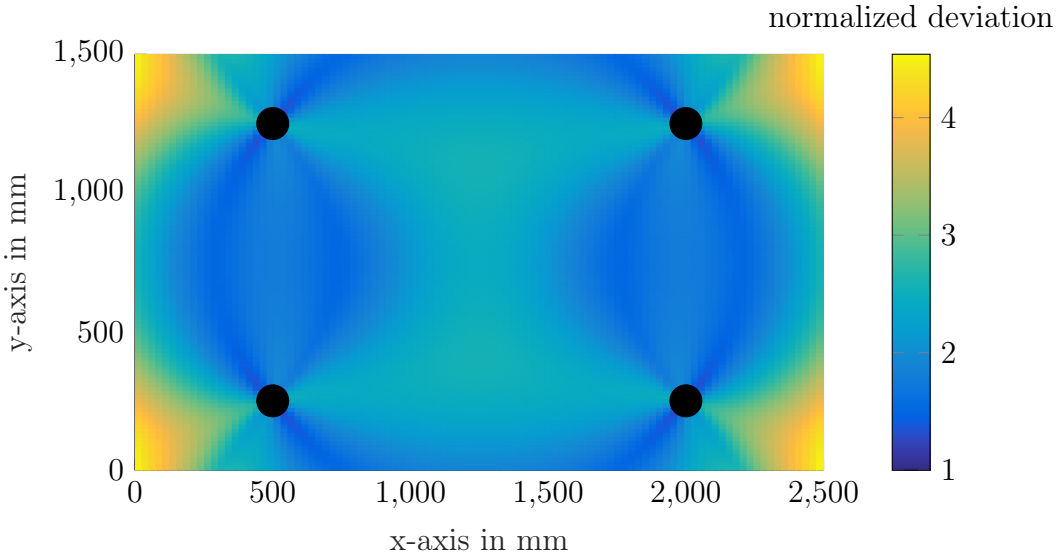


Figure 3.8: Normalized CRLBs plot for a linearly increasing measurement covariance function. Black markers depict the beacons.

3.3.4 Extended Kalman and Particle Filtering

Similar to hyperbolic acoustic localization of the previos section, the position vector $\boldsymbol{p}$ can be estimated with an EKF from the measurement vector $\boldsymbol{\tau}$. The equations (3.10) – (3.16) remain the same including the random walk model (3.7). The Jacobian differs, however. The non-linear measurement function $\bar{\boldsymbol{\tau}}_{\mathrm{RSS}}(\boldsymbol{p})$ (3.19) defines a vector of RSS mean values as functions of the receiver position. The components RSS_i introduced by the URSM in (3.18) are updated by RSS measurements (3.20). The Jacobian $\boldsymbol{J}_\mu(k)$ of $\bar{\boldsymbol{\tau}}_{\mathrm{RSS}}(\boldsymbol{p})$ consists of the derivatives $\boldsymbol{h}_i^\top(k)$ of (3.18):

$$\begin{aligned}\boldsymbol{h}_i^\top(k) &= \nabla_{\boldsymbol{p}} \mathrm{RSS}_i(\boldsymbol{p}(k)) \\ &= \left[-\frac{20}{\ln 10 \cdot \|\boldsymbol{p}(k) - \boldsymbol{r}_i\|^2} \cdot [\boldsymbol{p}(k) - \boldsymbol{r}_i]^\top - 20\,\alpha_i \,\log_{10} \mathrm{e}\, \frac{[\boldsymbol{p}(k) - \boldsymbol{r}_i]^\top}{\|\boldsymbol{p}(k) - \boldsymbol{r}_i\|} \right]_{\boldsymbol{p}=\hat{\boldsymbol{p}}^{(-)}(k)},\end{aligned} \tag{3.22}$$

$$\boldsymbol{J}_\mu(k) = \left[\boldsymbol{h}_1^\top(k) \;\cdots\; \boldsymbol{h}_N^\top(k) \right]^\top. \tag{3.23}$$

As an alternative to the EKF, a PF can be used to estimate the position from the measured signal strengths (3.20). Particle filters (also referred to as sequential Monte Carlo filters)

perform better than EKFs if the problem variance is large and the measurement model is highly non-linear [78]. The sampling importance resampling (SIR) PF [22] can be applied in the context of spherical localization.

The distribution of the receiver position is approximated by a particle set consisting of M particles $\{\boldsymbol{p}^{[m]}(k)\}$ indexed by the integer m. In addition, the particles are associated with weights $w^{[m]}(k)$, which indicate the importance of each particle. The weights are positive and are enforced to sum up to unity, that is, $\sum_{m=1}^{M} w^{[m]} = 1$.

At time step k, a temporary particle set $\{\tilde{\boldsymbol{p}}^{[m]}(k)\}$ is created from the particle set of the previous time step $k-1$ according to the random walk model

$$\tilde{\boldsymbol{p}}^{[m]}(k) = \boldsymbol{p}^{[m]}(k-1) + \boldsymbol{w}(k)\,. \tag{3.24}$$

The unnormalized weight $w_{\mathrm{u}}^{[m]}(k)$ is computed as the probability of the measurement $\boldsymbol{\tau}(k)$ for particle $\tilde{\boldsymbol{p}}^{[m]}(k)$:

$$w_{\mathrm{u}}^{[m]}(k) = \mathcal{N}(\boldsymbol{\tau}(k) \mid \bar{\boldsymbol{\tau}}_{\mathrm{RSS}}(\tilde{\boldsymbol{p}}^{[m]}(k)),\, \boldsymbol{C})\,. \tag{3.25}$$

Appendix A provides a brief overview of the fundamental probability concepts. In order to avoid particle degeneration, the temporary particle set is resampled. The weights w_{u} are normalized to sum up to unity, and all particles are drawn with replacement $\boldsymbol{p}^{[m]}(k) \sim \{\tilde{\boldsymbol{p}}^{[i]}(k)\}$ with probability $\propto w^{[i]}(k)$, $i = 1, \ldots, M$. The position estimate $\hat{\boldsymbol{p}}(k)$ can be obtained by averaging the resampled particles set

$$\hat{\boldsymbol{p}}(k) = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{p}^{[m]}(k)\,. \tag{3.26}$$

3.3.5 System Setup

The system consists of hardware components and embedded software. The beacons are anchored underwater and emit ambient EM signals. The signal generating unit of each beacon is a custom-made circuit board with a Radiometrix$^{\mathrm{TM}}$ USX2 multi-channel half-duplex UHF transceiver operating in the 433 MHz band. In order to spatially fix the beacon configuration, they are arranged as an array on a frame. This allows placing the localization systems at almost any desired position in the work space. Figure 3.9 shows a submerged array of four beacons and a detailed photo of a single beacon.

The mobile receiver unit consists of an underwater antenna, a modified DVB-T dongle capable of computing a power spectrum density, and a single board computer (SBC). The major design criteria for the module are size and cost, as it has to fit inside space constrained μAUVs such as HippoCampus.

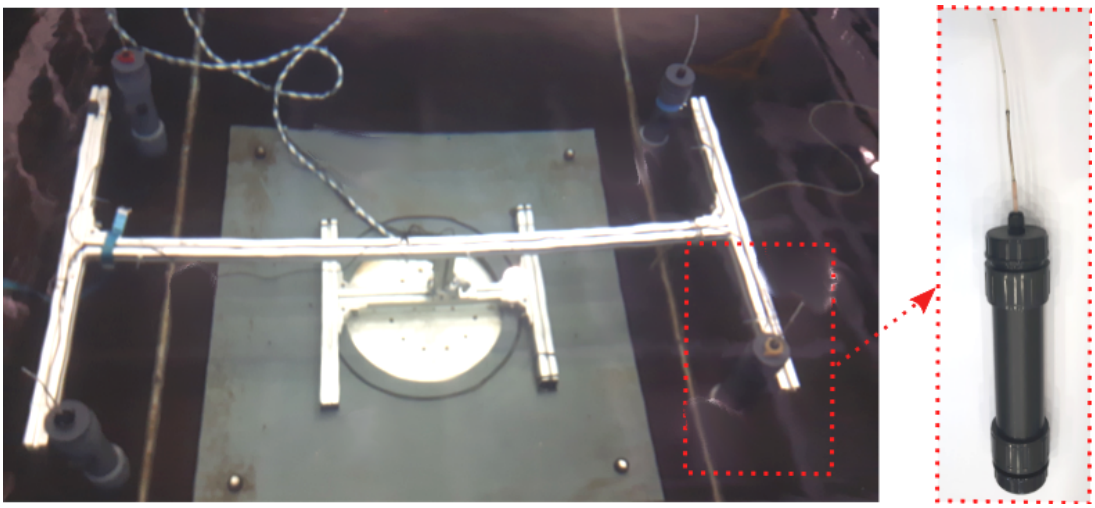


Figure 3.9: Submerged rack with four beacons. Each beacon continuously emits an EM signal at a unique frequency.

The mobile receiver unit carries out two main tasks. (1) It calculates real time RSS values based on the URSM. (2) It computes its position from the RSS values. The DVB-T dongle is used to digitize a segment of the EM spectrum as in-phase/quadrature (I/Q) samples. In order to drive the DVB-T dongle and to perform the RSS localization tasks, the SBC requests I/Q samples from the DVB-T dongle and computes the power density spectrum. The RSS values of the transmitter channels can be extracted from the spectrum and are used for range estimation.

Since the beacons transmit signals at different frequencies, a power density spectrum of the received signal has to be computed to determine the RSS values and to identify the corresponding beacons. In [63, 64, 65, 66, 36], a full-fledged spectrum analyzer is used, which is not deployable on μAUVs. In this section, a NooElec$^{\text{TM}}$ NESDR Mini DVB-T device is chosen to address the problem. It processes signal sequences received through an antenna within a range of 24−1,700 MHz. After demodulation and analog digital conversion (ADC), it transmits them via an USB interface to the SBC. The exploitation of the DVB-T dongle allows to bring EM-based localization to the μAUV domain. The core elements of the DVB-T dongle are the tuner and the demodulator. The integrated circuit tuner of the DVB-T dongle is an R820 chip. It receives analog EM signals, amplifies them, and performs bandpass filtering. Then, it downconverts the signal to a lower intermediate frequency (sub-sampling). This allows the subsequent 8 bit-ADC to sample at a much lower sampling rate than the carrier frequency of the incoming analog RF signal. The demodulator, a RTL2832U chip, contains the ADC and encodes the signal to I/Q samples via coded orthogonal frequency division multiplexing (COFDM). The I/Q samples

are then processed by the SBC for spectrum analysis and RSS estimation. The maximum sampling rate of the demodulator is 3.2 MS/s.

3.3.6 Experimental Validation

In this section, experimental results validating the feasibility and performance of the embedded RF localization system in water are presented. Localization results for static position hold and dynamic position estimation along trajectories are shown.

The experimental setup and the performed experiments are shown in Fig. 3.10. Four EM beacons are deployed in the experimental tank. In Fig. 3.10b, the dashed lines refer to the receiver unit trajectories and the stars refer to the static positions. The conductivity of the water amounts to approximately $0.031\,\frac{\mathrm{S}}{\mathrm{m}}$.

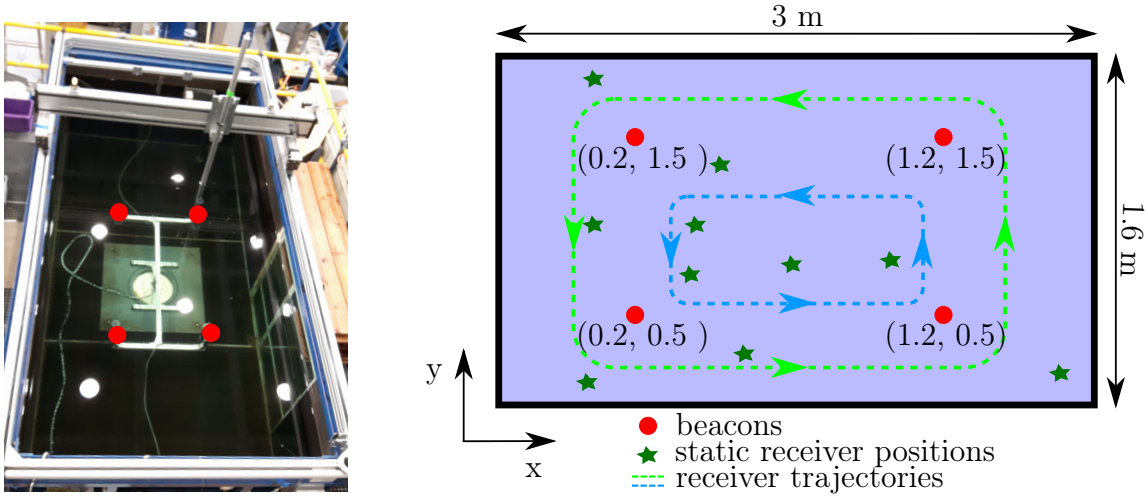
The four beacons transmit EM signals at 433.90 MHz, 434.05 MHz, 434.20 MHz, 434.35 MHz. The receiver unit is mounted to a horizontal movable gantry unit with a workspace of 3 m by 1.6 m. This allows moving the receiver unit along pre-programmed trajectories and take signal strength measurements with arbitrary high geometric resolution.

During static measurements, the localization system is able to reach sampling frequencies of 20 Hz. For dynamic measurements, the sampling frequency is approximately 4.5 Hz. This is mainly due to the long response time of the gantry position encoder. It is worth mentioning that the sampling frequency is independent of the number of emitting beacons, since the algorithm executes an FFT for the relevant part of the power density spectrum, including all transceiving frequencies of the beacons.

The RSS data can be used to validate the URSM (3.18) introduced by [63]. The measured data presented in Fig. 3.11 is used to fit α_i and Γ_i of (3.18). Figure 3.11 illustrates the measured RSS values as a function of distance for a single frequency and the fitted URSM. The data corresponds to the analytical model, whereby the scattering increases with distance. Moreover, the measurements deviate from the URSM in direct vicinity of the emitting beacon. This effect is due to a vertical offset between the beacons' horizontal transmission plane and the receiver node.

Static Position Estimation

For the static position estimation experiment, ten positions with known ground truths are chosen. The receiver unit is placed at those positions, and the receiver positions are estimated with the localization system. Figure 3.12 shows the experimental results for



(a) Photo of experimental tank. (b) Schematic of experiments.

Figure 3.10: Experimental setup to validate the embedded EM localization system.

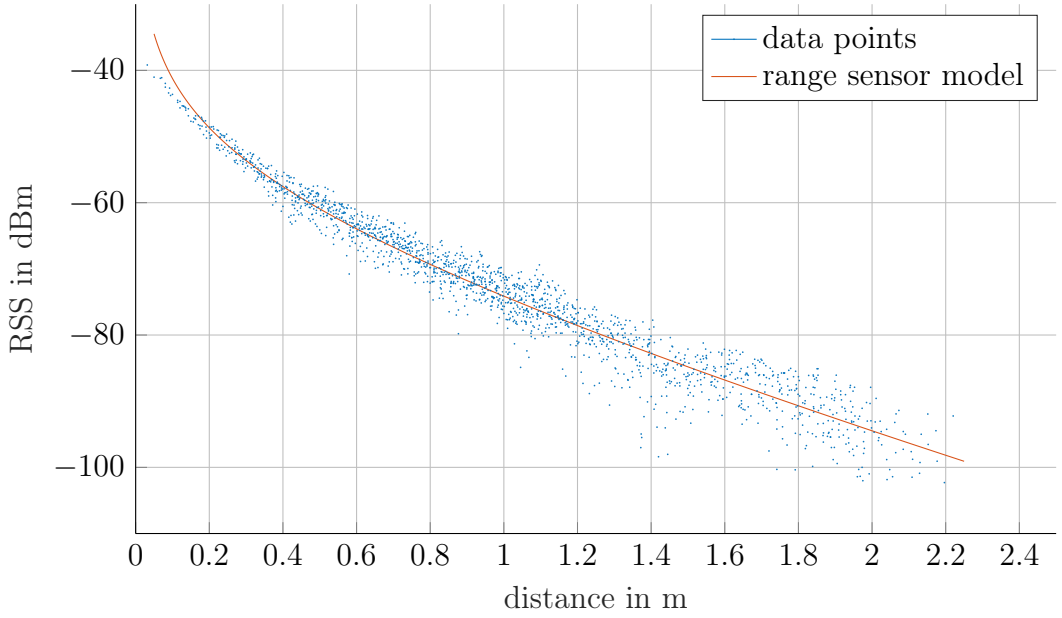


Figure 3.11: Underwater range sensor model and RSS measurements.

those positions, where the estimation is performed with a PF and $M = 1,000$ particles. While Park et al. [64] report root-mean-square errors between 1 mm and 2 mm for their original method, the errors in Fig. 3.12 are an order of magnitude higher. This is because data processing capabilities of the DVB-T dongle are inferior to the analyzing hardware in [64]. Another reason for the degraded performance are reflections in the experimental tank. The points in proximity to the tank walls (points 1, 2, 8, and 9) show systematic biases due to reflections, whereas results for points further away (points 3 and 4) are more accurate. Despite their biases, all measurements show rather small variances. This justifies the assumption that the deviations mainly result from the reflections and are thus caused by the characteristics of the small test basin.

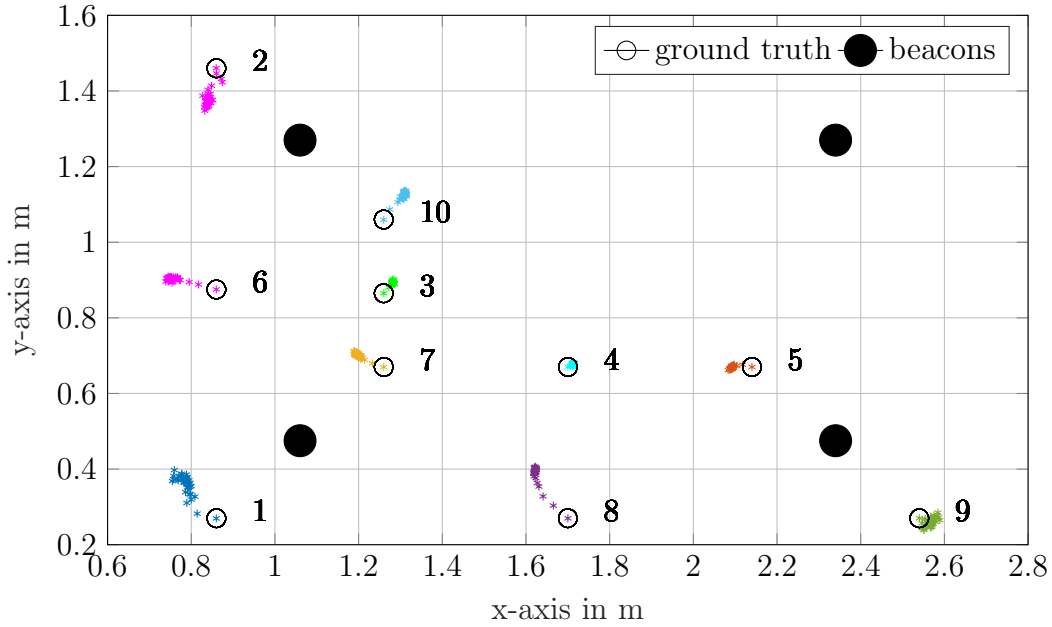


Figure 3.12: Static receiver self-localization at ten different positions. A PF with 1,000 particles merges the measurements.

Dynamic Position Estimation

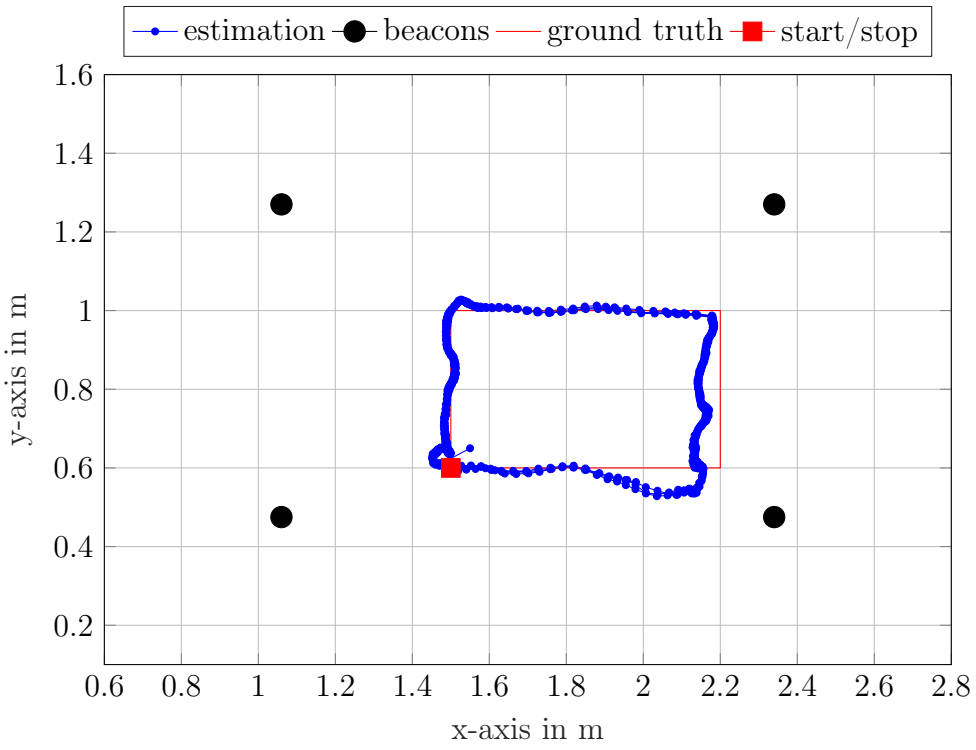
In the final experiment, the receiver unit is moved along two different rectangular trajectories similar to the one in [64], albeit smaller. The first trajectory lies completely in the convex hull spanned by the beacons. The second trajectory lies outside of the convex hull. The results for the two trajectories are shown in Fig. 3.13 and Fig. 3.14. For each trajectory, an EKF and a PF are used to fuse the RSS values for position estimation. The ground truths and the estimated positions of the receiver are shown in Fig. 3.13a and Fig. 3.13b for the inner rectangle and in Fig. 3.14a and Fig. 3.14b for the outer rectangle. The RSS measurements at the four distinct frequencies are illustrated in Fig. 3.15a and

Fig. 3.15b. As detailed above, the results are less accurate than the ones reported in [64]. However, for most μAUV applications in confined test tanks, these localization errors are acceptable, especially given the compact size and cost of the system.

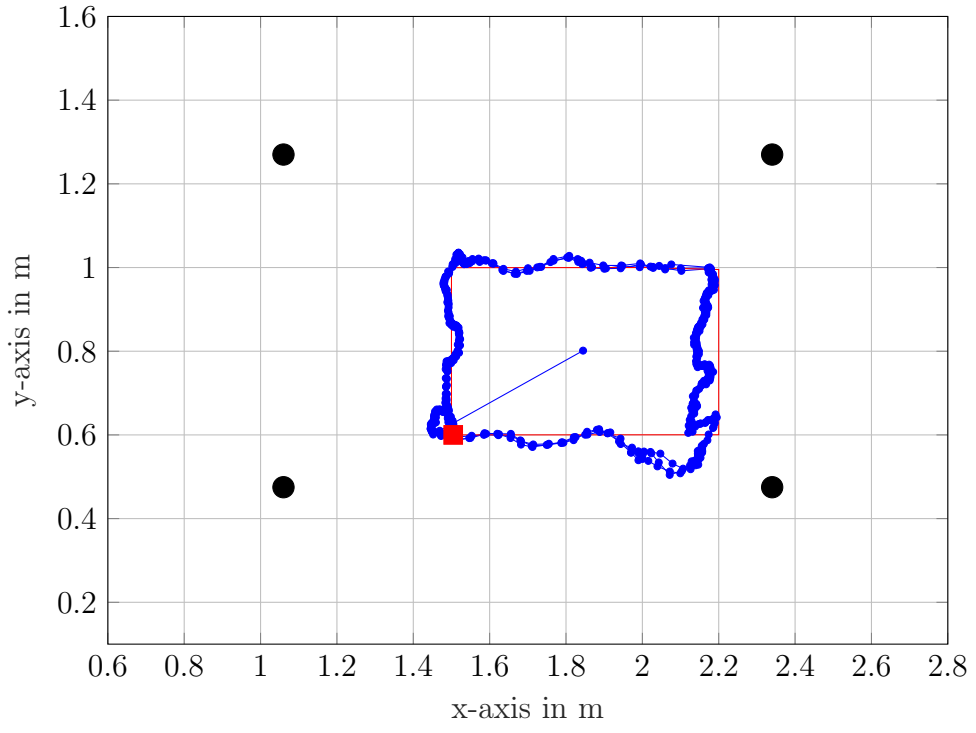
The estimated positions are compared against ground truth, and the RSS measurements are assigned to corresponding gantry positions. The RSS values in Fig. 3.15a and Fig. 3.15b exhibit discontinuities at time steps 100 and 300, respectively. At these points, the gantry did not provide ground truth data, and the RSS measurements were rejected for several consecutive time steps. Nevertheless, the system was able to recover the position as soon as the RSS measurements became available again. This demonstrates the robustness of the system.

3.4 Summary

This chapter covered the underwater self-localization problem that has to be solved for the deployment of μAUVs. Two approaches were introduced that are suitable for μAUV self-localization. One is TDOA-based hyperbolic localization, and the other is EM attenuation-based spherical localization. Compared to state-of-the-art a relatively high accuracy of up to 10 cm was achieved with the acoustic hyperbolic method. The second method based on the attenuation of EM signals yielded even higher positioning accuracy. The presented approaches are based on one-way signal transmission from anchored beacons to receiver. This allows for scalability to large fleets because the localization results are not affected by the number of receivers. For both methods hardware implementations were presented, which are low cost and compact in size. Experimental results demonstrated the feasibility of both localization concepts.

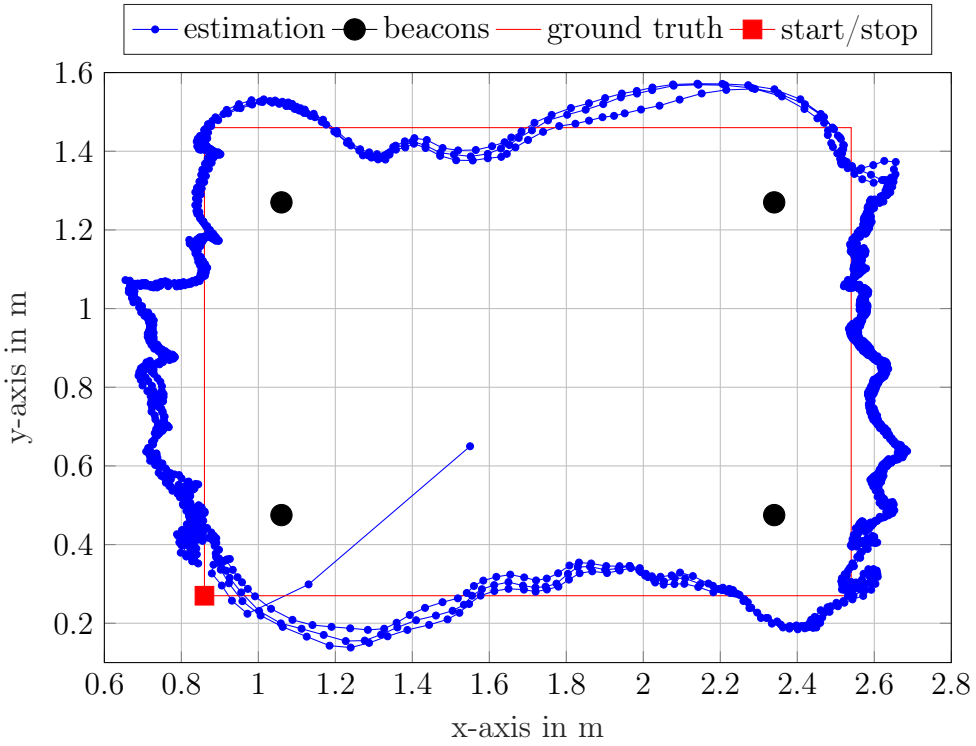


(a) Sensor fusion via EKF.

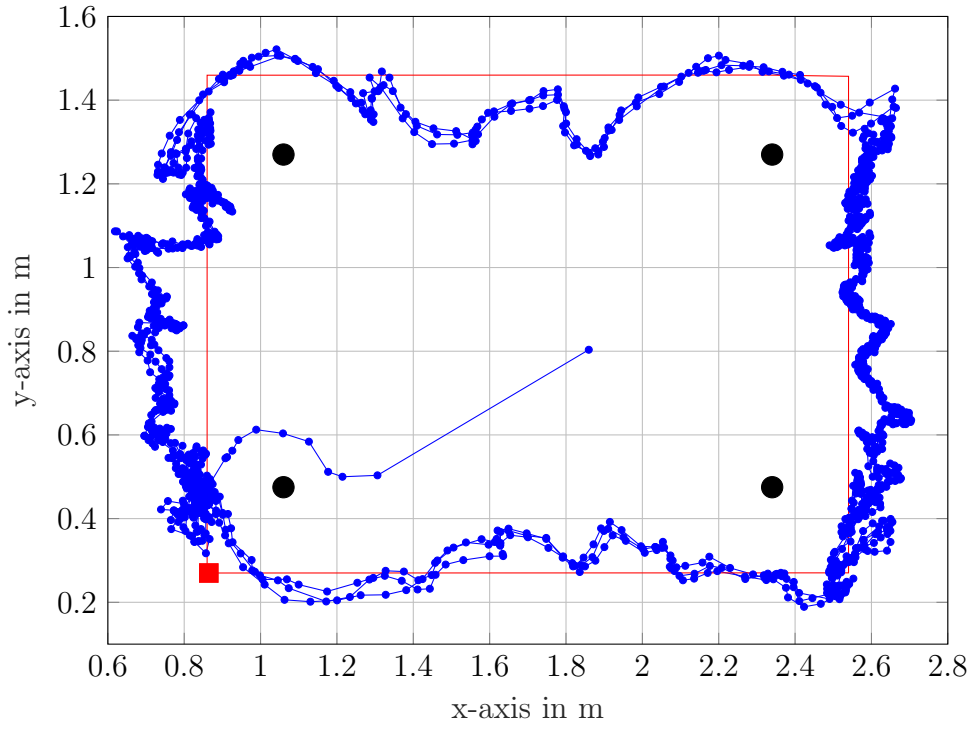


(b) Sensor fusion via PF.

Figure 3.13: Self-localization results along a rectangular trajectory within the convex hull of the beacons. The receiver traverses the rectangle three times.

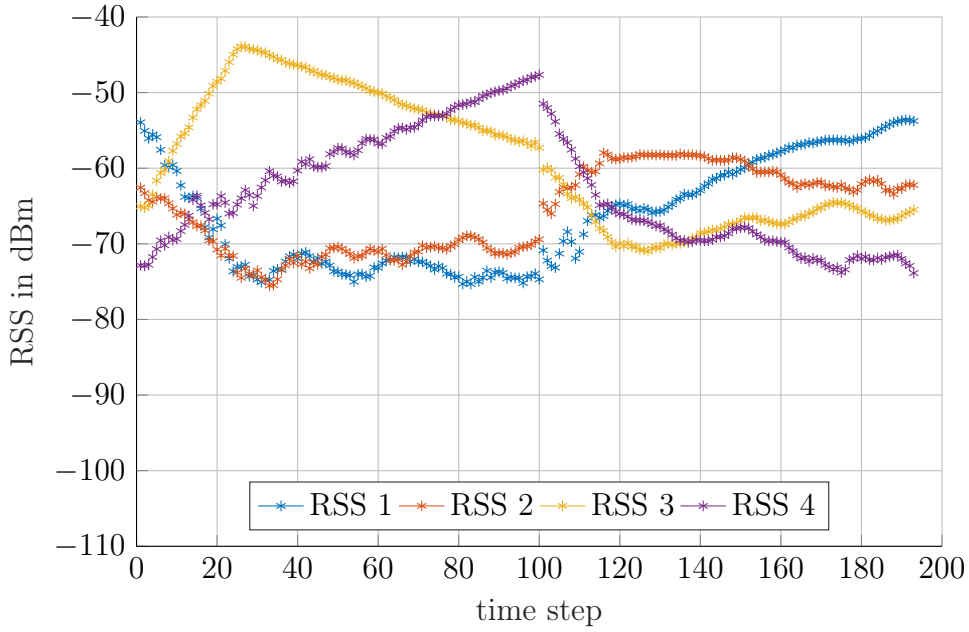


(a) Sensor fusion via EKF.

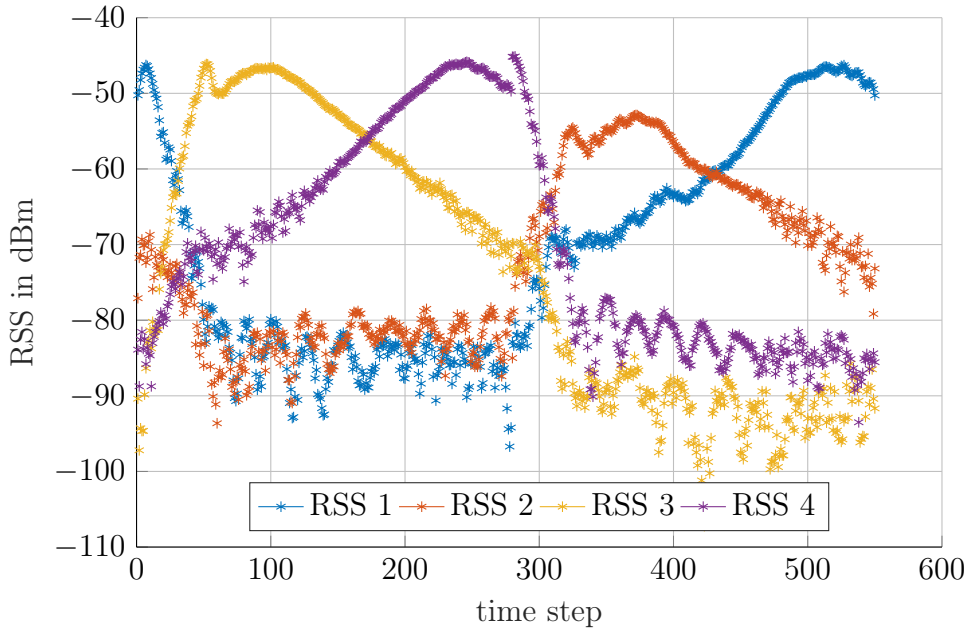


(b) Sensor fusion via PF.

Figure 3.14: Self-localization results along a rectangular trajectory outside the convex hull of the beacons. The receiver traverses the rectangle three times.



(a) Within the convex hull.



(b) Outside the convex hull.

Figure 3.15: RSS measurements at four frequencies for one turn of the receiver within and outside the convex hull spanned by the beacons.

Chapter 4 PI-GMRF Control for Field Exploration

This chapter proposes a high-level guidance control framework that allows μAUV fleets to autonomously explore unknown environmental fields. The μAUVs seek information to reduce uncertainty in their environmental fields knowledge. The fields of interest are scalar concentration fields or flow fields. Flow fields interfere with the motion of the μAUVs, whereas scalar concentration fields do not. The control laws that are introduced in this chapter can be used to either directly compute control sequences or to provide desired paths. For desired paths, the μAUVs can follow the paths with the low-level controllers that were introduced in Chapter 2. The chapter proceeds with the problem formulation and a solution approach based on stochastic optimal control. The stochastic optimal control problem is solved by means of a path-integral (PI) formulation, whereby a Gaussian Markov random field (GMRF) is deployed as a belief representation within the controller. The performance of the resulting path integral Gaussian Markov random field (PI-GMRF) controller is demonstrated in two simulation case studies.

4.1 Environmental Field Exploration Problem

Deploying μAUVs for autonomous information gathering requires an exploratory controller. The first step for developing control laws is to formulate the control problem, which follows from the overall operational objective. The problem of field exploration can be stated within the information theoretic control framework, which covers applications where information is directly coupled with motion.

4.1.1 System Definition and Problem Formulation

Consider n_s μAUVs. The i-th μAUV has the state $\boldsymbol{\xi}^i$ which describes its position, velocity, orientation, and angular velocity in a global coordinate system in accordance to (2.2) – (2.4) in Chapter 2. The equations of motion are restated here similarly to (2.26), however with flow influence. The motion of each μAUV is governed by

$$\dot{\boldsymbol{\xi}}^i = \boldsymbol{f}(\boldsymbol{\xi}^i) + \boldsymbol{\chi}(\boldsymbol{\xi}^i) + \tilde{\boldsymbol{u}}^i \,. \tag{4.1}$$

The function $\boldsymbol{f}$ describes the state evolution and can represent models of different fidelity. For path and trajectory planning, models based on particle dynamics are sufficient. However, rigid body dynamics are better suitable for the direct computation of control actions. The flow field $\boldsymbol{\chi}$ is assumed to be an additive force and acts as a disturbance on the μAUV motion. The function $\tilde{\boldsymbol{u}}^i$ is the scaled control input and allows each μAUV's control. A control law defines the control input $\tilde{\boldsymbol{u}}^i$. The μAUVs take noise corrupted measurements $\boldsymbol{z}^i$ of the scalar and the flow fields $\mathcal{F}_\mathrm{real}$ at their respective positions. The object $\mathcal{F}_\mathrm{real}$ represents the environmental fields and maps positions in the domain of interest to field values. The measurement model reads

$$\boldsymbol{z}^i \sim \boldsymbol{g}(\mathcal{F}_\mathrm{real}(\boldsymbol{\xi}^i), \boldsymbol{v}^i) \,. \tag{4.2}$$

The measurement noise $\boldsymbol{v}^i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_z)$ is Gaussian zero-mean and uncorrelated. The μAUV fleet collects measurements to infer the states of the fields. The estimate of the true fields is denoted by $\mathcal{F}$ which represents the scalar concentration field and the flow field.

The required behavior of the μAUVs consists of environmental observation, reasoning, and acting upon the available information. An embedded representation of the environment is introduced to this perception-action loop and is called *belief*. The idea is to allow the μAUVs to utilize the belief for controls. A belief is the *imagined* state of the environment and has to be derived from measurements. Belief representations can coincide with the field estimate $\mathcal{F}$, but they do not need to. The belief and the field estimate $\mathcal{F}$ fulfill different purposes. In information gathering problems, belief representations are maintained for closing the information loop in adaptive control. Hence, second order modes of the belief, i. e. the covariance, are very important because they are associated with uncertainty. Computational tractability is crucial for belief representations even at the expense of accuracy because control actions need to be derived on-line. The field estimate $\mathcal{F}$, however, is often associated with the overall mission goal such as obtaining an accurate representation of a concentration field, for example. Therefore, the field estimate only needs to represent mean field values (first modes). Also, field estimate computations can be performed off-board and even off-line after all measurements have been collected. Different modelling

approaches can be chosen for belief and field estimate. Fast models are preferred for the belief representation to allow online control. Computationally expensive models, such as physics-based models provide high-fidelity estimates of the fields of interest, but are often not feasible for on-line use. This chapter covers control laws for μAUV autonomy, with a focus on embedded belief representations of fields. Field estimates are simply assumed to be identical with the belief. Both, field estimates and belief are denoted by $\mathcal{F}$. The terms belief and estimate are used interchangeably in the remainder of this chapter.

The desired high-level behavior of a μAUV fleet consists of a continuous collection of field measurements. They are used to update the internal belief $\mathcal{F}$. Simultaneously, the belief benefits the derivation of control sequences that in turn lead to a further belief improvement. Figure 4.1 illustrates the decomposition of this exploratory behavior. The

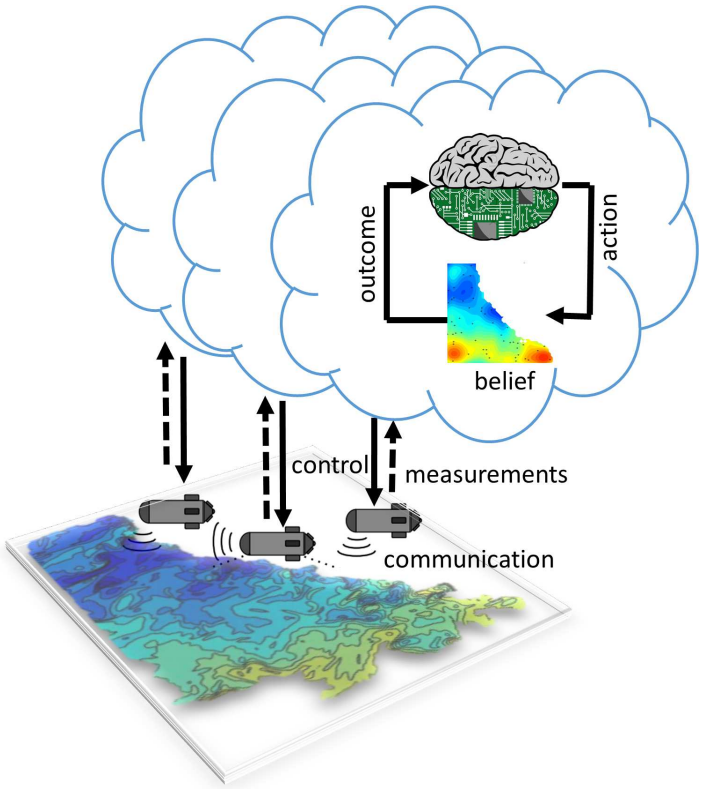


Figure 4.1: A μAUV fleet's high-level behavior in an environmental exploration mission. The μAUVs observe the state of environmental fields. The measurements are used to construct beliefs of the fields. The beliefs are required for the information theoretic control-loop. The goal is to derive control sequences that ensure a continuous improvement of the belief. Communication links among μAUVs allow for information exchange.

μAUVs are deployed in an environmental field where they can move, take measurements, and communicate with each other. Each μAUV maintains a belief of the environment and continuously updates with the collected measurements. A control loop runs on each

μAUV and evaluates the current belief to derive the control sequences. Note that each μAUV maintains its own on-board control loop instance, and the outlined approach does not require a central coordination unit. Communication links among μAUVs ensure information exchange. The exchanged information depends on the mission objective and might, for example, consist of field measurements.

4.1.2 Belief Representation

The previous subsection proposed to embed a belief representation of the environment within the controller. This subsection elaborates on possible mathematical formulations of such belief representations. Functions for belief representations need to fulfill two requirements. First, they need to represent distributions of spatial fields that can be updated by measurements. Second, they need to be embeddable into a controller structure.

Since exploration tasks require reasoning about uncertainties, an intuitive choice for beliefs would be a stochastic formulation. Stochastic models offer several distinct advantages in this application. Probability theory allows for inversion of measurements and model by computing conditional probabilities of the estimate given the measurements. This is regarded as a general inference scheme for reasoning, which is likely also present in biological decision making [20]. Note that the choice for a probabilistic framework does not imply that the environmental fields are inherently stochastic. In other words, their states do not randomly fluctuate. However, a probabilistic framework is chosen because the uncertainty is epistemic, and a probabilistic belief can capture a variety of possible field states at the same time.

Random fields are the stochastic extension of deterministic fields and can be interpreted as probability distributions of spatial functions. They are solutions of stochastic partial differential equations, which are currently one of the most actively studied topics in mathematical sciences. Random fields can be further subclassified into Markov random fields and these into GMRFs. Gaussian Markov random fields are of interest for spatial environmental field modeling and belief representation. Recent advances in stochastic field modeling motivate this choice because GMRFs allow for fast and accurate inference, even for very large measurement data sets. They also allow for a principled treatment of non-stationary fields, which is an active research topic in statistics [16].

The belief is stated as a GMRF:

$$\mathcal{F} = \mathrm{GMRF}(\boldsymbol{\mu}, \boldsymbol{Q}^{-1})\,. \tag{4.3}$$

A GMRF is parametrized by its field mean $\boldsymbol{\mu}$ and its precision matrix $\boldsymbol{Q}$. The inverse of a stochastic field's precision matrix is the process covariance matrix.

Random fields can be interpreted as a collection of an infinite number of spatial functions each with a different realization probability. Measurements collected by μAUVs update the GMRF distribution, such that field elements change their probability within the distribution. Therefore, certain measurements are intuitively more useful than others for updating the distribution. In order to explore environmental fields efficiently, control sequences need to yield informative measurements. The information quality (or sensing quality) of measurements can be formalized to assess their goodness. Sensing quality can be quantified with various information theoretic metrics [34]. In this chapter, predictive variance conditioned on measurements is chosen as a sensing quality metric. This criterion is often used in spatial statistics and is also called Bayesian A-optimality. The predictive variance of the field belief is evaluated at reference points $\boldsymbol{s}_s$:

$$\operatorname{Var}\left[\mathcal{F}(\boldsymbol{s}_s) \mid \boldsymbol{z}\right] = \sum_s \mathbb{E}\left[(\mathcal{F}(\boldsymbol{s}_s) - \mathbb{E}[\mathcal{F}(\boldsymbol{s}_s) \mid \boldsymbol{z}])^2 \mid \boldsymbol{z}\right] , \tag{4.4}$$

with $\mathbb{E}[\cdot]$ being the expectation operator. Another popular choice for assessing the sensing quality is the mutual information criterion, which is also known as Bayesian D-optimality. Instead of evaluating the predictive variance at reference points, the mutual information criterion assesses the conditional entropy.

The control objective is to find μAUV control sequences resulting in measurements that decrease the GMRF belief's predictive variance. In the next subsection, the control problem is formulated and solved. The solution provides control sequences for environmental data gathering.

4.1.3 Stochastic Optimal Control

The belief has been formulated as a random field that represents a distribution of spatial fields. Stochastic optimal control provides a principled and structured framework to include probabilistic elements. It is also a powerful method for controlling dynamical systems. Here, stochastic optimal control is predestined for formalizing data gathering.

Consider a finite horizon cost function J for a path $\boldsymbol{\tau}_j^i \equiv \boldsymbol{\xi}_{j:N}^i$ of the i-th μAUV at time t_j. The path $\boldsymbol{\tau}_j^i$ begins at time t_j from state $\boldsymbol{\xi}^i$ and ends at time t_{j+N} with a control horizon length N. The cost along the path $\boldsymbol{\tau}_j^i$ can be computed as

$$J(\boldsymbol{\tau}_j^i) = \mathop{\mathbb{E}}_{\boldsymbol{\tau}_j} \int_{t_j}^{t_{j+N}} \mathcal{L}^i(q^i(\boldsymbol{\xi}), \tilde{\boldsymbol{u}}) \, \mathrm{d}t \, . \tag{4.5}$$

The expectation $\mathbb{E}$ is taken over the states of all μAUVs to enforce coordination. Otherwise, μAUVs might choose similar redundant paths. The immediate cost $\mathcal{L}^i(q^i(\boldsymbol{\xi}), \tilde{\boldsymbol{u}})$ includes the state cost of the i-th vehicle $q^i(\boldsymbol{\xi})$ and a control cost, which is a function of the

control signal $\tilde{\boldsymbol{u}}$. The control cost will be analyzed in the next section. The state cost $q^i(\boldsymbol{\xi})$ depends on the i-th vehicle's own state and also on the states of other vehicles, which are denoted by $\boldsymbol{\xi}^i$ and $\boldsymbol{\xi}^{-i}$, respectively. Different formulations of the state cost are possible. The additive approach

$$q^i(\boldsymbol{\xi}) = q^i(\boldsymbol{\xi}^i, \boldsymbol{\xi}^{-i}) = q^i_{\text{field}}(\boldsymbol{\xi}^i) + q^i_{\text{coop}}(\boldsymbol{\xi}^{-i}) \tag{4.6}$$

is chosen. The predictive variance (4.4) is included in the field state cost $q^i_{\text{field}}(\boldsymbol{\xi}^i)$, which is defined as

$$q^i_{\text{field}}(\boldsymbol{\xi}^i) = \text{Var}\left[\mathcal{F}_j(\boldsymbol{\xi}^i) \mid \boldsymbol{z}_{0:j}\right] . \tag{4.7}$$

At time t_j, the field beliefs $\mathcal{F}_j$ are derived from all available measurements $z_{0:j}$ obtained until t_j. The predictive covariance can be evaluated in the belief space along the receding horizon, where the reference points in (4.4) are subsets of the μAUVs' expected paths. As stated in Chapter 1, information theoretic problems concerned with optimal data gathering are in general provably intractable, and the problem always needs to be simplified, which renders any solution suboptimal. Here, the simplification is that the field state cost (4.7) does not include an expectation over future measurements, which keeps the computational complexity tractable. In the next section, the control sequences will be computed in a receding horizon scheme. Since the problem is solved at every time step, the neglect of the expectation over future measurements is tolerated.

The cost $q^i_{\text{coop}}(\boldsymbol{\xi}^{-i})$ handles the coordination of the i-th μAUV with other fleet members. It ensures that the μAUVs do not pursue similar paths. Each μAUV considers the states of other μAUVs through q_{coop}. The penalty in the state cost is formulated as

$$q^i_{\text{coop}}(\boldsymbol{\xi}^{-i}) = \sum_{i \neq j}^{n_s} \frac{b_{\text{coop}}}{\left\|\boldsymbol{\xi}^i - \boldsymbol{\xi}^j\right\|} . \tag{4.8}$$

The parameter b_{coop} is an adjustable tuning parameter. It determines the repulsion between the μAUVs. The cost term (4.8) leads to a behavior where μAUVs show coordination, which is a desired behavior in exploration.

At time t_j, the optimal control can be obtained by minimizing the finite horizon cost function (4.5).

4.2 Path Integral Control for Exploration

The previous section stated the problem of field exploration as a stochastic optimal control problem. The information theoretic cost function (4.5) has to be optimized under the

consideration of system dynamics. The optimization leads to a control sequence for informative field measurements. Although stochastic optimal control is a powerful framework which can be applied to many control problems, it is often impossible to find computationally tractable solutions. Solutions for only few special cases have been reported in literature [59]. These cases are usually linear in their dynamics or restrictive in cost function design. A well known stochastic optimal control solution for linear systems is the linear quadratic Gaussian regulator. In recent years, significant progress has been made in efficiently solving certain types of nonlinear stochastic optimal control problems [30, 31]. By exploiting a log-transformation of the value function, the solution of certain stochastic optimal control problems can be stated as a path integral [32]. Path integrals have been studied in the context of quantum mechanics and can be (approximately) solved through forward sampling. Compared to other sampling-based methods, such as the rapidly exploring random tree family, path integral control has various advantages. It is derived from first principles of stochastic optimal control, it includes system dynamics seamlessly, and it has only few tuneable parameters. The path integral approach has been proven to be suitable for combining kinodynamic models with machine learning tasks [76]. It was applied to problems such as formation control for quadrotors [21], aggressive driving with ground vehicles [81], and learning to control an inverted pendulum [62]. All these applications of path integral control are feasible for embedded systems and provide control actions in real-time.

This section extends the path integral control formulation to spatial field exploration problems. It presents the derivation of an exploratory control law which solves the stochastic optimal problem introduced in the previous section. The controller considers the μAUV dynamics (4.1) and minimizes the finite horizon cost function (4.5). It also reduces the optimization of the stochastic optimal control problem to a path integral computation.

As a notational convention, we will index variables (e. g. path, state, control action, etc.) starting at time t_j and ending at time t_{j+N} by $(\cdot)_{j:N}$. A variable centered around time t_m is indexed by $(\cdot)_m$. The only exception is the notion of the path, where $\boldsymbol{\tau}_j \equiv \boldsymbol{\xi}_{j:N}$. In order to derive the PI-GMRF control law, the ansatz for the control function $\tilde{\boldsymbol{u}}^i$ in (4.1) is chosen as

$$\tilde{\boldsymbol{u}}^i = \boldsymbol{G}(\boldsymbol{u}^i + \boldsymbol{\epsilon}^i)\,. \tag{4.9}$$

The control affine form is a prerequisite for the path integral reformulation [76]. We introduce a parametrized control vector $\boldsymbol{u}^i$ whose dimensionality is possibly much larger than the dimensionality of $\tilde{\boldsymbol{u}}^i$. Zero-mean Gaussian noise $\boldsymbol{\epsilon}^i$ with variance $\boldsymbol{\Sigma}_\epsilon$ is added to $\boldsymbol{u}^i$ to allow for adaption in the control space. The sum is multiplied with the control transition matrix $\boldsymbol{G}$ to obtain the desired dimensionality of $\tilde{\boldsymbol{u}}^i$. The control form (4.9) allows for a great flexibility, because the derivation of the control law is now extended to

a higher dimensional space, where $\boldsymbol{u}^i$ lives. The choice of $\boldsymbol{G}$ is not restricted. Inserting (4.9) into (4.1) results in

$$\dot{\boldsymbol{\xi}}^i = \boldsymbol{f}(\boldsymbol{\xi}^i) + \boldsymbol{\chi}(\boldsymbol{\xi}^i) + \boldsymbol{G}(\boldsymbol{u}^i + \boldsymbol{\epsilon}^i)\,. \tag{4.10}$$

In the previous section, the state-cost q^i in the immediate cost $\mathcal{L}^i(\boldsymbol{\xi}, \tilde{\boldsymbol{u}})$ was defined by (4.6). However, the definition of the control dependent cost was postponed until now. Note that in (4.9), only $\boldsymbol{u}^i$ determines the applied control effort and hence should be considered in the immediate cost instead of $\tilde{\boldsymbol{u}}$. A quadratic control cost with a semi-definite weight matrix $\boldsymbol{R}$ is chosen to penalize large control efforts. The final immediate cost of the problem becomes

$$\mathcal{L}^i(\boldsymbol{\xi}, \boldsymbol{u}) = q^i(\boldsymbol{\xi}) + \frac{1}{2}\boldsymbol{u}^{i\top}\boldsymbol{R}\boldsymbol{u}^i\,. \tag{4.11}$$

4.2.1 Path Integral Control Formulation

This subsection derives the path integral control formulation of the exploration problem. The derivation uses results from [77], where a class of nonlinear stochastic optimal control problems was solved based on the stochastic Hamilton-Jacobi-Bellman (HJB) equation. An alternative analysis of path integral control laws without the stochastic HJB equation is presented in [81]. The following derivation is from the perspective of the i-th μAUV, where the subscript i is omitted for clarity. As in the previous section, we consider the control problem at arbitrary time t_j with a control horizon N. At time t_j, the considered μAUV maintains a corresponding belief representation of the underlying fields based on the measurements available at time t_j.

The first step in solving optimal control problems is defining a value function V. The value function V is a function of the μAUV state and describes the finite horizon cost function (4.5)'s minimum value. It is defined as

$$V(\boldsymbol{x}_j) = \min_{\boldsymbol{u}} J(\boldsymbol{\tau}_j) = \min_{\boldsymbol{u}} \mathop{\mathbb{E}}_{\boldsymbol{\tau}_j} \int_{t_j}^{t_{j+N}} \mathcal{L}(q(\boldsymbol{\xi}), \boldsymbol{u})\,\mathrm{d}t\,. \tag{4.12}$$

The value function (4.12) is the solution of a stochastic HJB equation. For this problem, it becomes

$$-\partial_t V = \min_{\boldsymbol{u}} \Big(q(\boldsymbol{\xi}) + \frac{1}{2}\boldsymbol{u}^\top \boldsymbol{R}\boldsymbol{u} + (\nabla_\xi V)^\top(\boldsymbol{f} + \boldsymbol{\chi} + \boldsymbol{G}\boldsymbol{u}) + \frac{1}{2}\mathrm{tr}((\nabla_{\xi\xi} V)\boldsymbol{G}\boldsymbol{\Sigma}_\epsilon \boldsymbol{G}^\top) \Big)\,. \tag{4.13}$$

Taking the derivative with respect to $\boldsymbol{u}$ of the argument on the right hand side of (4.13) yields the optimal control function

$$\hat{\boldsymbol{u}} = -\boldsymbol{R}^{-1}\boldsymbol{G}^\top(\nabla_\xi V)\,. \tag{4.14}$$

Substituting (4.14) into the HJB equation (4.13) yields the nonlinear partial differential equation (PDE)

$$-\partial_t V = q(\boldsymbol{\xi}) + (\nabla_\xi V)^\top (\boldsymbol{f} + \boldsymbol{\chi}) - \frac{1}{2}(\nabla_\xi V)^\top \boldsymbol{G}\boldsymbol{R}^{-1}\boldsymbol{G}^\top (\nabla_\xi V) + \frac{1}{2}\mathrm{tr}\left((\nabla_{\xi\xi} V)\boldsymbol{G}\boldsymbol{\Sigma}_\epsilon \boldsymbol{G}^\top\right) . \tag{4.15}$$

We can apply a log-transformation to transform (4.15) into a linear PDE [32]. Set

$$V = -\lambda \log \psi \tag{4.16}$$

and $\lambda \boldsymbol{R}^{-1} = \boldsymbol{\Sigma}_\epsilon$, whereby λ is a constant, to obtain the Chapman-Kolmogorov PDE

$$-\partial_t \psi = \frac{1}{\lambda} q(\boldsymbol{\xi})\psi + (\boldsymbol{f} + \chi)(\nabla_\xi \psi) + \frac{1}{2}\mathrm{tr}((\nabla_{\xi\xi}\psi)\boldsymbol{G}\boldsymbol{\Sigma}_\epsilon \boldsymbol{G}^\top) . \tag{4.17}$$

Although (4.17) is now linear in ψ, analytical solutions cannot be obtained in general. However, the Feynman-Kac theorem states the solution of Chapman-Kolmogorov PDEs [59] as path integrals. The solution of (4.17) as a path integral is

$$\psi = \mathop{\mathbb{E}}_{\boldsymbol{\tau}_j}\left[\exp\left(-\frac{1}{\lambda}\int_{t_j}^{t_{j+N}} q \,\mathrm{d}t\right)\right] . \tag{4.18}$$

The path integral (4.18) can be stated as a limit in discrete time [77] with $\Delta t = t_{j+1} - t_j$ being a time step:

$$\psi = \lim_{\Delta t \to 0} \int p(\boldsymbol{\tau}_j \mid \boldsymbol{x}_j) \exp\left(-\frac{1}{\lambda}\sum_{m=j}^{N} q_m \Delta t\right) \mathrm{d}\boldsymbol{\tau}_j , \tag{4.19}$$

which is a convenient formulation for implementation purposes.

Combining (4.14) and (4.16) allows to state the optimal control action in terms of the path integral (4.19):

$$\hat{\boldsymbol{u}} = -\boldsymbol{R}^{-1}\boldsymbol{G}^\top (\nabla_\xi V) = \boldsymbol{R}^{-1}\boldsymbol{G}\frac{\nabla_\xi \psi}{\psi} . \tag{4.20}$$

The optimal exploratory control sequence can be obtained by computing a path integral and its derivative. It is well known that path integrals can be approximately computed by forward sampling. For the case of (4.19), the computing entails sampling μAUV paths and evaluating their costs. Taking a closer look at this formulation reveals that an optimization problem was transformed into an integration problem. Also, the path integral formulation does not require gradient computations of the cost function by exploiting probability-weighted averaging of sampled paths.

4.2.2 Discrete Receding Horizon Scheme

The results of the previous subsection can be implemented within a discrete receding horizon formulation. The equations are stated for a fixed point in time t_j from the perspective of a single μAUV, as in the previous subsection.

First, a discrete-time representation of the system dynamics is derived from (4.10). The resulting discretized equations of motion are used within the controller formulation. They allow to compute virtual roll-out paths by taking the control sequence as an input and propagate it through the dynamics. These virtual roll-out paths are used to evaluate the potential cost of a sampled control sequence. At time-step t_j, the controller's internal discrete equations of motion read

$$\boldsymbol{\xi}_{j+1} = \boldsymbol{\xi}_j + (\boldsymbol{f}(\boldsymbol{\xi}_j) + \boldsymbol{\chi}(\boldsymbol{\xi}_j))\Delta t + \boldsymbol{G}_j(\boldsymbol{u}_j \Delta t + \boldsymbol{\epsilon}_j \sqrt{\Delta t})\,. \tag{4.21}$$

The control sequence along the finite control horizon $\boldsymbol{u}_{j:N}$ is discretized into N steps and (4.21) becomes

$$\boldsymbol{\xi}_{j+1} = \boldsymbol{\xi}_j + (\boldsymbol{f}(\boldsymbol{\xi}_j) + \boldsymbol{\chi}(\boldsymbol{\xi}_j))\Delta t + \begin{bmatrix} 0 \\ 0 \\ \boldsymbol{g}_1^\top \end{bmatrix} \left(\begin{bmatrix} 0 \\ 0 \\ \boldsymbol{u}_{j:N}\Delta t \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \boldsymbol{\epsilon}_{j:N}\sqrt{\Delta t} \end{bmatrix} \right). \tag{4.22}$$

The column vector $\boldsymbol{g}_1$ is constructed such that the first element is equal to one and all other elements are zero. Consequently, at the prediction step $j+m$, the vector $\boldsymbol{g}_m$ contains a unit entry at its $(m-1)$-th position and all other elements are zero at time-step j. This direct time indexing allows for flexibility, but can yield computational complexity. This approach is feasible in our application because a μAUV's equations of motion are low-dimensional. Alternatively, the control input can be parametrized with dynamic motion primitives as in [76]. However, this approach reduces the control space.

The optimal control action (4.20) can be explicitly computed for the structure of (4.22). First, ψ is computed. Since the dynamics (4.21) are Markovian, the conditional distribution $p(\boldsymbol{\tau}_j \mid \boldsymbol{x}_j)$ can be written as a product, and (4.19) becomes

$$\psi = \lim_{\Delta t \to 0} \int \prod_{m=j}^{N} p(\boldsymbol{\xi}_{m+\Delta t} \mid \boldsymbol{\xi}_m) \exp\left(-\frac{1}{\lambda} \sum_{m=j}^{N} q_m \Delta t \right) \mathrm{d}\boldsymbol{\tau}_j\,. \tag{4.23}$$

The transition probability $p(\boldsymbol{\xi}_{j+\Delta j} \mid \boldsymbol{\xi}_i)$ is Gaussian because the exploration noise $\boldsymbol{\epsilon}$ is zero-mean Gaussian. Hence, (4.23) is

$$\psi = \lim_{\Delta t \to 0} \int \prod_{m=j}^{N} \mathcal{N}\left(\underbrace{\boldsymbol{\xi}_m + \boldsymbol{f}(\boldsymbol{\xi}_m) + \boldsymbol{\chi}(\boldsymbol{\xi}_m)}_{\text{mean}},\ \underbrace{\boldsymbol{g}_m \boldsymbol{\Sigma}_\epsilon \boldsymbol{g}_m^\top}_{\text{variance}} \right) \exp\left(-\frac{1}{\lambda} \sum_{m=j}^{N} q_m \Delta t \right) \mathrm{d}\boldsymbol{\tau}_j\,. \tag{4.24}$$

The normal distribution in (4.23) can be expressed with the exponential function:

$$\begin{aligned} \psi \propto \lim_{\Delta t \to 0} \int \exp\Big(-\frac{1}{\lambda} \sum_{m=j}^{N} q_m \Delta t - \frac{1}{2\lambda} \sum_{m=j}^{N} (\boldsymbol{\xi}_{m+\Delta t} - \boldsymbol{\xi}_m - \boldsymbol{f}(\boldsymbol{\xi}_j) - \boldsymbol{\chi}(\boldsymbol{\xi}_j))^\top (\boldsymbol{g}_m \boldsymbol{\Sigma}_\epsilon \boldsymbol{g}_m^\top)^{-1} \\ \cdot (\boldsymbol{\xi}_{m+\Delta t} - \boldsymbol{\xi}_m - \boldsymbol{f}(\boldsymbol{\xi}_j) - \boldsymbol{\chi}(\boldsymbol{\xi}_j)) \Big) \mathrm{d}\boldsymbol{\tau}_j\,, \end{aligned} \tag{4.25}$$

where we omitted the constant of the normal distribution for the sake of clarity. Appendix B summarizes properties of the normal distribution. By inserting (4.25) into (4.20) and exploiting the relationship $\lambda \boldsymbol{R}^{-1} = \boldsymbol{\Sigma}_\epsilon$, the optimal control sequence is rearranged to

$$\hat{\boldsymbol{u}}_{j:N} = \int P(\boldsymbol{\tau}_j)\boldsymbol{u}_{\mathrm{L}}(\boldsymbol{\tau}_j)\mathrm{d}\boldsymbol{\tau}_j \,, \tag{4.26}$$

with the probability

$$P(\boldsymbol{\tau}_j) = \frac{\exp(-\frac{1}{\lambda}S(\boldsymbol{\tau}_j))}{\int \exp(-\frac{1}{\lambda}S(\boldsymbol{\tau}_j))\mathrm{d}\boldsymbol{\tau}_j} \,, \tag{4.27}$$

where

$$S(\boldsymbol{\tau}_j) = \sum_{m=1}^{N} q_j(\boldsymbol{\tau}_j) + \frac{1}{2}(\boldsymbol{u}_m + \boldsymbol{M}_m\boldsymbol{\epsilon}_m)^\top \boldsymbol{R}(\boldsymbol{u}_m + \boldsymbol{M}_m\boldsymbol{\epsilon}_m) \tag{4.28}$$

is also called path cost-to-go. The matrix $\boldsymbol{M}_m$ reads

$$\boldsymbol{M}_m = \frac{\boldsymbol{R}^{-1}\boldsymbol{g}_m\boldsymbol{g}_m^\top}{\boldsymbol{g}_m^\top\boldsymbol{R}^{-1}\boldsymbol{g}_m} \,. \tag{4.29}$$

The local control is defined as

$$\boldsymbol{u}_{\mathrm{L},m} = \boldsymbol{M}_m\boldsymbol{\epsilon}_m \,. \tag{4.30}$$

At time-step t_j, the complete optimal control sequence $\hat{\boldsymbol{u}}_{j:N}$ is computed according to (4.26) through sampling. However, only the first input $\hat{\boldsymbol{u}}_j$ of the computed sequence is applied to the μAUV. The control action leads to a state transition of the μAUV where it takes the next field measurement $\boldsymbol{z}_{j+1}$ at its new position. Next, the belief representation is updated with the new measurement. Subsequently, the optimal control problem is solved again at time-step t_{j+1} in order to obtain the control input $\hat{\boldsymbol{u}}_{j+1}$ from the sequence $\hat{\boldsymbol{u}}_{j+1:j+N+1}$. The μAUV communicates its measurements and intended control sequences to other μAUVs within its communication range. At the same time, it receives other μAUVs' measurements which it includes in its belief computation. The intended control sequences from other μAUVs are included in (4.8). When considering other μAUVs' paths, it is computationally intractable to include all possible paths for all μAUVs. A common approach, which is also deployed here, is to communicate the current intended best path or best control sequence. The required steps to derive a control action are illustrated in Fig. 4.2.

Numerical Implementation of the Control Input For clarity, we state the discretized equations for computing the control input in an algorithmic form. They origin directly from the derivations above, but include a sampling strategy to solve the path integral (4.26) iteratively. The equations are solved at each time step and provide the exploratory control sequence for an available belief.

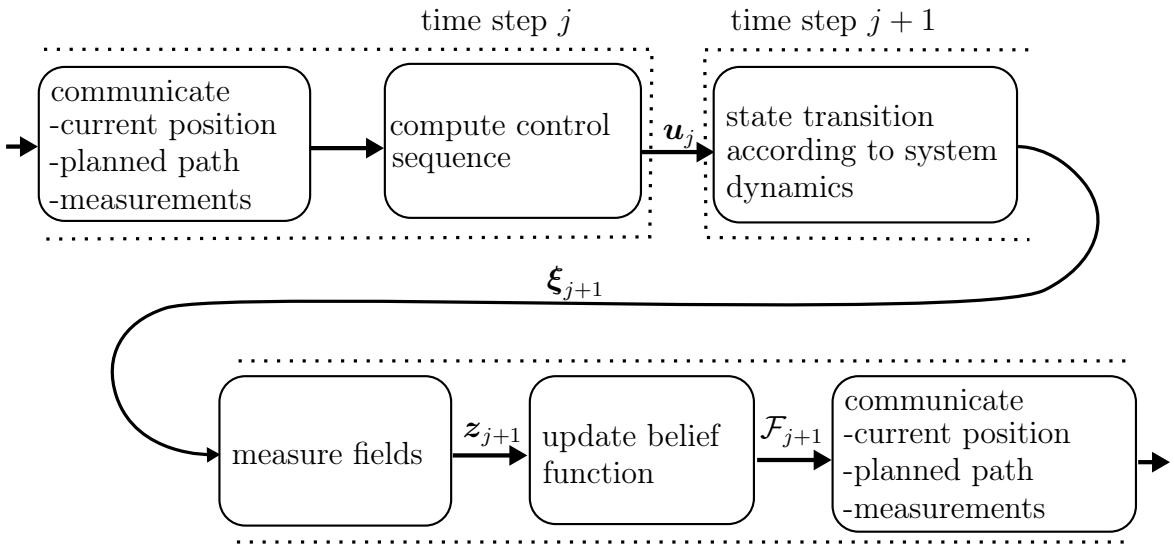


Figure 4.2: The steps performed by a single μAUV at each time-step in exploratory control. The control input computed at the previous time-step leads to a state transition and hence to a new μAUV position. The μAUV takes a new measurement at its position and updates its belief representation. The new measurements, the own position and the planned path, are communicated to vehicles in reach. Afterwards, a new control sequence is computed.

At time step t_j, initialize the control sequence to coincide with the control sequence of time step t_{j-1}. Repeat until convergence or until transition to time step t_{j+1}:

1. Create K virtual roll-out paths $\boldsymbol{\tau}_k$ from the current vehicle state $\boldsymbol{\xi}_j$ by sampling the stochastic control sequence $\boldsymbol{u}_{j:N}+\boldsymbol{\epsilon}_{k,j:N}$ along the control horizon and by propagating the control sequence through the controller internal μAUV dynamics (4.21).

2. Execute the following steps for every virtual path roll-out k and every control horizon step m.

   - Compute $\boldsymbol{M}_m = \dfrac{\boldsymbol{R}^{-1}\boldsymbol{g}_m\boldsymbol{g}_m^\top}{\boldsymbol{g}_m^\top\boldsymbol{R}^{-1}\boldsymbol{g}_m}$. This is same for all k paths.

   - Compute the path cost by using the state cost q that is dependent on the field belief and the intended control sequences of other μAUVs:
     $S(\boldsymbol{\tau}_{k,m}) = \sum_{m=1}^{N} q_j(\boldsymbol{\tau}_{k,m}) + \frac{1}{2}(\boldsymbol{u}_m + \boldsymbol{M}_m\boldsymbol{\epsilon}_{k,m})^\top \boldsymbol{R}(\boldsymbol{u}_m + \boldsymbol{M}_m\boldsymbol{\epsilon}_{k,m})$.

   - Evaluate the probability of the discrete path pieces for each roll-out:
     $P(\boldsymbol{\tau}_{k,m}) = \frac{\exp(-\frac{1}{\lambda}S(\boldsymbol{\tau}_{k,m}))}{\sum_{k=1}^{K}\exp(-\frac{1}{\lambda}S(\boldsymbol{\tau}_{k,m}))}$.

3. Compute the control correction: $\Delta\boldsymbol{U}_m = \sum_{k=1}^{K} P(\boldsymbol{\tau}_{k,m})\boldsymbol{M}_m\boldsymbol{\epsilon}_{k,m}$.

4. The control correction matrix $\Delta \boldsymbol{U}_m$ has the dimension $N \times N$, because the control correction vector is computed at every step. Hence, we average $\Delta \boldsymbol{U}_m$ and obtain the updated control sequence $\boldsymbol{u}_{j:N} \leftarrow \boldsymbol{u}_{j:N} + \frac{1}{N} \Delta \boldsymbol{U}_m \mathbf{1}$.

The loop is repeated. The control sequence $\boldsymbol{u}_{j:N}$ is updated until convergence or until the next control command has to be applied to the system. In either case, the first entry of the control sequence $\boldsymbol{u}_{j:N}$ is applied to the μAUV, resulting in a new μAUV state at time step t_{j+1}. A new field measurement can be taken, which in turn leads to a field belief update that allows to repeat the control loop.

4.3 Random Field Modeling and Inference

The previous section introduced a control architecture for computing exploratory control sequences for μAUVs based on path integrals. The computation requires the evaluation of the field state cost q_{field} which is obtained from (4.7). The predictive variance in (4.7) can be computed for a field belief that is represented by a GMRF. This section covers the random field structure which is embedded within the PI-GMRF controller.

4.3.1 Overview

Research on representing spatio-temporal fields with Gaussian random fields (GRFs) became increasingly popular over the past couple of years [71]. This is motivated by the growing amount of environmental sensors and data. Gaussian random fields allow to perform nonparametric regression which is a method for obtaining function representations of fields from measurements at discrete points. Physics-based models, such as the Navier-Stokes equation, combined with filtering methods offer an alternative to GRFs for reconstructing fields from measurements. However, Gaussian random fields offer distinct structural advantages over physics-based models in the context of spatial field studies. Random fields do not require boundary conditions, they are computationally much faster, and they provide a covariance measure, which helps to assess field uncertainty.

Gaussian random fields without the Markov property have been widely used for modelling data-driven processes in many applications. While GRFs are in principle suitable for modeling environmental fields, their computational scalability hinders applications with many measurements. This is a problem when computations need to be carried out on-board of resource-constrained μAUVs. Sparse GRFs have been suggested in [7] to reduce the computational burden. Recent efforts in modeling environmental fields incorporate GMRFs to approximate GRFs. Results by [48] motivate the application of GMRFs instead

of GRFs for spatial fields. The authors have proven that a lattice generates approximate solutions of a stochastic partial differential equation whose solution is a continuous GRF with a Matérn covariance function. GMRFs possess the Markov property in space and hence are computationally lighter than GRFs. This makes GMRFs more attractive for mobile robots. Appendix C gives a brief introduction to Markov models. The computational complexity of GRF regression is $\mathcal{O}(n^3)$ while GMRF regression in the plane scales with $\mathcal{O}(n^{\frac{3}{2}})$, where n is the number of measurements. Applications where GMRFs are used to navigate autonomous agents are rare so far. A regression framework where a scalar field is modeled with a GMRF is presented in [83].

Although GMRFs are designed for undirected graphs, this section extends them to a continuous domain by interpolating with shape-functions. Furthermore, we show how the predictive mean and covariance can be computed efficiently along an arbitrary μAUV path without having to compute field estimates at any other domain location. This allows evaluating the cost of sampled paths during the controls computation (4.26) very efficiently. The GMRF representation is suitable for scalar fields. However, the method easily extends to flow fields by treating each dimensional component of the flow field as a scalar field.

4.3.2 Gaussian Markov Random Field Model

Consider a discretization of the field domain of interest $\mathcal{D}$ into a two- or three-dimensional lattice graph, called a grid. The vertices of the grid are located at $\boldsymbol{s}_{\text{grid}} \in \mathcal{D}$. Let $\boldsymbol{w}$ be the vector of unknown field values at the vertices $\boldsymbol{s}_{\text{grid}}$, whereby all random field values are jointly Gaussian and form a Markov random field:

$$\boldsymbol{w} \sim \text{GMRF}(\boldsymbol{\mu}, \boldsymbol{Q}^{-1})\,. \tag{4.31}$$

The Markov property of a GMRF is equivalent to the property that its precision matrix element $Q_{ij} = 0$ if and only if two vertices $\boldsymbol{s}_{\text{grid},i}$ and $\boldsymbol{s}_{\text{grid},j}$ are conditionally independent. The Markov property ensures sparsity of $\boldsymbol{Q}$ which is the reason why regression of two dimensional GMRFs scales with $\mathcal{O}(n^{\frac{3}{2}})$ and regression of three dimensional GMRFs with $\mathcal{O}(n^2)$, whereby n is the number of measurement points. This is an advantage compared to their GRF counterparts, which have the computational complexity of $\mathcal{O}(n^3)$. Different constructions of the precision matrix $\boldsymbol{Q}$ are possible. For a two-dimensional lattice, the precision matrix can be constructed such that the Gaussian full conditionals fulfill

$$\begin{aligned} \mathbb{E}[w_{i,j} \mid \boldsymbol{w}_{-i,j}] &= \frac{1}{\rho}(w_{i-1,j} + w_{i+1,j} + w_{i,j-1} + w_{i,j+1})\,, \\ \text{Var}[w_{i,j} \mid \boldsymbol{w}_{-i,j}] &= \frac{1}{\rho}\,, \end{aligned} \tag{4.32}$$

with the constant $|\rho| > 4$, which guarantees stability of the GMRF. The structure of (4.32) implies that each vertex only communicates with its four grid neighbours directly. A similar construction is easily obtained for a three-dimensional lattice, whereby each vertex would have six direct neighbours. A GMRF with the precision matrix structure described by (4.32) approximates a Matérn field, whose covariance monotonically decreases over the Euclidean distance $\|\boldsymbol{s} - \boldsymbol{s}'\|$ and converges to zero. This is a desired property for modeling environmental fields because most physical fields possess a similar characteristic. The squared exponential covariance function, which is often chosen for GRFs, exhibits this property as well. A time dimension can be included for fields that show temporal variability. Time can be treated as an independent dimension. Intuitively, the autocovariance of such fields decreases over time. This can be considered by extending the well known autoregressive AR(1) model to the field value vector $\boldsymbol{w}$:

$$\mathrm{Var}[\boldsymbol{w}_{t+n}, \boldsymbol{w}_t] = \frac{\sigma_z}{1-\phi^2}\phi^{|n|}\,, \tag{4.33}$$

where t indexes time, n indicates a time shift, and $|\phi| < 1$. In GMRF regression, this property yields more weight to more recent measurements, whereas older measurements have less influence.

Given the probabilistic forward model (4.31) for $\boldsymbol{w}$, we can state the measurement equation (4.2) for the field in probabilistic notation as

$$p(\boldsymbol{z} \mid \boldsymbol{w}) = \mathcal{N}(\boldsymbol{A}(\boldsymbol{s}_z)\boldsymbol{w},\ \sigma_z^2\boldsymbol{I})\,, \tag{4.34}$$

where $\boldsymbol{z}$ combines all field measurements taken at the corresponding positions $\boldsymbol{s}_z$, and σ_z^2 is the variance of the measurement noise. The observation matrix $\boldsymbol{A}$ maps from discrete grid vertices $\boldsymbol{s}_{\text{grid}}$ of the GMRF to continuous measurement locations $\boldsymbol{s}_z$. This allows to represent the concentration field at any location of the domain and not only at the grid vertices. An efficient construction of $\boldsymbol{A}$ with polynomial shape functions is introduced in Subsection 4.3.4.

4.3.3 Gaussian Markov Random Field Inference

For GMRFs, the field measurements $\boldsymbol{z}$ and the field values at the vertices $\boldsymbol{w}$ are by design assumed to be jointly a GMRF as well [68]. Here, we exploit a major advantage of the Bayesian framework. For a given probabilistic measurement model $p(\boldsymbol{z} \mid \boldsymbol{w})$, the inversion $p(\boldsymbol{w} \mid \boldsymbol{z})$ can be computed.

This predictive posterior distribution is the conditional distribution

$$p(\boldsymbol{w} \mid \boldsymbol{z}) = \mathcal{N}\left(\mathbb{E}[\boldsymbol{w} \mid \boldsymbol{z}],\ \mathrm{Var}[\boldsymbol{w} \mid \boldsymbol{z}]\right) \ \text{with} \tag{4.35}$$

$$\mathbb{E}[\boldsymbol{w} \mid \boldsymbol{z}] = \boldsymbol{\mu} + \frac{1}{\sigma_z^2} \boldsymbol{Q}_{\boldsymbol{w}|\boldsymbol{z}}^{-1} \boldsymbol{A}^\top (\boldsymbol{z} - \boldsymbol{A}\boldsymbol{\mu}) \ \text{and} \tag{4.36}$$

$$\mathrm{Var}[\boldsymbol{w} \mid \boldsymbol{z}] = \boldsymbol{Q}_{\boldsymbol{w}|\boldsymbol{z}}^{-1} = (\boldsymbol{Q} + \frac{1}{\sigma_z^2} \boldsymbol{A}^\top \boldsymbol{A})^{-1}\,. \tag{4.37}$$

In general, the mean $\boldsymbol{\mu}$ is unknown, and can be approximated by the arithmetic mean of the collected measurements $\boldsymbol{\mu} \approx \overline{\boldsymbol{\mu}}_{\boldsymbol{z}}$. The predictive distribution allows to infer the unknown field values $\boldsymbol{w}$ at the vertices of the field grid from measurements $\boldsymbol{z}$. Hence, the concentration field can be estimated at discrete locations from measurements taken at arbitrary positions.

The cost-to-go (4.28) in the path integral control formulation requires the covariance evaluation along paths $\boldsymbol{\tau}_j$. A naive approach would compute the posterior distribution at all grid vertices $\boldsymbol{s}_{\text{grid}}$ according to (4.35) first. Afterwards, an interpolation would be required to obtain covariance estimates along a sampled path. This requires a lot of computational resources. Consider that we might sample hundreds of trajectories in each time step, and the cost of each path has to be evaluated to derive the control sequence. Instead, we suggest to compute the posterior distribution $p(\boldsymbol{w}_\tau \mid \boldsymbol{z})$ directly, where $\boldsymbol{w}_\tau$ are only the field estimates along a path. The computation of values which do not directly lie on sampled paths is not required. This is an elegant property of the probabilistic field modeling framework and is in general not applicable within physics-based field modeling.

We introduce the path observation matrix $\boldsymbol{A}_\tau = \boldsymbol{A}(\boldsymbol{\tau}_j)$ that maps field values $\boldsymbol{w}$ onto $\boldsymbol{w}_\tau$. The posterior distribution $p(\boldsymbol{w}_\tau \mid \boldsymbol{z})$ can be computed as

$$\begin{aligned}
p(\boldsymbol{w}_\tau \mid \boldsymbol{z}) &= \mathcal{N}(\mathbb{E}[\boldsymbol{w}_\tau \mid \boldsymbol{z}],\ \mathrm{Var}[\boldsymbol{w}_\tau \mid \boldsymbol{z}]) \\
&= \mathcal{N}(\boldsymbol{A}_\tau\, \mathbb{E}[\boldsymbol{w} \mid \boldsymbol{z}],\ \boldsymbol{A}_\tau \mathrm{Var}[\boldsymbol{w} \mid \boldsymbol{z}] \boldsymbol{A}_\tau^\top), \\
\mathbb{E}[\boldsymbol{w}_\tau \mid \boldsymbol{z}] &= \boldsymbol{A}_\tau \boldsymbol{\mu} + \boldsymbol{A}_\tau \left(\boldsymbol{Q} + \frac{1}{\sigma_z^2} \boldsymbol{A}^\top \boldsymbol{A}\right)^{-1} \frac{1}{\sigma_z^2} \boldsymbol{A}^\top (\boldsymbol{z} - \boldsymbol{\mu}), \\
\mathrm{Var}[\boldsymbol{w}_\tau \mid \boldsymbol{z}] &= \boldsymbol{A}_\tau \left(\boldsymbol{Q} + \frac{1}{\sigma_z^2} \boldsymbol{A}^\top \boldsymbol{A}\right)^{-1} \boldsymbol{A}_\tau^\top\,.
\end{aligned} \tag{4.38}$$

The formulation is computationally efficient because the term $(\boldsymbol{Q} + \frac{1}{\sigma_z^2} \boldsymbol{A}^\top \boldsymbol{A})^{-1}$ does not depend on the virtual path roll-outs of the controller, but only on the already available measurements. Hence, each μAUV only needs to compute this inverse when a new measurement arrives and can reuse it for the evaluation of all virtual path roll-outs to derive the control sequence.

4.3.4 Weighted Shape Functions

The observation matrix $\boldsymbol{A}$ and the path observation matrix $\boldsymbol{A}_\tau$ map the field values $\boldsymbol{w}$ from discrete vertices to the measurement locations and to the sampled paths, respectively. This subsection introduces the construction of $\boldsymbol{A}$, which can be easily extended to $\boldsymbol{A}_\tau$. In order to preserve the computational advantages which GMRFs posses over GRFs, $\boldsymbol{A}$ has to be sparse. The gridded structure of the GMRF decomposes the field domain $\mathcal{D}$ into finite sub-domains or elements $\mathcal{D}_e$:

$$\mathcal{D} = \overline{\bigcup_e \mathcal{D}_e}\,, \tag{4.39}$$

whereby n_{vert} vertices of the grid $\boldsymbol{s}_{\text{grid}}$ are associated with one element $\mathcal{D}_e$ and completely define its geometry. The field $\mathcal{F}^e$ in an element domain is approximated by $\mathcal{F}_h^e$, a sum of weighted shape functions. Given an element $\mathcal{D}_e$ with n_{vert} vertices, which have the coordinates $\mathbf{s}_i,\ i = 1, \ldots, n_{\text{vert}}$, the random field function $\mathcal{F}^e$ can be approximated by polynomial shape functions $\phi^e(\boldsymbol{s})$ and the corresponding stochastic field values w_i^e as weights:

$$\mathcal{F}^e \approx \mathcal{F}_h^e = \sum_{i=1}^{n_{\text{vert}}} \phi_i^e(\mathbf{s}) w_i^e\,. \tag{4.40}$$

The polynomials ϕ_i^e are chosen such that the condition

$$\phi_i^e(\mathbf{s}_k) = \begin{cases} 1 \text{ if } i = k, \\ 0 \text{ otherwise} \end{cases} \tag{4.41}$$

is fulfilled. Furthermore, each shape function vanishes outside of its corresponding element.

The shape functions are used to construct $\boldsymbol{A}$. The j-th row $\boldsymbol{A}_j$ of $\boldsymbol{A}$ is associated with the j-th measurement which was taken at position $\boldsymbol{s}_j$. The position $\boldsymbol{s}_j$ is located within element $\mathcal{D}_{e'}$, that is, $\boldsymbol{s}_j \in \mathcal{D}_{e'}$. Hence, n_{vert} components in the j-th row $\boldsymbol{A}_j$ are associated with the n_{vert} vertices and result in non-zero entries of the j-th row:

$$\boldsymbol{A}_j = [0 \ \cdots \ \underbrace{\phi_1^{e'}(\boldsymbol{s}_j) \ \cdots \ \phi_{n_{\text{vert}}}^{e'}(\boldsymbol{s}_j)}_{\text{associated with } \mathcal{D}_{e'}} \ \cdots \ 0]. \tag{4.42}$$

Note, $\boldsymbol{A}$ is sparse because each row contains only n_{vert} non-zero elements. The path matrix $\boldsymbol{A}_\tau$ can be constructed in a similar way, but instead of measurement locations, sampled paths are used.

The construction is illustrated with a two-dimensional example. Four vertices $n_{\text{vert}} = 4$ compose an element. Each element has side lengths a and b. Figure 4.3 illustrates a sample

shape function defined in such an element. A local coordinate system $\mathcal{K}_l(x_l, y_l)$ is introduced to define the shape functions for each element. The coordinate system originates in the center of an element. Suggested shape functions are

$$\begin{aligned}
\phi_1^e &= \frac{1}{ab}\left(x_l - \frac{a}{2}\right)\left(y_l - \frac{b}{2}\right) , \\
\phi_2^e &= -\frac{1}{ab}\left(x_l + \frac{a}{2}\right)\left(y_l - \frac{b}{2}\right) , \\
\phi_3^e &= \frac{1}{ab}\left(x_l + \frac{a}{2}\right)\left(y_l + \frac{b}{2}\right) , \\
\phi_4^e &= -\frac{1}{ab}\left(x_l - \frac{a}{2}\right)\left(y_l + \frac{b}{2}\right) .
\end{aligned} \tag{4.43}$$

Other shape functions are possible as long as they fulfill condition (4.41). As required by condition (4.41) the shape functions (4.43) vanish at all element nodes except for one at which they attain unit value. By definition, all shape functions vanish outside their local element. For a GMRF with 100×100 vertices, the corresponding matrix $\boldsymbol{A}$ has zero entries at 99.96 % of its elements. This sparsity is independent of the number of taken measurements.

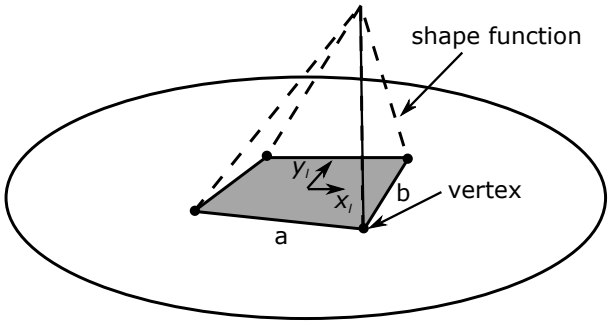


Figure 4.3: Rectangular element defined by four vertices of the domain grid. The filled area illustrates the sub-set $\mathcal{D}_e$. A polynomial shape function is defined in a local coordinate system whose origin is in the element center. The illustrated shape function takes unit value at one vertex (solid line) and vanishes at all other vertices.

4.4 Case Studies

In this section, the PI-GMRF controller is analyzed in numerical simulations. Numerical simulations allow to isolate and test critical components of the algorithm and more importantlly to run experiments with controlled initial conditions to collect statistics. Two case studies are presented. First, a single μAUV has a known and constant belief, based

which control sequences are derived. This scenario evaluates only the path integral part of the PI-GMRF controller. Second, three μAUVs are exploring an unknown scalar field while being disturbed by an unknown flow field. For both case studies, a self-propelled particle model in the plane with orientation and unit mass is chosen as the controller internal model. The model is commonly used for studying collective motion of underwater vehicles [61]. The i-th μAUV has the state $\boldsymbol{\xi}^i = [x^i \; y^i \; \alpha^i]^\top$ which describes the position and orientation in a global coordinate system. The motion of each μAUV is governed by

$$\begin{bmatrix} \dot{x}^i \\ \dot{y}^i \\ \dot{\alpha}^i \end{bmatrix} = \begin{bmatrix} v\cos(\alpha^i) + \chi_x(x^i) \\ v\sin(\alpha^i) + \chi_y(y^i) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \tilde{u}^i \end{bmatrix}, \tag{4.44}$$

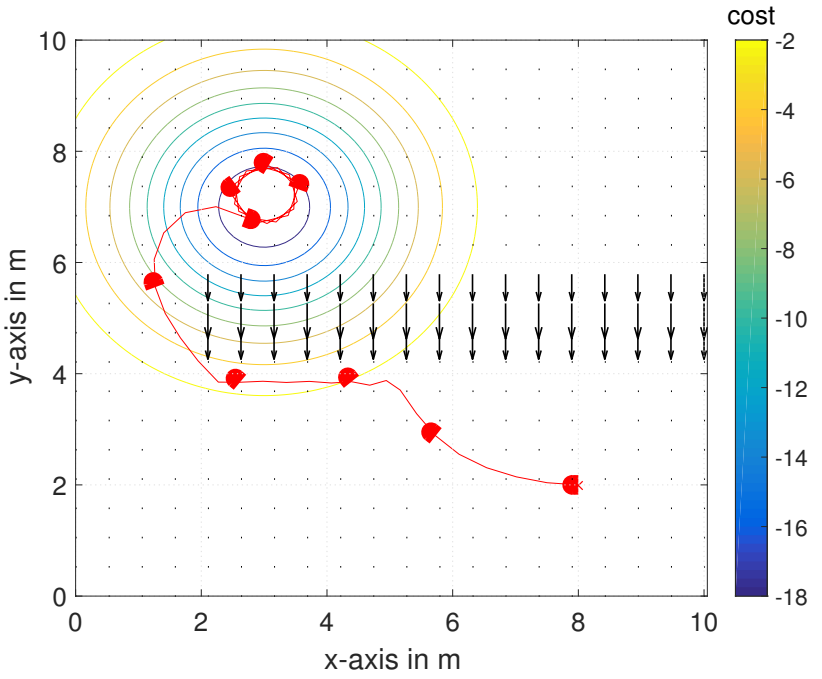
where v is the μAUV's velocity relative to the flow. We assume a constant vehicle velocity v. The flow field $\boldsymbol{\chi} = [\chi_x \; \chi_y]^\top$ has only translational influence on a μAUV and acts as a disturbance. The only controllable state is the orientation α^i through the angular velocity, whereby $\tilde{u}^i$ is the control input. Note that (4.44) is a special case of (4.1).

4.4.1 Cost Minimization for a Static Belief

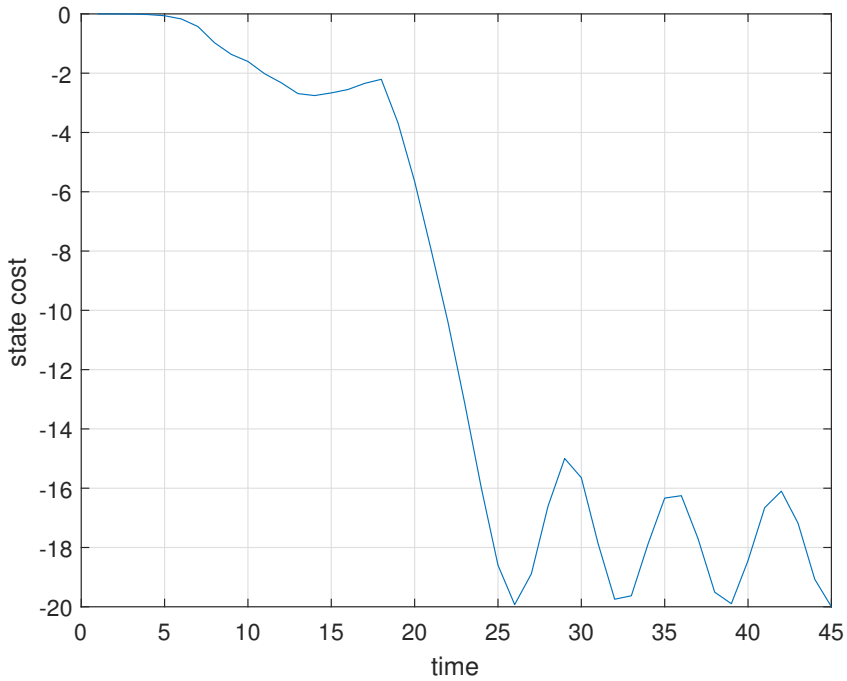
In the first case study, the path integral controller for navigating μAUVs is analyzed. A single μAUV has the goal to minimize its cost along its path while being subject to (4.44). The belief is constant, time-invariant, and known to the μAUV. The field cost function is set to

$$q_{\text{field}}(\boldsymbol{x}) = -20\exp\left(\frac{-(x-3)^2-(y-7)^2}{5}\right). \tag{4.45}$$

A flow field is present in the domain and has a flow velocity that exceeds the μAUV velocity. However, the flow field is precisely known to the μAUV at all times. The initial position of the μAUV is $[x_0 \; y_0]^\top = [8 \; 2]^\top$. The simulation setup and the resulting μAUV path are shown in Fig. 4.4. In the beginning of the simulation, the μAUV directly navigates along the steepest gradient of the field cost in order to decrease the cost as quickly as possible. When reaching the flow, which it cannot overcome, it steers around the flow to further decrease the cost along its projected receding horizon. After reaching the area where the cost function has a minimum, the μAUV locks into a pattern which guarantees the smallest cost along its path. Note that, by design of the study, the μAUV cannot just stop because its velocity is set to a constant nonzero value. Since the μAUV is subject to particle dynamics with orientation and a finite angular velocity, it reaches a stable circular path. Such a behavior is usually observed in formation control based on nonlinear coupled oscillators [46]. However, nonlinear coupled oscillator controllers offer only very limited capabilities to include environmental information. In [21], synchronization was achieved



(a) Path of the μAUV, which minimizes cost.



(b) Cost over time for the μAUV path.

Figure 4.4: Simulation results for path derivation with the path integral control law for a single μAUV. The path minimizes the cost while being subject to particle dynamics with orientation. The start position of the $[x_0 \ y_0]^\top = [8 \ 2]^\top$. Red markers illustrate equitemporal vehicle positions.

with path integral control as well. However, the underlying dynamics were single integrator dynamics, and no field was considered. Figure 4.4 demonstrates that path integral control leads to a mode where a single vehicle with unicycle dynamics synchronizes on a function.

4.4.2 Exploration of an Advection-Diffusion Field

In the second case, study a fleet of μAUVs explores an unknown scalar field while being disturbed by a flow field. The scalar and flow fields are shown in Fig. 4.5. In the presented scenario, the scalar and flow fields are not coupled, but just overlayed. Both fields are time-invariant and represent a scenario that can be encountered in a process engineering application, where the scalar field could be a chemical concentration of interest. The two-dimensional domain length is $10\,\mathrm{m} \times 10\,\mathrm{m}$. For clarity, the case study is set up in two dimensions, an extension to three dimensions is possible. The patchy flow field has only a velocity component in y-direction which amounts to $0.4\,\frac{\mathrm{m}}{\mathrm{s}}$ at $y = 5\,\mathrm{m}$ and decreases towards the top and the bottom of the tank. The scenario in Fig. 4.5 illustrates why preplanning the μAUV paths cannot work and an adaptive control strategy is required. The flow influence forces the vehicles to unexpected positions and might completely prevent them from visiting certain locations. A controller for environmental modeling has to adapt to the flow field in order to perform well.

We deploy $n_\mathrm{s} = 3$ μAUVs in order to estimate the field shown in Fig. 4.5. The μAUVs are steered by PI-GMRF controllers to minimize the predictive variance of the scalar field. The simulated μAUV dynamics correspond to (4.44). Unlike in the first case study of this chapter, the equations of motion for creating the virtual roll-outs do not include the real flow field because the real flow field is not known to the μAUV fleet. Instead, the controller learns the fluid field on the fly and exploits the current mean of the flow belief knowledge. The x-component and the y-component of the flow field are estimated separately. Hence, the belief consists in total of three GMRFs. Note that the flow uncertainty is not included in the cost function of the PI-GMRF controller. The μAUVs are not trying to actively explore the flow field, they just exploit their passively acquired knowledge of the flow to improve the control performance for concentration field exploration.

The μAUVs start at rest with initial orientations $\alpha_0^1 = \alpha_0^2 = \alpha_0^3 = -\pi/2$. The initial positions are randomly chosen along the coordinate $y = 9.5\,\mathrm{m}$,, and the x-coordinates are sampled from a uniform distribution $U(2, 8)\,\mathrm{m}$ with the condition that the μAUVs are 0.1 m apart from each other. The parameters of the PI-GMRF controller implementation are summarized in Table 4.1. The μAUVs update the control sequence ten times at each time step to ensure convergence. The communication radius is chosen such that all-to-all communication is always possible. Hence, the μAUVs can always communicate

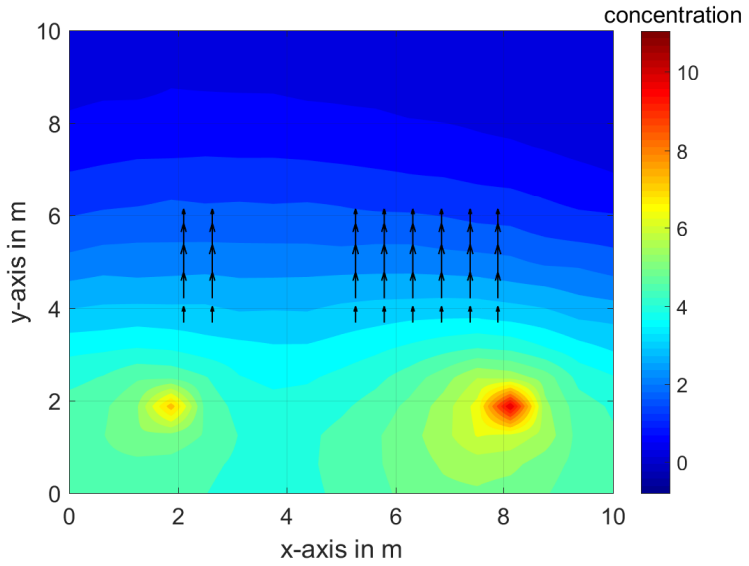


Figure 4.5: Environmental fields which serve as exploration scenario. The concentration is indicated by a colormap and the flow field by the flow velocity vectors in black. Warmer colors indicate higher concentrations and colder colder colors indicate lower concentrations. The initial position of the μAUV fleet is along $y = 9.5\,\mathrm{m}$ and the x-coordinate is sampled from a uniform distribution $U(2, 8)\,\mathrm{m}$ with the condition that the AUVs are $0.1\,\mathrm{m}$ apart from each other.

measurements and intended control actions with each other. Due to the communication topology, the three μAUVs share the same belief because they have access to the same measurement data. Results that include statistical analyses are based on 100 simulations for each unique parameter combination to obtain reliable statistics.

PI-GMRF Controller Feasibility

Figure 4.6 illustrates the predictive GMRF means of the concentration field and the flow field for four different points in time. The control horizon is $N = 15\,\mathrm{s}$. The colored lines in Figure 4.6 illustrate the travelled paths of the μAUVs. The current positions are indicated by colored markers. The mean of the belief at time step $1\,\mathrm{s}$ is a constant surface function over the domain, which represents the extrapolation of the first three very similar measurements. This is the best estimate the μAUVs can derive at this point with the limited information obtained. At time step $40\,\mathrm{s}$, the μAUVs have spread out in the domain. The black μAUV encountered the flow field, and the belief starts to reflect this information based on the measurements taken along the black path. At time step $80\,\mathrm{s}$, the fields already resemble the original fields, and after $120\,\mathrm{s}$, the μAUVs have obtained an accurate estimate of both fields, the scalar and the flow field.

Table 4.1: Parameters used for the simulations.

parameter	symbol	value
control horizon	N	0–15 s
μAUV velocity	v	0.5 $\frac{\text{m}}{\text{s}}$
simulation time	–	120 s
time step	Δt	1 s
trajectory samples	K	15
control loop updates	n_{updated}	10
cooperative cost	b_{coop}	0.2
number of grid vertices	$n_{\boldsymbol{s}_{\text{grid}}}$	144
concentration measurement variance	σ_{z}^2	0.3
flow field measurement variance	σ_{flow}^2	0.05

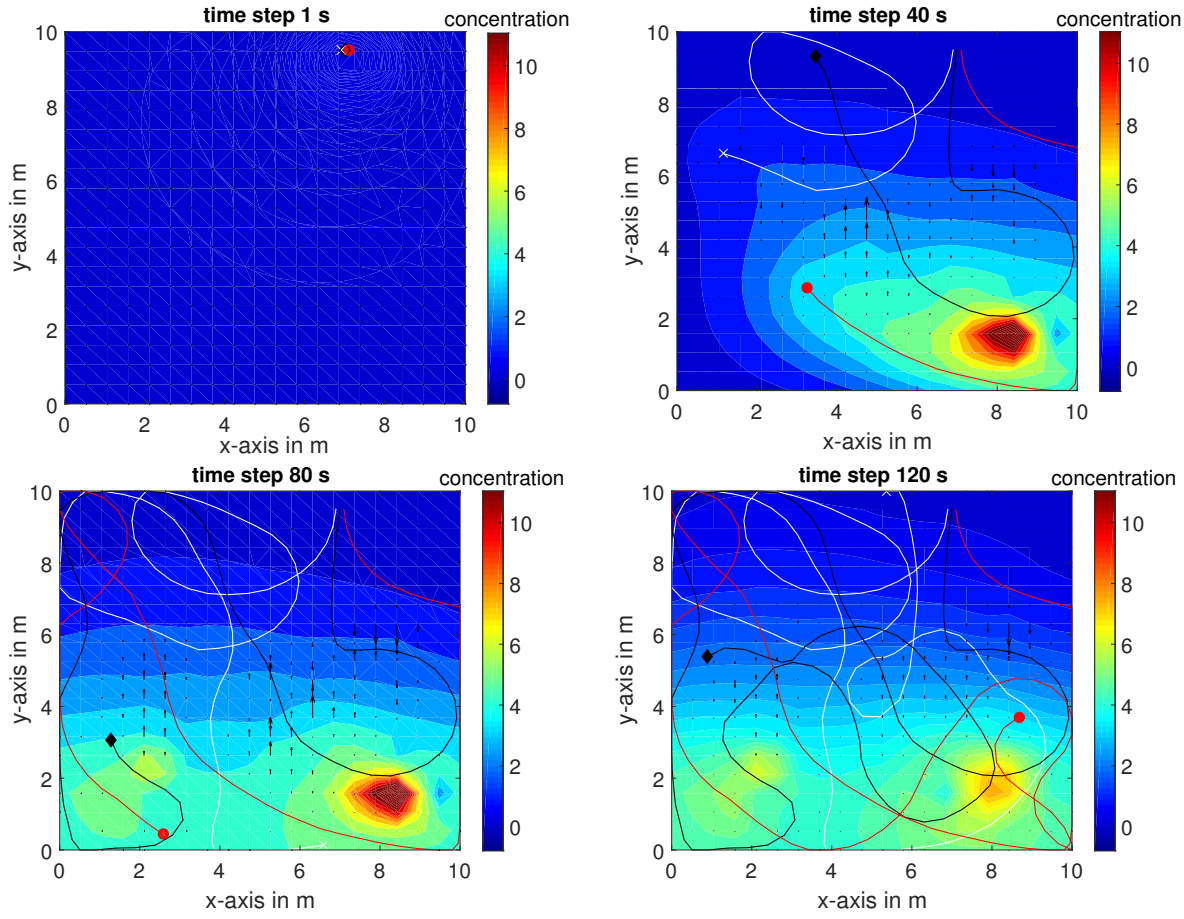


Figure 4.6: Field estimates for four different time-steps of a sample scenario realization. The current μAUV positions are indicated by markers while the traveled paths are illustrated by the white, black, and red lines.

Benchmark against Random Walk

Figure 4.7 shows the sum of the predictive variance (4.4) over time. The reference points are evaluated at the field vertices. The blue graph indicates the PI-GMRF control method, and the red graph is a random walk strategy, for which the control inputs are sampled from a normal distribution in each time step. The bold curves are the medians, and the envelopes represent the interquartile ranges (IQR) which are measures of statistical dispersion. Median and IQR are chosen because the data is not normally distributed.

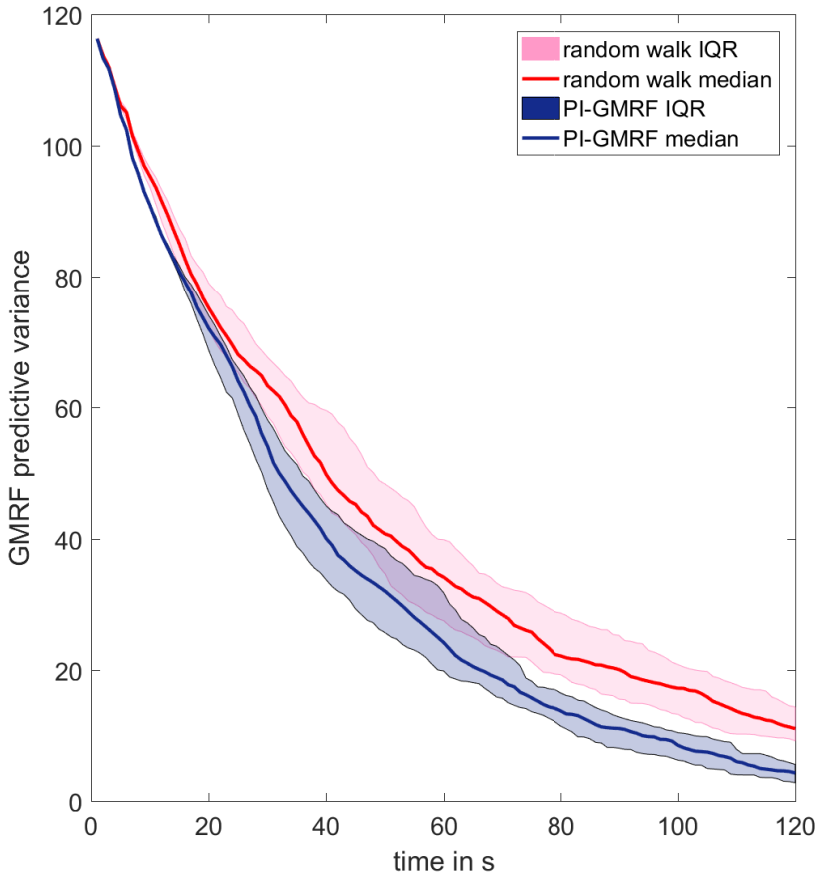


Figure 4.7: Medians and IQRs of the predictive variance evaluated at GMRF vertices as functions of time. The results are illustrated for a random walk strategy and the PI-GMRF control method.

The statistical dispersion of the results in Fig. 4.7 is due to three factors: (1) The initial positions are random. (2) The measurements are noise corrupted. (3) The controller inherently has exploration noise, and the number of trajectory samples is bounded. For both control strategies in Fig. 4.7, the IQR dispersion of the curves is zero in the beginning and increases afterwards, which is illustrated by the widths of the envelopes. Towards

the end of the simulations the dispersion decreases again. The dispersion is small in the beginning because initially, the environment is unexplored in each simulation run. During that phase, all μAUV motions lead to the best possible improvement of the field estimate. However, with progressing time, control adaption becomes more important to effectively continue exploration. Towards the end, most of the field has been explored, and not much variation in the exploration behavior is possible. The PI-GMRF controlled exploration missions lead to a faster decrease of the predictive variance, for example the field uncertainty, than the random walk strategy. The 75%-quartile of the PI-GMRF approach outperforms the 25%-quartile of the random walk strategy.

Control Horizon Influence and Computational Time The influence of a parameter on the controller performance can be determined by analyzing how long it takes to reach a desired field knowledge. The box plot in Fig. 4.8 shows the influence of the control horizon N, which is one of the main parameters of the PI-GMRF controller. Figure 4.8 illustrates the time at which the GMRF predictive variance at the grid vertices dips below 20, the so-called crossing time. A predictive variance of 20 corresponds to a field exploration success of approximately 85 %. Medians and IQR values for the random walk strategy and three different control horizons are shown in Fig. 4.8.

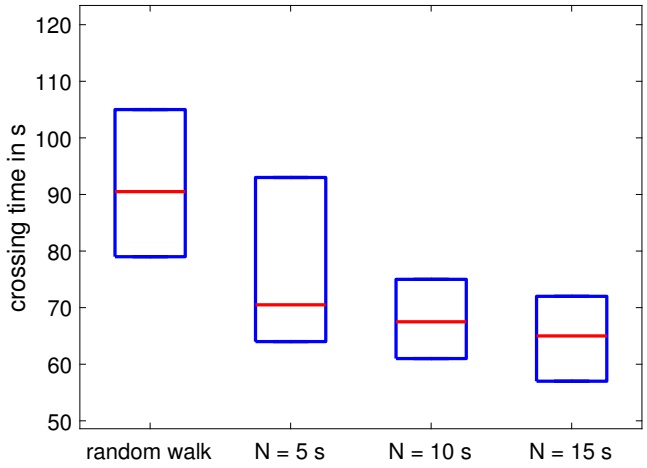


Figure 4.8: Box plot for the crossing time depending on the control horizon N. The crossing time is a measure for the exploration speed. The bottom and the top of each box are the first and third quartiles, respectively. The inside band is the median.

The random walk strategy requires the longest time to drive the predictive variance of the concentration field estimate below 20. The crossing times for $N = 10\,\mathrm{s}$ and $N = 15\,\mathrm{s}$ resemble each other. The crossing time for the control horizon $N = 5\,\mathrm{s}$ has the largest spread and a scewed distribution. In some simulation runs with $N = 5\,\mathrm{s}$, the controller

either performs similarly well as longer control horizons. In other simulations the controller shows an inferior performance, that has more in common with the random walk strategy.

The PI-GMRF controller is designed for μAUVs and one of its main benefits is the computational time. The presented results were obtained on a Quadcore-CPU with 2.66 GHz and 8 GB RAM. The control sequence computations during one time-step take approximately 200 ms for a control horizon of $N = 15$ s. Thereby, three GMRFs (scalar field and flow field) with 300 measurements each are considered, and 150 control sequence roll-outs (15 samples times 10 iteration updates) are evaluated. The computational time scales linearly with the control horizon. It amounts to approximately 100 ms for $N = 10$ s and 70 ms for $N = 5$ s. Current resource constrained μAUVs possess enough computing power to run the presented PI-GMRF controller.

4.5 Summary

This chapter introduced the PI-GMRF controller for field exploration, mapping, and monitoring with μAUVs. The PI-GMRF controller extends and combines the path integral control framework with GMRF regression. Path integral control was derived from the first principles of stochastic optimal control. The optimal control sequence is approximately obtained by forward sampling. A path integral approach allows for explicit consideration of system dynamics, and the informative control sequences can be derived efficiently on the fly. Both are important features for a controller that navigates computationally light μAUVs in confined test tanks. The PI-GMRF controller is based on a receding finite horizon scheme. At every time step, an optimal control sequence is computed for a finite control horizon, but only the first control command is executed on the μAUV. The control framework includes belief representations of the concentration fields and flow fields. Both are modeled with GMRFs. Gaussian Markov random fields allow to compute posterior distributions conditioned on measurement and scale better than their more popular GRF counterparts. The PI-GMRF controller coordinates the μAUVs in order to reduce predictive field covariance. The presented formulation does not require discretization of field, state, or control space. The control cost function includes a term that penalizes control sequences that would lead to similar paths within the fleet. This ensures coordination. Each μAUV runs its own PI-GMRF controller instance. Hence, there is no central unit required that would coordinate the μAUVs. The fleet members only need to communicate their measurements and their intended paths within their communication network. In this chapter, two scenarios were analyzed by means of numerical simulations. The first scenario introduced a static belief that was completely known to a single μAUV with the goal to

minimize a cost function. The result shows that the path integral controller exploits its belief and derives an optimal motion. The second scenario included the exploration of an initially unknown diffusive concentration field with three μAUV. The fleet was subject to an unknown flow field which interfered with its motion. This scenario represents a confined tank application that can be encountered in the chemical engineering industry. The PI-GMRF controller outperforms a random walk strategy. The framework is computationally light enough to run on embedded hardware of μAUVs such as HippoCampus.

Chapter 5 Conclusions

This dissertation covered the application of μAUVs for exploration and monitoring of environmental fields in confined environments. Design of μAUV systems, underwater self-localization, and control for information gathering were identified as the main problems. Solution approaches to these problems were presented in order to bridge the gap between state-of-the-art and exploratory μAUV behavior. The methods introduced in this dissertation bring mobile underwater vehicle systems closer to autonomous collaborative operations, which are particularly interesting for industry applications. This chapter summarizes the key contributions of this dissertation and concludes with possible future research directions.

5.1 Summary

A μAUV platform called HippoCampus for monitoring and exploring confined underwater environments was presented in Chapter 2. The mechatronic design of the HippoCampus μAUVs is inspired by recent advances in the field of micro aerial vehicles. The equations of motion for HippoCampus were derived and used for analyzing differential flatness of the system. Based on this dynamical analysis, a geometric attitude control system was proposed. It is based on a nonlinear geometric approach, which does not assume the common simplification that rotations evolve in Euclidean space. This is important for agile behavior. Besides attitude control, path following is another important functionality of μAUVs. A path following controller based on the receding horizon sliding control law was introduced. It allows to specify a path via waypoints and connect them with line segments. Simulations demonstrated that a μAUV follows such a path despite uncertainty about system parameters. The contributions of Chapter 2 advance state-of-the-art in μAUV hardware design and its controls.

One of the biggest challenges in deploying autonomous underwater systems is self-localization. If a μAUV does not know its position, it cannot perform useful operations. Chapter 3 introduced two approaches for the underwater self-localization problem. Both approaches are passive, i. e. only one-way signal transmission is required. A receiver senses ambient reference signals that are generated by anchored beacons. The first method determines time delays of arrival (TDOAs) of acoustic signals. The signals are generated from a noise distribution, and the TDOAs are determined through cross-correlation. Results demonstrated that self-localization in air and water is possible with the presented approach. The hardware is compact enough to fit into the hull of a μAUV. For very small underwater environments, reflections and deverberation can significantly decrease the performance. The second presented method was based on the attenuation of EM waves in water. The receiver measures the strength of an ambient EM signal at its own position and determines the ranges to the emitting beacons. Miniaturization of the overall system and the analyses of the performance trade-offs were two of the main contributions to the field of μAUVs. Experiments demonstrated that the developed method delivered sufficient localization performance, even in small tanks. The methods introduced in Chapter 3 bring on-board localization to the μAUV domain and close the position loop. This is a key step to render autonomous operations in confined underwater environments possible.

Chapter 4 introduced a novel, high-level control approach for autonomous field exploration with μAUVs. The path integral Gaussian Markov random field (PI-GMRF) controller extends and combines path integral control and GMRF inference. The formulation is based on stochastic optimal control, whereby a belief representation of the environmental fields is maintained within the controller. Field measurements are used to update the GMRF belief. Since GMRFs are stochastic fields, the covariance is readily available for evaluating control actions. An optimal control sequence is approximately obtained by forward sampling along a receding horizon. The approach allows for explicit consideration of system dynamics. The PI-GMRF controller was developed for computationally constrained mobile robots. Due to the iterative sampling implementation control sequences can be derived on the fly. Unlike POMDP-based methods, the PI-GMRF controller does not require discretization of the field, state, and control space. Simulations demonstrated that the PI-GMRF controller outperforms a random walk strategy. The PI-GMRF controller represents a novel methodological approach for information gathering. It demonstrates that probabilistic belief representations of environmental fields can be maintained and updated within a stochastic optimal control formulation.

5.2 Future Directions

This dissertation introduced methods and systems that enable μAUV operations in confined underwater environments. In order to increase the performance of autonomous underwater systems, future research directions should include improving hardware, self-localization, and advancing the PI-GMRF approach.

The HippoCampus design should be upgraded with increased computing hardware. This will allow to increase the performance of the PI-GMRF controller. Like all sampling based approaches, the PI-GMRF controller guarantees convergence to some expectation limit. The overall performance will improve with increased accuracy of the underlying path integral approximation, specifically by computing more inner loop path samples. Improved computing hardware will also enable a larger control receding horizon. The vehicle application would become more versatile if more sensing capabilities were included.

The main problem of the acoustic localization problem are interferences and reverberation due to the confined environment. Reverberation is a convolution of the reference signal with itself. For a known reference signal, dereverberation can be achieved. However, the presented acoustic self-localization approach relies on an unknown reference signal, which is an advantage from a conceptional point of view. Current advancements in signal processing and speech recognition render novel dereverberation methods possible without knowledge of the reference signal. The so-called blind deverbaration methods can be extended and used as signal postprocessing to robustify the presented acoustic localization.

The RF localization system can be extended to yaw estimation by deploying an altered antenna design. The system's performance can be significantly improved by developing customized hardware for signal processing instead of relying on an off-the-shelf DVB-T dongle.

The PI-GMRF approach includes a field belief that is represented by a stationary GMRF. In order to increase the controller's performance, more sophisticated GMRFs should be included. Accurate representation of nonstationary spatio-temporal fields is not possible with the current GMRF methods. Recently, Lindgren et al. [48, 71] introduced a novel GMRF based methodology which stands out for modeling nonstationary fields [6]. The PI-GMRF approach uses the predictive variance as the information metric, the mutual information criterion could provide improved results for nonstationary fields. Current assumptions on the decorrelation scales of the field are set as a priori for the controller. An adaption might also increase the performance, especially for nonstationary fields.

With the aforementioned improvements, further application areas for the HippoCampus μAUV may arise. Currently, the system does not interact with its environment and serves as a sensor platform. With improved localization and autonomy collaborative manipulation, intervention and transport missions can come into reach for μAUVs.

References

[1] F. Allgöwer, R. Findeisen, and Z. K. Nagy. Nonlinear Model Predictive Control: From Theory to Application. *Journal of the Chinese Institute of Chemical Engineers*, 35 (3):299–315, 2004.

[2] E. Borhaug, A. Pavlov, and K. Y. Pettersen. Straight line path following for formations of underactuated underwater vehicles. In *46th IEEE Conference on Decision and Control (CDC)*, pages 2905–2912, New Orleans, LA, USA, 2007.

[3] E. Borhaug, A. Pavlov, and K. Y. Pettersen. Integral LOS control for path following of underactuated marine surface vessels in the presence of constant ocean currents. In *47th IEEE Conference on Decision and Control (CDC)*, pages 4984–4991, Cancun, Mexico, 2008.

[4] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. Durrant-Whyte. Information based adaptive robotic exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 540–545, Lausanne, Switzerland, 2002.

[5] S. Butail and D. A. Paley. Three-dimensional reconstruction of the fast-start swimming kinematics of densely schooling fish. *Journal of the Royal Society Interface*, 9 (66):77–88, 2012.

[6] M. Cameletti, F. Lindgren, D. Simpson, and H. Rue. Spatio-temporal modeling of particulate matter concentration through the spde approach. *Advances in Statistical Analysis*, 97(2):109–131, 2013.

[7] L. Csató and M. Opper. Sparse on-line Gaussian processes. *Neural computation*, 14 (3):641–668, 2002.

[8] 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.

[9] M. Deffenbaugh, J. G. Bellingham, and H. Schmidt. The relationship between spherical and hyperbolic positioning. In *MTS/IEEE OCEANS'96*, volume 2, pages 590–595, Fort Lauderdale, FL, USA, 1996.

[10] L. Devries, S. J. Majumdar, and D. A. Paley. Observability-based optimization of coordinated sampling trajectories for recursive estimation of a strong, spatially varying flowfield. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 70(1-4):527–544, 2013.

[11] A. Dikarev, A. Griffiths, S. A. Watson, B. Lennox, and P. R. Green. Combined multiuser acoustic communication and localisation system for μauvs operating in confined underwater environments. In *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV)*, Girona, Spain, 2015.

[12] W. Dunbar and R. Murray. Model predictive control of coordinated multi-vehicle formations. In *41st IEEE Conference on Decision and Control (CDC)*, pages 1–15, Las Vegas, NV, USA, 2002.

[13] E. Fiorelli, N. E. Leonard, P. Bhatta, D. A. Paley, R. Bachmayer, and D. M. Fratantoni. Multi-AUV control and adaptive sampling in Monterey Bay. *IEEE Journal of Oceanic Engineering*, 31(4):935–948, 2006.

[14] T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control.* John Wiley & Sons, Hoboken, NJ, USA, 2011.

[15] T. I. Fossen, M. Breivik, and R. Skjetne. Line-of-sight path following of underactuated marine craft. In *Manoeuvring and Control of Marine Craft 2003 (MCMC 2003): A Proceedings Volume from the 6th IFAC Conference*, pages 244–249, Girona, Spain, 2003.

[16] G.-A. Fuglstad, F. Lindgren, D. Simpson, and H. Rue. Exploring a new class of non-stationary spatial gaussian random fields with varying local anisotropy. *Statistica Sinica*, 25:115–133, 2015.

[17] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pages 138–149, 2003.

[18] J. L. Garriga and M. Soroush. Model Predictive Control Tuning Methods: A Review. *Industrial & Engineering Chemistry Research*, 49:3505–3515, 2010.

[19] R. A. 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.

[20] S. J. Gershman, E. J. Horvitz, and J. B. Tenenbaum. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, 349 (6245):273–278, 2015.

[21] V. Gómez, S. Thijssen, A. Symington, S. Hailes, and H. J. Kappen. Real-time Stochastic Optimal Control for multi-agent quadrotor systems. *arXiv preprint arXiv:1502.04548*, 2016.

[22] N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993.

[23] R. Graham and J. Cortés. Adaptive information collection by robotic sensor networks for spatial estimation. *Automatic Control, IEEE Transactions on*, 57(6):1404–1419, 2012.

[24] 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.

[25] A. Hackbarth, A. Gray, and E. Kreuzer. Multi-Agent Motion Control of Autonomous Vehicles in 3D Flow Fields. *Proc. Appl. Math. Mech.*, 12(1):733–734, 2012.

[26] 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.

[27] A. Hansen and J. K. Hedrick. Receding Horizon Sliding Control for Linear and Nonlinear Systems. In *American Control Conference (ACC)*, pages 1629–1634, Chicago, USA, 2015.

[28] G. A. Hollinger and G. S. Sukhatme. Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33(9):1271–1287, 2014.

[29] J. Jonsson, J. Sundqvist, H. Nguyen, H. Kratz, M. Berglund, S. Ogden, K. Palmer, K. Smedfors, S. Wagner, and G. Thornell. Miniaturized submersible for exploration of small aqueous environments. In *MTS/IEEE OCEANS*, pages 1–8, Kona, HI, USA, 2011.

[30] H. J. Kappen. Linear theory for control of nonlinear stochastic systems. *Physical review letters*, 95(20):200201–1–200201–4, 2005.

[31] H. J. Kappen. Path integrals and symmetry breaking for optimal control theory. *Journal of statistical mechanics: theory and experiment*, 2005(11):1–25, 2005.

[32] H. J. Kappen. An introduction to stochastic control theory, path integrals and reinforcement learning. *Cooperative Behavior in Neural Systems*, 887:149–181, 2007.

[33] D. J. Klein, P. K. Bettale, B. I. Triplett, and K. A. Morgansen. Autonomous Underwater Multivehicle Control with Limited Communication: Theory and Experiment. In *2nd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*, Killaloe, Ireland, 2008.

[34] A. Krause. *Nonmyopic Observation Selection: Theory and Applications*. PhD thesis, Carnegie Mellon University Pittsburgh, PA, 2007.

[35] D. G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of Chemical, Metallurgical, and Mining Society of South Africa*, pages 201–215, 1951.

[36] K. Kwak, D. Park, W. K. Chung, and J. Kim. Underwater 3-D Spatial Attenuation Characteristics of Electromagnetic Waves With Omnidirectional Antenna. *Transactions on Mechatronics*, 21(3):1409–1419, 2016.

[37] L. Lapierre and B. Jouvencel. Robust nonlinear path-following control of an AUV. *Journal of Oceanic Engineering*, 33(2):89–102, 2008.

[38] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001.

[39] J. Le Ny and G. J. Pappas. Adaptive deployment of mobile robotic networks. *IEEE Transactions on Automatic Control*, 58(3):654–666, 2013.

[40] A. Ledergerber, M. Hamer, and R. D'Andrea. A robot self-localization system using one-way ultra-wideband communication. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3131–3137, Hamburg, Germany, 2015.

[41] T. Lee. *Computational geometric mechanics and control of rigid bodies*. PhD thesis, University of Michigan, 2008.

[42] T. Lee, M. Leoky, and N. H. McClamroch. Geometric tracking control of a quadrotor UAV on SE(3). In *49th IEEE Conference on Decision and Control (CDC)*, pages 5420–5425, Atlanta, GA, USA, 2010.

[43] A. M. Lekkas. *Guidance and path-planning systems for autonomous vehicles*. PhD thesis, Norwegian University of Science and Technology, 2014.

[44] A. M. Lekkas and T. I. Fossen. A time-varying lookahead distance guidance law for path following. *IFAC Proceedings Volumes*, 45(27):398–403, 2012.

[45] A. M. Lekkas and T. I. Fossen. A quaternion-based LOS guidance scheme for path following of AUVs. *IFAC Proceedings Volumes*, 46(33):245–250, 2013.

[46] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, 2007.

[47] X. Lin and S. Guo. Development of a spherical underwater robot equipped with multiple vectored water-jet-based thrusters. *Journal of Intelligent & Robotic Systems*, 67(3-4):307–321, 2012.

[48] 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(4):423–498, 2011.

[49] G. Loianno, C. Brunner, G. McGrath, and V. Kumar. Estimation, Control and Planning for Aggressive Flight with a Small Quadrotor with a Single Camera and IMU. *IEEE Robotics and Automation Letters*, 2016.

[50] K.-C. Ma, L. Liu, and G. S. Sukhatme. Informative Planning and Online Learning with Sparse Gaussian Processes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2102–2108, Daejeon, Korea, 2016.

[51] A. Marino and G. Antonelli. Experiments on sampling/patrolling with two autonomous underwater vehicles. *Robotics and Autonomous Systems*, 67:61–71, 2015.

[52] 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, WA, USA, 2015.

[53] 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, China, 2011.

[54] 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, China, 2014.

[55] V. Myers and D. P. Williams. A POMDP for multi-view target classification with an autonomous underwater vehicle. In *MTS/IEEE OCEANS*, pages 1–5, Seattle, WA, USA, 2010.

[56] S. Napora and A. D. Paley. Observer-Based Feedback Control for Stabilization of Collective Motion. *IEEE Transactions on Control Systems Technology*, 21(5):1846–1857, 2013.

[57] P. Naylor and N. D. Gaubitch. *Speech dereverberation*. Springer Science & Business Media, Dordrecht, Netherlands, 2010.

[58] S.-R. Oh and J. Sun. Path following of underactuated marine surface vessels using line-of-sight based model predictive control. *Ocean Engineering*, 37(2):289–295, 2010.

[59] B. Øksendal. *Stochastic Differential Equations*. Springer, Berlin, 2003.

[60] C. Osterloh, B. Meyer, A. Amory, T. Pionteck, and E. Maehle. Monsun II–towards autonomous underwater swarms for environmental monitoring. In *Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, 2012.

[61] D. A. Paley and C. Peterson. Stabilization of collective motion in a time-invariant flowfield. *Journal of Guidance, Control, and Dynamics*, 32(3):771–779, 2009.

[62] Y. Pan, E. A. Theodorou, and M. Kontitsis. Model-based Path Integral Stochastic Control: A Bayesian Nonparametric Approach. *arXiv preprint arXiv:1412.3038*, 2014.

[63] D. Park, K. Kwak, W. K. Chung, and J. Kim. Development of underwater distance sensor using EM wave attenuation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5125–5130, Karlsruhe, Germany, 2013.

[64] 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, Germany, 2015.

[65] D. Park, K. Kwak, W. K. Chung, and J. Kim. Development of underwater short-range sensor using electromagnetic wave attenuation. *Journal of Oceanic Engineering*, 41 (2):318–325, 2016.

[66] 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.

[67] B. L. Partridge. The Structure and Function of Fish Schools. *Scientific American*, 246(6):114–123, 1982.

[68] H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. CRC Press, Boca Raton, FL, USA, 2005.

[69] H. Rue and H. Tjelmeland. Fitting gaussian markov random fields to gaussian fields. *Scandinavian Journal of Statistics*, 29(1):31–49, 2002.

[70] A. Ryan and J. K. Hedrick. Particle filter based information-theoretic active sensing. *Robotics and Autonomous Systems*, 58(5):574–584, 2010.

[71] D. Simpson, F. Lindgren, and H. Rue. Think continuous: Markovian Gaussian models in spatial statistics. *Spatial Statistics*, 1:16–29, 2012.

[72] J.-J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall International Inc., Upper Saddle River, NJ, USA, 1991.

[73] M. S. Stanković and D. M. Stipanović. Extremum seeking under stochastic noise and applications to mobile sensors. *Automatica*, 46(8):1243–1251, 2010.

[74] L. V. Steenson, S. Turnock, A. Phillips, M. E. Furlong, C. Harris, E. Rogers, L. Wang, K. Bodles, and D. W. Evans. Model predictive control of a hybrid autonomous underwater vehicle with experimental verification. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 2014.

[75] N. Sydney and D. A. Paley. Multivehicle coverage control for a nonstationary spatiotemporal field. *Automatica*, 50(5):1381–1390, 2014.

[76] E. Theodorou, J. Buchli, and S. Schaal. A generalized path integral control approach to reinforcement learning. *Journal of Machine Learning Research*, 11(Nov): 3137–3181, 2010.

[77] E. Theodorou, J. Buchli, and S. Schaal. Reinforcement learning of motor skills in high dimensions: A path integral approach. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2397–2403, Anchorage, AK, USA, 2010.

[78] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, Cambridge, MA, USA, 2005.

[79] S. A. Watson. *Mobile platforms for underwater sensor networks*. PhD thesis, The University of Manchester, 2012.

[80] C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*. Cambridge, MA, USA, 2006.

[81] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou. Aggressive driving with model predictive path integral control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1433–1440, Stockholm, Sweden, 2016.

[82] 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.

[83] 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.

[84] 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, 2015.

[85] B. Zhang and G. Sukhatme. Adaptive Sampling for Estimating a Scalar Field using a Robotic Boat and a Sensor Network. In *IEEE International Conference on Robotics and Automation (ICRA)*, Roma, Italy, 2007.

Appendix

A Fundamental Concepts in Probability

In this appendix, the notation and basic concepts in probability that are used throughout this thesis are introduced.

Let X denote a random variable and x a specific value that X might be. The space of values that X can take may be discrete or continuous in measure. A probability density function (pdf) $p(\cdot)$ defined on X is simply written as $p(x)$. The pdf is positive for all x and it integrates (sums) to a total probability of one. Throughout this dissertation, the terms probability, probability distribution, and probability density function are used interchangeably and it is assumed that all continuous distributions actually possess densities.

The joint distribution of n variables, $p(x_1, \cdots, x_n)$ is denoted as $p(x_{1:n})$ or $p(\boldsymbol{x})$. For independent variables $x_{1:n}$, the joint distribution is given by

$$p(x_{1:n}) = \prod_{i=1}^{n} p(x_i). \tag{A.1}$$

The conditional probability $p(x \mid y)$ is defined by

$$p(x \mid y) = \frac{p(x, y)}{p(y)}, \tag{A.2}$$

with x being the dependent variable, given that y takes specific fixed values.

A joint probability distribution can be conditioned on other variables as well. For example, two independent variables x and y can be conditioned on a variable z and can be expressed as a product

$$p(x, y \mid z) = p(x \mid z)p(y \mid z). \tag{A.3}$$

This relation is known as conditional independence. Conditional independence does not imply absolute independence.

The chain rule

$$p(x_{1:n}) = p(x_1) \prod_{i=2}^{n} p(x_i \mid x_{1:i-1}) \tag{A.4}$$

expands a joint pdf in terms of conditional distributions.

The equation

$$p(x) = \int p(x \mid y)p(y)\mathrm{d}y \tag{A.5}$$

is often referred to as theorem of total probability.

Equally important is Bayes rule:

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)}. \tag{A.6}$$

It relates a conditional probability distribution of the type $p(x \mid y)$ to its "inverse" $p(y \mid x)$.

Appendix

B Normal Distribution

The normal distribution $\mathcal{N}$ plays a central role in this thesis. This appendix summarizes the main properties of the normal distribution. The monograph by Williams and Rasmussen [80] is recommended for a thorough introduction.

Consider the multivariate normal distribution for the joint random vector $\boldsymbol{x}' = (x'_1, \ldots, x'_k)$

$$f(\boldsymbol{x}'|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}} \cdot \exp\left(-\frac{1}{2}(\boldsymbol{x}' - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}' - \boldsymbol{\mu})\right), \tag{B.1}$$

which can be abbreviated as

$$\boldsymbol{x}' \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{B.2}$$

where $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\Sigma}$ is the variance. By partitioning the Gaussian random vector $\boldsymbol{x}'$ into the two jointly Gaussian random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ (B.2) becomes

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \boldsymbol{A} & \boldsymbol{C} \\ \boldsymbol{C}^\top & \boldsymbol{B} \end{bmatrix}\right). \tag{B.3}$$

The marginal distribution of $\boldsymbol{x}$ is

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{A}) \tag{B.4}$$

and the conditional distribution of $\boldsymbol{x}$ given $\boldsymbol{y}$ is

$$p(\boldsymbol{x} \mid \boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}_x + \boldsymbol{C}\boldsymbol{B}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_y), \boldsymbol{A} - \boldsymbol{C}\boldsymbol{B}^{-1}\boldsymbol{C}^\top). \tag{B.5}$$

Thus, the conditional expectation and the covariance matrix can be written as

$$\mathbb{E}(\boldsymbol{x} \mid \boldsymbol{y}) = \boldsymbol{\mu}_x + \boldsymbol{C}\boldsymbol{B}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_y) \tag{B.6}$$

and

$$\mathrm{Var}(\boldsymbol{x} \mid \boldsymbol{y}) = \boldsymbol{A} - \boldsymbol{C}\boldsymbol{B}^{-1}\boldsymbol{C}^\top. \tag{B.7}$$

Appendix

C Markov Models

In this appendix, Markov models with an emphasis on Markov fields are briefly introduced. A detailed description of Markov random fields is provided in [68].

C.1 Markov Chains

A Markov chain is a representation of a sequence of random variables $(x_1, x_2, \ldots, x_i)$ which can be specified by the conditional distribution $p(x_i \mid x_{i-1}, \ldots, x_1)$. A first-order Markov chain is illustrated in Fig. C.1. The underlying assumption

$$p(x_i \mid x_{i-1}, \ldots, x_1) = p(x_i \mid x_{i-1}) \tag{C.1}$$

is called first-order Markov assumption. It implies that x_{i+1} and x_{i-1} are conditionaly independent if x_i is given. However, conditional independence does not include independence. All foregoing variables are still linked implicitly.

Similarly, an n-th-order Markov assumption can be stated by

$$p(x_i \mid x_{i-1}, \ldots, x_1) = p(x_i \mid x_{i-1}, \ldots, x_{i-n})\,. \tag{C.2}$$

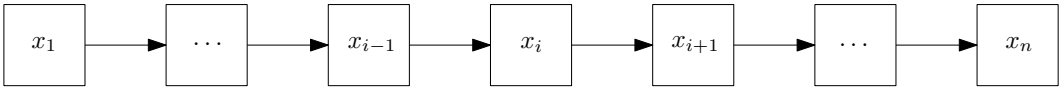


Figure C.1: First-order Markov chain.

C.2 Markov Random Fields

A Markov random field is described by an undirected graph $S = (V, E)$, where $V = \{1, \ldots, n\}$ is the set of the vertices and E is the set of the edges of the graph, in which the vertices and edges are related via a neighbouring system

$$N = \{N_i \mid \forall i \in V\}. \tag{C.3}$$

Here, N_i is a set of vertices neighbouring i in the graph toplogy. The properties for such a neighbouring system are

(1) $i \notin N_i$,

(2) $i \in N_j \leftrightarrow j \in N_i$.

For a regular lattice, the neighbouring system could be described as

$$N_i = \{j \in V \mid d(\boldsymbol{x}_j, \boldsymbol{x}_i) \leq r; j \neq i\}, \tag{C.4}$$

where $\boldsymbol{x}_j$ and $\boldsymbol{x}_i$ are the positions of the vertices j and i and $d(\cdot, \cdot)$ is a distance measure.

A Markov random field can be defined as a family of random variables $F = \{F_i, \ldots, F_n\}$, which are defined on the set of vertices V. Each variable F_i takes a value $f_i \in L$, where L is a label set, and the probability of F taking the value f_i is

$$p(F_i = f_i) = p(f_i). \tag{C.5}$$

For a Markov random field on V, the following two properties hold true:

(1) $p(f) > 0 \quad \forall f \in F$ (positivity),

(2) $p(f_i \mid f_{V-\{i\}}) = p(f_i \mid N_i)$l (Markovianity).

Positivity is directly derived from principles of probability theory, and the Markovianity depicts that, given all neighbours to i, the non-neighbours do not have any influence on the probability of i. In other words, labels without edges in between have to be conditionally independent. Thus, the condition

$$p(\boldsymbol{x}_i, \boldsymbol{x}_j \mid \boldsymbol{x}_{-\{ij\}}) = p(\boldsymbol{x}_i \mid \boldsymbol{x}_{-\{ij\}}) p(\boldsymbol{x}_j \mid \boldsymbol{x}_{-\{ij\}}) \tag{C.6}$$

has to be fulfilled if and only if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are not neighbouring each other.

C.3 Gaussian Markov Random Fields

A Gaussian Markov random field has the same structure as a Markov random field adding one requirement. The probability distribution of $F = \{F_i, \ldots, F_n\}$ is normally distributed with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The conditional independence between pairwise unconnected vertices, thus the information of the graph S, is included within the covariance $\boldsymbol{\Sigma}$. The inverse covariance matrix is the so-called precision matrix $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1}$. It turns out that

$$p(\boldsymbol{x}_i, \boldsymbol{x}_j \mid \boldsymbol{x}_{-\{ij\}}) = p(\boldsymbol{x}_i \mid \boldsymbol{x}_{-\{ij\}})p(\boldsymbol{x}_j \mid \boldsymbol{x}_{-\{ij\}}) \quad \Longleftrightarrow \quad \boldsymbol{Q}_{ij} = 0. \tag{C.7}$$

This property of Gaussian Markov random fields renders the precision matrix sparse, which yields an immense reduction of computational effort compared, for example, to Gaussian random fields.