Automated Planning, Sensing and Control for Autonomous Underwater Robotic Systems

Christos C. Constantinou

Department of Mechanical Engineering and
Material Science and Engineering

Cyprus University of Technology

A thesis submitted for the degree of

Doctor of Philosophy

December 2019

To my family

Acknowledgements

First of all, I would like to thank my advisor, Prof. Savva Loizou for the guidance and support that he provided during my PhD on both professional and personal matters. His positive spirit, continuous support, the discussion of new ideas have always been a source of inspiration. I have to admit that I feel privileged to have worked with him.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Anastasis Georgiades and Prof. Savvas Chatzichristofis.

Special thanks to Dr. George Georgiades for the discussions, and for his collaboration in the projects that we were working together and for being a good friend.

I would also thank Dr. Svetlana Potyagaylo for her collaboration and contribution to Aquabot project and Arestis Vrontis for his collaboration in Robodillos project.

My thanks also go to Prof. Daniel Toal, who gave me access to the research facilities during my visiting to the University of Limerick.

Thanks are also due to the Cyprus Research Promotion Foundation and the European Commission for partially supporting this work.

I would like to express my sincere thanks to my beloved wife Demetra for her patience, faith, and continuous support during the final stages of this PhD.

Last but not least, I would like to express my deepest gratitude to my parents, Costas and Eleftheria, my sister Marina and her daughters Eleftheria and Efthimia. This dissertation would not have been possible without their warm love, and endless support.

Christos C. Constantinou
December 2019

Abstract

Today's autonomous robotic systems have a significant impact on industrial applications and in academic disciplines. This dissertation considers a broad range of topics, from formal methods, sensor fusion, image processing, non-linear control to controller synthesis of motion tasks with applications in underwater, mobile and underground robots.

One of the most significant challenges in the robotics area lies in the area of motion and task planning. Motion planning is the robot able to move in the workspace while at the same time avoiding obstacles. On the other hand, task planning refers to the robot's ability to execute a specific task in the workspace. The main aim is to be able for a given task in a high-level language the robot to compile this specification into low-level descriptions in order to accomplish a task.

Autonomous underwater robots typically have to accomplish missions in an unknown and usually unstructured environment. The mission complexity grows considering to limited robot sensing systems as well as the limited on-line communications. For instance, GPS is not applicable due to the inefficient underwater electromagnetic transmission. In addition, vision-based systems are limited due to poor visibility in murky waters. The actuating system is usually composed of thrusters and control surfaces; all of them have non-linear dynamics and are strongly affected by the hydrodynamic effects. The ocean currents and flows imply additional difficulties for the ROV control system making the robot to deviate away from its desired state or path. Estimates of the flow velocity provided by various sensors or techniques may be incorporated into the control loop to compensate for the drift phenomenon. In this thesis, we address the problem of underwater visual inspection task as a combination of (i) a problem of localization and state estimation of the ROV with respect to the target by fusing information from different sources; (ii) a problem of control of an under-actuated underwater vehicle in the proximity to the target; (iii) a problem of full coverage of fishnet cages.

Autonomous multi-agent coverage of large-scale under-ground sewer networks is also addressing in this dissertation. Sewer network systems are typically dendritic networks converging in the downstream direction without closed loops. In network systems theory such networks are characterized as a tree or more precisely directed tree networks where the directionality is inherited from the sewage flow direction. Sewer network flow channels dimensions are typically restricted, allowing only a single inspection robot at a given position. Robots operating in such networks can only interchange positions at channel junctions. Wireless communications in underground sewer networks are much more challenging than in above-ground settings. The main transmission path is through the underground network's channels, usually non-line-of-sight and with severe attenuation over corridor bends and turns and there are also issues related to multi-path reflections.

The performance of the proposed methodologies is verified in realistic simulations in 2D and 3D virtual environments. Furthermore, extensive experimental validation on the actual hardware was carried out in a controlled environment at Robotics, Control, and Decision Systems (RCDS) laboratory and in the field (open sea) under real conditions.

Περίληψη

Τα αυτόνομα ρομποτικά συστήματα παίζουν σημαντικό ρόλο στις βιομηχανικές εφαρμογές και στους ακαδημαϊκούς κλάδους τα τελευταία χρόνια. Η παρούσα διδακτορική διατριβή εξετάζει ένα ευρύ φάσμα θεμάτων, από τις επίσημες μεθόδους, τη συγχώνευση αισθητήρων, την επεξεργασία εικόνων, τον μη γραμμικό έλεγχο και την σύνθεση αυτόματων ελεγκτών κυρίως για διαδικασίες κίνησης με εφαρμογές σε υποβρύχια, κινητά και υπόγεια ρομπότ.

Μία από τις σημαντικότερες προκλήσεις στον τομέα της ρομποτικής έγκειται στον τομέα της κίνησης και του προγραμματισμού των εργασιών. Ο σχεδιασμός κινήσεων έγκειται το ρομπότ να μπορεί μετακινηθεί στον χώρο εργασίας, αποφεύγοντας ταυτόχρονα εμπόδια. Από την άλλη πλευρά, ο σχεδιασμός εργασιών αναφέρεται στην ικανότητα του ρομπότ να εκτελέσει μια συγκεκριμένη διαδικασία στο χώρο εργασίας. Ο κύριος στόχος είναι το ρομποτ να μπορεί να εκτελέσει μια διαδικασία από μια γλώσσα υψηλού επιπέδου, να καταρτίσει αυτή την προδιαγραφή σε περιγραφές χαμηλού επιπέδου για να ολοκληρώσει μια διαδικασία.

Τα αυτόνομα υποβρύχια ρομπότ συχνά καλούνται να πραγματοποιούν αποστολές σε ένα άγνωστο και αδόμητο περιβάλλον. Η πολυπλοκότητα της αποστολής αυξάνεται λόγω των περιορισμένων συστήματων ανίχνευσης των υποβρυχίων ρομπότ και στις περιορισμένες επικοινωνίες. Για παράδειγμα, το ***GPS*** σύστημα δεν είναι εφαρμόσιμο εξαιτίας της δυσκολίας της ηλεκτρομαγνητικής μετάδοσης σε υποβρύχια περιβάλλοντα. Επιπλέον, το σύστημα όρασης του υποβρύχιου ρομπότ είναι περιορισμένο λόγω της χαμηλής ορατότητας σε θολά νερά και μακρινές αποστάσεις. Το σύστημα ενεργοποίησης αποτελείται συνήθως από προωθητήρες, οι οποίοι περιγράφονται από μη γραμμικά δυναμικά μοντέλα και επηρεάζονται έντονα από την υδροδυναμική. Τα ρεύματα και οι ροές των ωκεανών επιφέρουν πρόσθετες δυσκολίες για το σύστημα ελέγχου του ***ROV***, επηρεάζοντας με αυτό τον τρόπο το ρομπότ να αποκλίνει από την επιθυμητή κατάσταση ή διαδρομή. Οι εκτιμήσεις της ταχύτητας ροής που παρέχεται από διάφορους αισθητήρες ή τεχνικές μπορούν να ενσωματωθούν στο βρόχο ελέγχου για να αντισταθμιστεί το φαινόμενο της ολίσθησης. Σε αυτή τη διατριβή, αντιμετωπίζουμε το πρόβλημα της

υποβρύχιας οπτικής επιθεώρησης ως συνδυασμό : (ι) ενός προβλήματος εντοπισμού και εκτίμησης της κατάστασης του ***ROV*** σε σχέση με τον στόχο με τη συγχώνευση πληροφοριών από διαφορετικές πηγές αισθητήρων, (ιι) ένα πρόβλημα ελέγχου ενός υποεπενεργοποιημένου υποβρυχίου οχήματος προς τον εγγύς στόχο, (ιιι) πρόβλημα πλήρους κάλυψης/επιθεώρησης δίκτυα ιχθειοκαλειεργιών. Ένα άλλο πολύ σημαντικό πρόβλημα στην υποβρύχια ρομποτική είναι η συνεργασία μεταξύ πολλαπλών ρομπότ. Αυτό έχει ως πλεονέκτημα να επεκτείνει τη χρήση υποβρύχιων ρομποτικών συστημάτων. Οι κύριες προκλήσεις στην συνεργασία πολλαπλών υποβρυχίων συστημάτων είναι η χαρτογράφηση των ακτών ή θαλάσσιων ζωνών, μεταφορά αντικειμένων και εγκατάσταση αυτών σε μεγάλα βάθη και η κάλυψη περιοχών με σκοπό την εξερεύνηση ή την προστασία των θαλασσών και ωκεανών. Η συνεργασία μεταξύ των υποβρυχίων ρομποτικών συστημάτων είναι πολύ σημαντική όταν τα αντικείμενα πρέπει να εγκατασταθούν ή να μεταφερθούν κατά τη διάρκεια υποβρύχιων επιχειρήσεων.

Η αυτόνομη κάλυψη πολλαπλών πρακτόρων σε υπόγεια δίκτυα υπονόμων παρουσιάζεται επίσης σε αυτή τη διατριβή. Τα συστήματα δικτύου αποχέτευσης είναι συνήθως δενδριτικά δίκτυα. Τα κανάλια ροής δικτύου αποχέτευσης είναι χαρακτηριστικά γιατί σε πολλές περιπτώσεις επιτρέπουν μόνο ένα μόνο ρομπότ επιθεώρησης σε μια δεδομένη θέση λόγω των περιορισμένων διαστάσεων. Τα ρομπότ που λειτουργούν σε τέτοια δίκτυα μπορούν να ανταλλάξουν θέσεις μόνο στις διασταυρώσεις διαύλων. Οι ασύρματες επικοινωνίες σε υπόγεια δίκτυα αποχέτευσης είναι πολύ πιο απαιτητικές από ότι σε εφαρμογές στην επιφάνεια. Ο κύριος δίαυλος μετάδοσης γίνεται μέσω των καναλιών του υπόγειου δικτύου, συνήθως χωρίς οπτική επαφή και με σοβαρή εξασθένηση στις στροφές του διαδρόμου και, φυσικά, ζητήματα που σχετίζονται με τις ανταναклάσεις των διαδρομών.

Η απόδοση των προτεινόμενων μεθοδολογιών επαληθεύεται από ρεαλιστικές προσομοιώσεις σε εικονικά περιβάλλοντα δύο και τριών διαστάσεων ($2D$ και $3D$). Επιπλέον, η εκτεταμένη πειραματική επικύρωση του πραγματικού υλικού πραγματοποιήθηκε σε ελεγχόμενο περιβάλλον στο εργαστήριο Ρομποτικής, Αυτομάτου Ελέγχου και Συστημάτων Αποφάσεων και σε πραγματικό περιβάλλον (σε ανοιχτή θάλασσα) υπό πραγματικές συνθήκες.

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

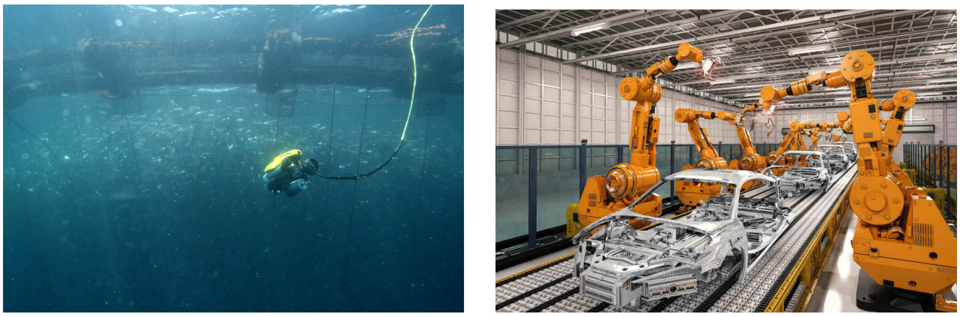
Introduction

Autonomous robots are making a considerable impact on many aspects of modern life such as industrial manufacturing, surgical robots, transportation, and exploration of the deep space and seas. As technology growth is continuous, more and more advances being made in the area of robotics. For example, embedded controllers (computers) have become more efficient and powerful in terms of size, capacity, processing power and the same time more affordable. Recently robotic development receives increased interest from both academia and industrial disciplines. Robots are now more reliable and they can be used in many applications supporting humans during complex operations and expanding our limitations giving new horizons for exploration and understandings such as space explorations, hazardous environments, and precision tasks. The figures below show robotic applications in every day life; Fig. 1.1(a) shows the amazon warehouse where robots transport the pallets for one location to another, helping in the classification of the products while in Fig. 1.1(b) an amazon drone delivers a package.

(a) Amazon warehouse (b) Amazon drone for package delivering

Figure 1.1: Amazon is using robots for warehouse applications

The automobile industry used industrial robot manipulators for the assembly car lines Fig. 1.2(a). Fig. 1.2(b) shows RCDS Lab's an underwater robot inspecting an aquaculture for the fisheries' industry needs.

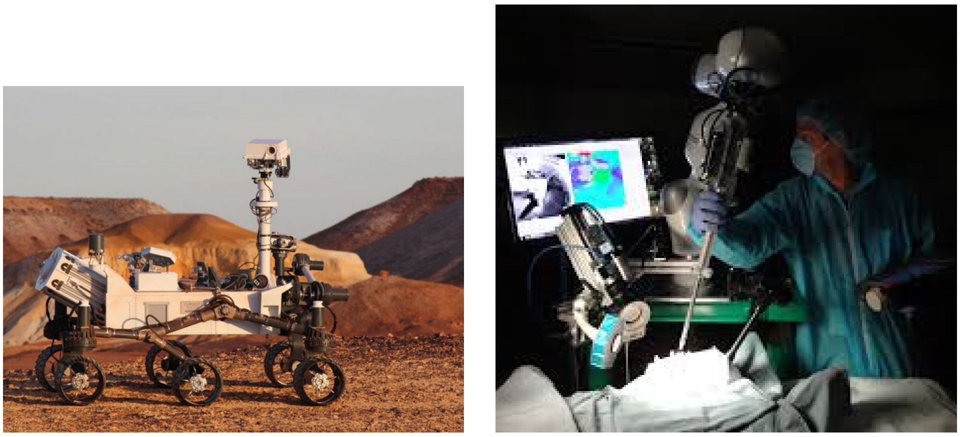


(a) RCDS Lab's underwater robot for aquaculture inspection

(b) Industrial manipulators in automobile industry

Figure 1.2: Examples for robots usage in industrial sector

Robots also help for planet exploration Fig. 1.3(a). Another growing area for robot applications is the medical robotic systems where they support doctors in surgery tasks Fig. 1.3(b). This dissertation covers a wide range of the robotics field from vision-based algorithms, sensor fusion techniques, controllers development to high-level motion task planning. Even though the major focus of this thesis is underwater robotic systems it is not limited to this area since the algorithms were being developed in a framework in order to satisfy the general area of robotics including mobile and underground robots.



(a) NASA Mars rover

(b) The Smart Tissue Autonomous Robot (STAR), Sheikh Zayed Institute For Pediatric Innovation

Figure 1.3: Robots for space exploration and medical sector

(a) Google autonomous car (b) Promobot service robot

Figure 1.4: Robots in every day life

1.1 Motivating Applications

The motivation of this work comes from real-world application problems in the robotics field. In this section, we are presenting the motivations of this work.

1.1.1 Motion and Task Planning

One of the most significant challenges in the robotics area lies in the area of motion and task planning [20]. Motion planning [69], [68] is the robot's ability to move in the workspace while avoiding the obstacles. On the other hand, task planning refers to the robot's ability to execute a specific task in the workspace. For example, you can imagine that you have several robots in your house and you want them to execute some tasks:

- Robot 1 could you bring me a cup of coffee;
- Robot 2 go and clean the kids' room;
- Robot 3 go outside and cut the grass

However, even for a simple task, its execution by a robot, is not trivial. A simple example is given below:

Example 1 *Consider a mobile robot that is located in specific workspace* $\mathscr{W}$ *and it has to move from an arbitrary initial position to the desired destination avoiding some obstacles in the environment, and visiting first the location B, then go to A and finally go to C as shown in Fig. 1.5.*

The Fig. 1.5 indicates the robot R as a blue triangle, the Region of Interests (RoI) in the workspace $\mathscr{W}$ with yellow cycles $RoI : \{A,\ B,\ C\}$ and the elliptic obstacles with red color O.

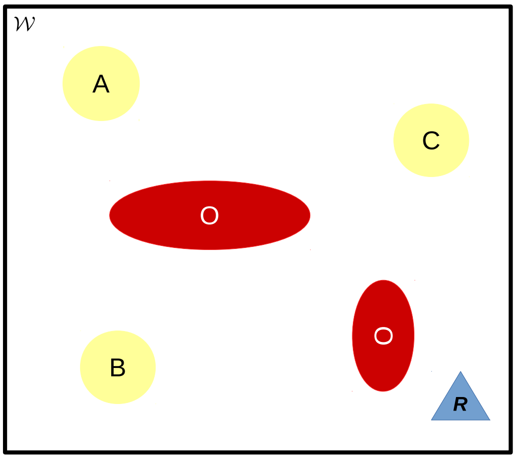


Figure 1.5: Simple example of robot motion task

In the Example. 1 the goal is to be able for a given task specification in a high-level language, the robot has to automatically translate this specification into a set of low-level motion controllers in order to accomplish the task Fig. 1.6. A high-level specification is given by a temporal logic formula [84] (i.e Linear Temporal Logic (LTL) [79]) over environmental and controller predicates. Based on the specification, a set of possible solutions to the problem can be constructed. An LTL formula can be translated in Buchi Automaton [98], [43] taking also into account the robot constraints. The robot control strategy can be implemented as a hybrid automaton [49] giving the control strategy for the robot.

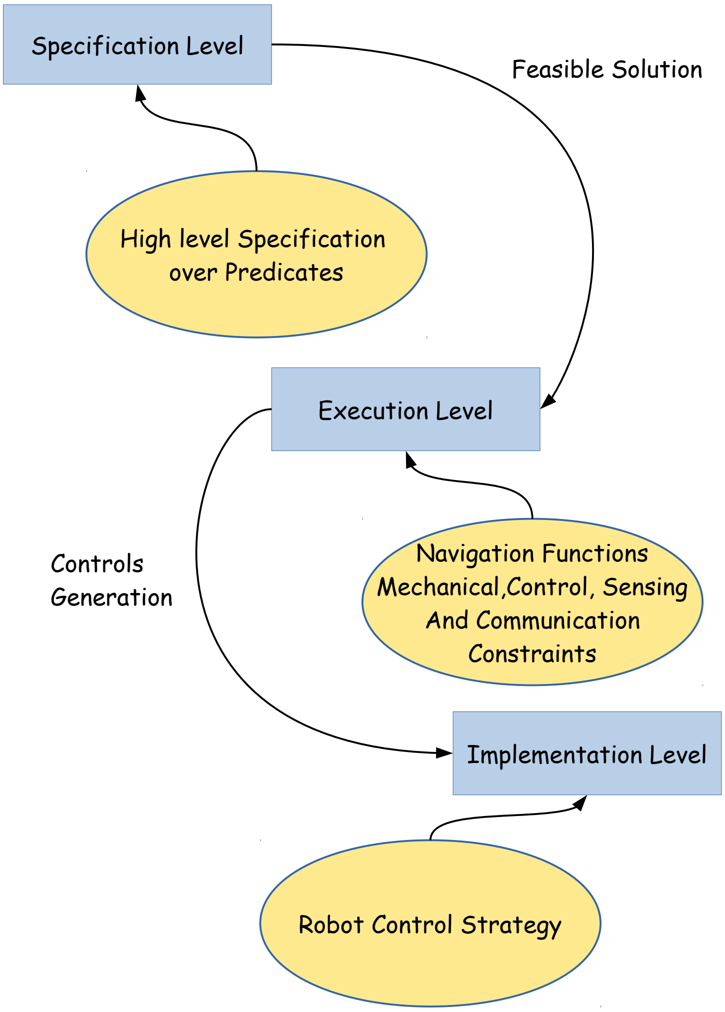


Figure 1.6: Motion Task Planning Scheme

1.1.2 Underwater Robotic Systems

Underwater robots are often complex systems that combine mechanical elements as propellers/thrusters, electromechanical devices such as motors, digital circuits such as processors and sensors, and software programs such as embedded controllers Fig. 1.7.

Autonomous underwater robots [106], [5] have to accomplish missions in an unknown and unstructured environment. The mission complexity grows with limitations in robot sensing and communications. For instance, GPS (Global Positioning System) is not applicable due to the inefficiency of underwater electromagnetic transmission. In addition, the vision-based system is limited due to poor visibility in murky waters (even in clean

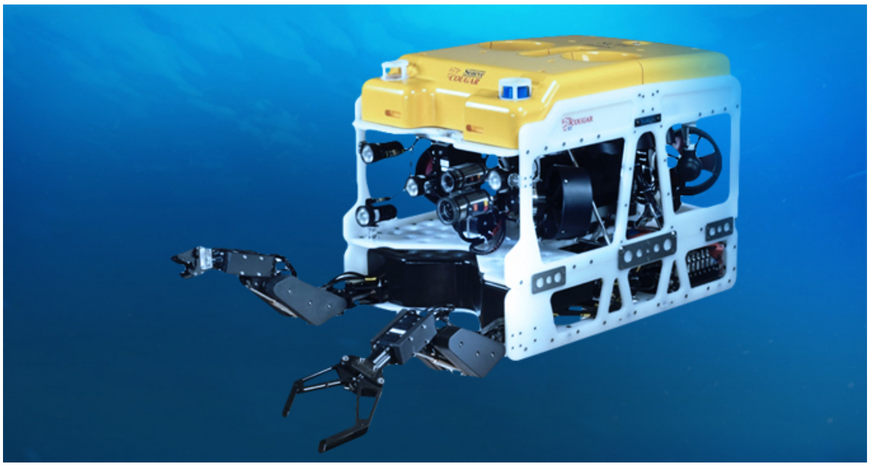


Figure 1.7: Saab Seaeye's Cougar XT

waters visibility is limited to several meters). The actuating system is usually composed of thrusters and control surfaces; all of them have non-linear dynamics and are strongly affected by the hydrodynamic effects. One of the major problems in underwater robotic systems is the localization task due to the communication constraints, and most importantly the absence of GPS. Due to limited communications of underwater vehicles, they are obliged to operate independently, without continuous human control, and in many cases, the vehicles should operate completely autonomously. AUVs and ROVs are currently used for scientific survey tasks, oceanographic sampling, underwater archeology, aquacultures inspection, and under-ice survey. Also, underwater vehicles are used for military applications, such as mine detection and more ambitious applications such as long-term undersea surveillance. Presently, AUVs are not typically used for sampling or manipulation tasks like those done routinely by ROVs, as typical work environments tend to be complex and challenging even to skilled human pilots.

The main sensors that usually are used by underwater robotic systems are as follows:

- Compass. A gyrocompass can provide an estimate of geodetic north accurate to a fraction of a degree. Magnetic compasses can provide estimates of magnetic north with an accuracy of less than 1°.

- Gyroscope. The term gyroscope denotes any instrument measuring inertial angular rotation;

- Inertial measurement unit (IMU). An IMU provides information about the vehicle's linear acceleration and angular velocity. These measurements are combined to form estimates of the vehicle's attitude including an estimate of geodetic (true) north from the most complex units;

- Depth sensor. The depth sensor measures the water pressure and gives the vehicle's depth. These estimates are reliable and accurate;
- Vision systems. Cameras can be used to obtain estimates of relative, and in some cases absolute, motion and used to perform tasks such as visual tracking of pipelines, visual inspection tasks, visual servoing or image mosaicking.

In this thesis, the contributions towards an underwater system suitable for visual inspection tasks for fishnets fault detection as follows:

- The development of advanced sensing for the underwater robot calculating the relative posture of the ROV with respect to the target with the accompanied software and hardware development.
- The localization and state estimation problem of the ROV with respect to the target by fusing information from different sources and sensors.
- The control of an under-actuated underwater vehicle in the proximity to the target.
- The full coverage of fishnet cages by the ROV.

1.1.3 Multi-Agent Coverage

Another interesting topic that is analyzed in the current work is the coverage task under communication constraints in an underground environment by multiple underground robots. Multi-agent coverage algorithms have been successfully applied by robotic systems in the past [40] in order to explore a specific area of interest. In underground environments like sewers and mining, the communication constraints problem between the agents remains a challenging problem. Distributed algorithms using Voronoi partitions and Lloyd's algorithm have been utilized in [27], [28] to tackle the problem. Solutions to the constrained coverage problem based on virtual potential fields were proposed in [85], [51]. A perimeter surveillance problem using a set of cooperative robots with heterogeneous speed capabilities under communication constraints is presented in [2] while an investigation into how a team of robotic agents can self-organize for the exploration of a building, subject to the constraint of maintaining line-of-sight communications for reactive multi-agent robotic teams is proposed in [7]. In [83] a multi-agent territory exploration task with communication constraints is presented.

1.2 List Of Publications

Journal Papers

1. C.C. Constantinou, S.G. Loizou and G.P. Georgiades, ”A Laser Vision System for Relative 3-D Posture Estimation of an Underwater Vehicle with Hemispherical Optics”. (under revision, IEEE Journal of Oceanic Engineering)
2. C.C. Constantinou, S. Potyagaylo, S.G. Loizou, ”Towards Multi- Sensor UKF-based Localization of an Underwater Robotic Vehicle for Aquaculture Inspection Operations”. (under submission)
3. C.C. Constantinou, S. Potyagaylo, S.G. Loizou, ”Modeling and Control for Underwater Inspection Operations”. (under submission)
4. C.C. Constantinou and Savvas G. Loizou, ”Automatic Controller Synthesis of Motion-Tasks with Real-Time Objectives for multi-agent system”. (under preparation)

Conference Papers

1. Christos C. Constantinou and Savvas G. Loizou, ”Cooperative manipulation task by multiple underwater robots”, (under preparation)
2. Christos C. Constantinou and Savvas G. Loizou, ”Automatic Controller Synthesis of Motion-Tasks with Real-Time Objectives”, IEEE Conference on Decision and Control (CDC), Miami, USA, December 17-19, 2018.
3. Savvas G. Loizou and Christos C. Constantinou,” Multi-Robot Coverage on Dendritic Topologies Under Communication Constraints”, 55th IEEE Conference on Decision and Control(CDC), Las Vegas, USA, December 12-14, 2016.
4. C.C. Constantinou, S.G. Loizou , and G.P. Georgiades, “A Laser Vision System for Relative 3-D Posture Estimation of an Underwater Vehicle to Mesh-like Targets”, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, October 9-14, 2016.
5. S. Potyagaylo, C.C. Constantinou, G.P. Georgiades, S.G. Loizou, “Asynchronous UKF-based Localization of an Underwater Robotic Vehicle for Aquaculture Inspection Operations”, OCEANS 2015, October 19-22, 2015.

6. C.C. Constantinou, S.G. Loizou, G.P. Georgiades, S. Potyagaylo, D. Skarlatos, “Adaptive calibration of an underwater robot vision system based on hemispherical optics”, IEEE Autonomous Underwater Vehicles (AUV), Oxford, MS, USA, October 6-9, 2014

1.3 Contributions and Outline

This dissertation comprises both theoretical and applied contributions in the field of Robotics and Autonomous Systems.

Fig. 1.8 shows the main contributions of this thesis in the field of robotics. The orange color denotes the contributions of this thesis in each area.

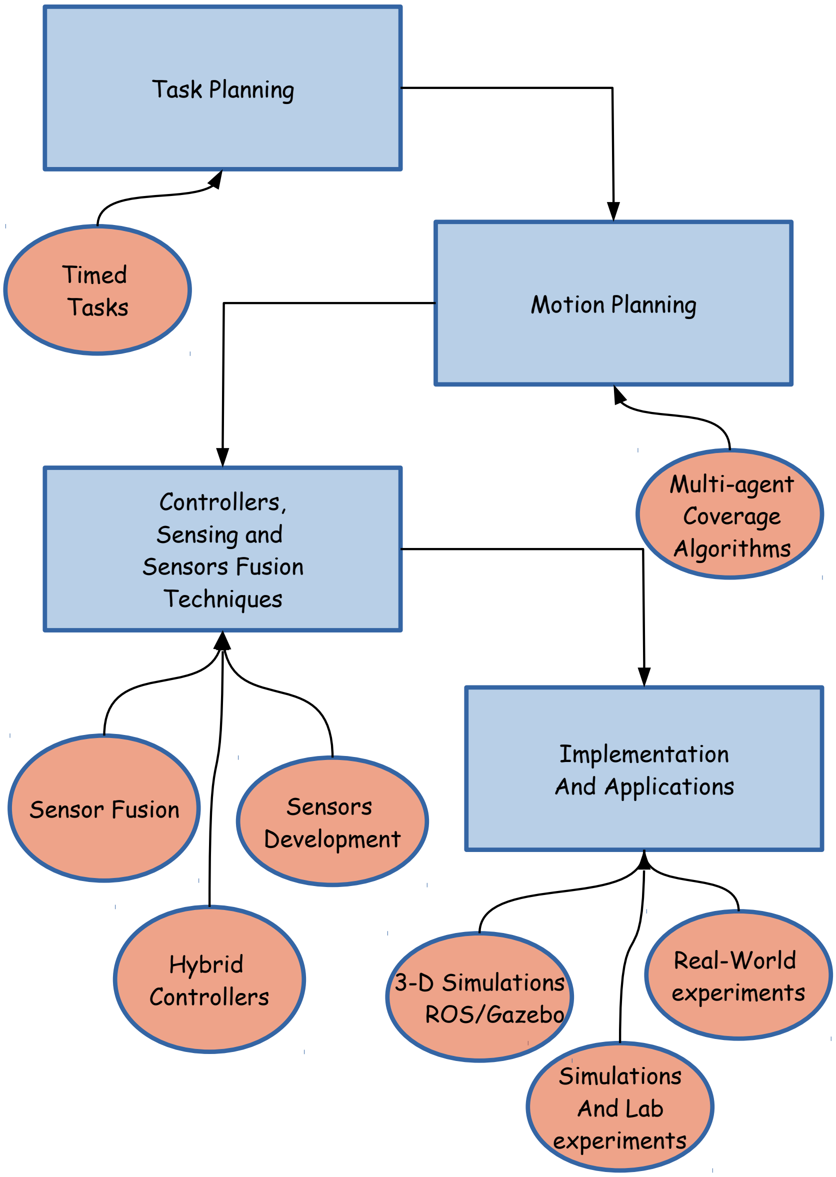


Figure 1.8: Contribution Scheme

In each chapter, we are introducing the basic notations and preliminaries of the proposed methodologies. The first part of this dissertation consisted of Chapter 2 where a novel methodology for automatic controller synthesis of motion-tasks with real-time objectives is presented. In the second part, three chapters present an underwater system suitable for undersea inspection operations. More specifically, in Chapter 3 a Laser Vision System (LVS) for relative 3-D posture estimation of an underwater vehicle with hemispherical optics is presented while in Chapter 4 a sensor fusion methodology for the underwater vehicle localization is proposed. Chapter 5 presents a hybrid controller for the aquaculture coverage inspection task. In the third part, Chapter 6 presents a multi-robot coverage methodology on dendritic topologies under communication constraints. Finally, conclusions and future work are provided in Chapter 7.

The outline of this dissertation accompanied with the contributions in each chapter is following below:

In Chapter 2 a novel approach for an automatic controller synthesis of motion tasks with real-time objectives is presented. One of the most significant challenges in robotics lies in the area of automatic motion-task planning. The main aim is to be able to specify a task in a high-level language and the robot compiles this specification into low-level descriptions in order to accomplish a task. A simple example is a robot being able to move from an initial position to the desired destination avoiding some obstacles or visiting specific regions while avoiding some other regions. New applications in the robotics field such as cooperative tasks from multiple robots and tasks that should be completed in a specific time interval bring new considerations that have to be addressed in the design of motion task planning algorithms, due to constraints, and uncertainties of the physical world. This chapter presents a novel methodology that is able to address some of these problems and in addition, allows the execution of timed tasks. More specifically, a novel methodology for synthesizing motion tasks with real-time objectives is presented.

- *Christos C. Constantinou and Savvas G. Loizou, "Automatic Controller Synthesis of Motion-Tasks with Real-Time Objectives", IEEE Conference on Decision and Control (CDC), Miami, USA, December 17-19, 2018.*

- *Christos C. Constantinou and Savvas G. Loizou, "Automatic Controller Synthesis of Motion-Tasks with Real-Time Objectives for multi-agent system". (under preparation)*

In Chapter 3 the development of one of the sensing modalities of a robotic visual inspection system for underwater inspection operations is presented. The work describes the development of an analytical model for three-medium refraction, that takes into account the non-linear hemispherical optics for image rectification. It also describes the development of a refractive index estimation method, for the medium external to the underwater vehicle's dome. The proposed system uses three line-lasers within the field of view of the camera, thus ensuring consistency in the reflections from mesh-like targets. The algorithms developed in this work provide appropriately filtered point-cloud data sets from each laser, as well as high-level information like distance and relative orientation of the target with respect to the ROV in real-time. In addition, an automatic calibration procedure, along with the accompanying hardware for the system has been developed in order to reduce the calibration overhead required by regular maintenance operations for underwater robots operating in sea-water. Furthermore, an image spatial filter was developed for discriminating between mesh and non-mesh like targets in the LVS measurements.

- *C.C. Constantinou, S.G. Loizou and G.P. Georgiades, "A Laser Vision System for Relative 3-D Posture Estimation of an Underwater Vehicle with Hemispherical Optics". IEEE Journal of Oceanic Engineering (under revision)*
- *C.C. Constantinou, S.G. Loizou , and G.P. Georgiades, "A Laser Vision System for Relative 3-D Posture Estimation of an Underwater Vehicle to Mesh-like Targets", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, October 9-14, 2016.*
- *C.C. Constantinou, S.G. Loizou, G.P. Georgiades, S. Potyagaylo, D. Skarlatos, "Adaptive calibration of an underwater robot vision system based on hemispherical optics", IEEE Autonomous Underwater Vehicles (AUV), Oxford, MS, USA, October 6-9, 2014.*

In Chapter 4 a methodology for localizing an autonomous tethered underwater robotic vehicle deployed for inspection operations in an underwater environment is presented. The developed methodology is based on an Unscented Kalman Filter (UKF) to fuse information from onboard sensors along with a priori knowledge about the aquaculture's geometry and the underwater robot's hydrodynamics. The proposed algorithm incorporates several modifications that take into account the sensors' measurements that are available asynchronously and with varying frequency. The performance of the proposed methodology is assessed through accuracy and consistency metrics generated from simulation benchmarks.

- *S. Potyagaylo, C.C. Constantinou, G.P. Georgiades, S.G. Loizou, "Asynchronous UKF-based Localization of an Underwater Robotic Vehicle for Aquaculture Inspection Operations", OCEANS 2015, October 19-22, 2015.*
- *C.C. Constantinou, S. Potyagaylo, S.G. Loizou, "Towards Multi- Sensor UKF-based Localization of an Underwater Robotic Vehicle for Underwater Inspection Operations". (under submission)*

The next Chapter 5 proposes the development of a robotic underwater system for aquaculture inspection tasks and algorithms and techniques for its localization and control in underwater GPS-denied environments. The localization algorithm uses information from several sources including an onboard inertial sensor, an onboard camera combined with line lasers and a priory knowledge about the aquaculture geometry. The control hybrid architecture includes an online estimator of the flow velocity around aquaculture provided by various sensors to compensate for the drift and necessary controllers required to track the virtual moving target. The proposed algorithms are validated through several simulations for small-scale aquaculture under a realistic level of noises and disturbances.

- *S. Potyagaylo, C.C. Constantinou, S.G. Loizou, "Visual-Inertial Control of an Underwater Vehicle for Underwater Inspection Operations", (under submission, conference paper)*
- *C.C. Constantinou, S. Potyagaylo, S.G. Loizou, "Trajectory Tracking Control of an Underwater Vehicle for Underwater Inspection Operations", (under submission, journal paper)*

Chapter 6 presents a novel algorithm for performing multi-robot coverage on networks with a dendritic topology where the communication topology is location-dependent and where the motion of each robot is constrained by the presence of the other robots in the network. The algorithm provides complete network coverage by the minimum number of robots, maintenance of communication constraints and robot collision avoidance. The minimum number of robots required for coverage is a by-product of the proposed algorithm. The efficiency of the algorithm is demonstrated through simulation studies.

- *Savvas G. Loizou and Christos C. Constantinou, "Multi-Robot Coverage on Dendritic Topologies Under Communication Constraints", 55th IEEE Conference on Decision and Control(CDC), Las Vegas, USA, December 12-14, 2016.*

Part II

Motion Task Planning

Chapter 2

Automatic Controller Synthesis of Motion-Tasks with Real-Time Objectives

In this chapter, a novel approach for an automatic controller synthesis of motion tasks with real-time objectives is presented. One of the most significant challenges in robotics lies in the area of automatic motion-task planning. The main aim is to be able to specify a task in a high-level language that can be automatically broken down into low-level controller descriptions in order to accomplish a task. A simple example is a robot moving from an initial position to the desired destination avoiding some obstacles or visiting specific regions while avoiding some other regions. New applications in the robotics field such as cooperative tasks from multiple robots and tasks that should be completed in a specific time interval bring new considerations that have to be addressed in the design of motion task planning algorithms. Since robots belong to our physical world are subject to physical laws, constraints, and uncertainties. This chapter presents a novel methodology that is able to address some of these problems and in addition, allows the execution of timed tasks. More specifically, a novel methodology for synthesizing motion tasks with real-time objectives is presented. The proposed methodology utilizes Linear Temporal Logic (LTL) to define the motion task sequencing. Timed motion objectives are handled by an underlying hybrid automaton that utilizes the concept of Navigation Transformation (NT) to provide a time-abstraction of the navigation tasks. This enables real-time execution of the navigation tasks with analytical guarantees on the safety and the execution time. The resulting system is correct-by-construction and the performance of the proposed methodology is demonstrated through non-trivial simulations. The results in this chapter were originally presented by the author in [24].

2.1 Introduction

One of the most interesting and challenging topics in the field of robotics is the motion task planning. During the last decade the use of formal methods has played a leading role in the development of the field. Formal high level languages such as Linear Temporal Logic (LTL) [89], [79], Computational Tree Logic (CTL) [21], Metric Temporal Logic (MTL) [64], and Metric Interval Temporal Logic (MITL) [4] have been successfully used for motion task specification and motion planning.

Several studies have tackled the problem of robot motion planning utilizing model checking techniques to satisfy formulas expressible in LTL such as in [34]. In [45], utilizing model checking, the authors present a systematic way for synthesizing a hybrid control strategy for motion and action planning for an autonomous robot under LTL task specifications. A fully automated framework for control of linear systems under LTL specifications is proposed in [62] and [95]. In [65] a methodology for controlling a group of robots satisfying a high-level user-specified behavior is proposed, where the behaviors are expressed in a subset of LTL, capturing reactive tasks. In [33], the authors present a design of closed-loop hybrid controllers that guarantee the generation of continuous robot trajectories that satisfy temporal specification but not real-time objectives. An efficient reactive controller synthesis for a fragment of linear temporal logic that can be used to specify common motion planning tasks such as safe navigation, response to the environment, surveillance, and persistent coverage is proposed in [103]. In [73] a methodology for automatically synthesizing motion tasks based on LTL specifications is described.

One important objective for the motion task planning problem is to be able to specify motion tasks with real-time constraints, and this is the main topic of the current work. Consider a scenario, where a robot is given a specific time frame to complete a task. For example, a worker robot has to continuously cycle through a motion task that requires going to work-cell B from anywhere on the work floor after exactly 20 seconds, picking up a package and then delivering it to work-cell A after exactly 10 seconds. Several works toward this goal have appeared in the literature. In [70], MTL is utilized for switching controller synthesis on finite-state abstractions of dynamical systems. In [77], [4], [14], the use of MITL is proposed, that leads to non-deterministic Timed Automata. The non-determinism complicates the task of controller synthesis problems and typically model checking techniques are utilized to derive a solution. In [110], [82] model checking techniques were utilized to find a feasible solution. The main aim of the current work is the development of a methodology which is able to tackle timed tasks in a real-time in a more efficient way (in terms of correctness by construction, absence of non-determinism related issues, and increased computational efficiency) compared to real-time languages (e.g. MTL, MITL), and without the need to resort to model checking techniques.

The main contributions of this work are :

- The Time Abstracting Hybrid Automaton (TAHA) gives us the advantage of combining the Navigation Transformation (NT) [71] with a fragment of LTL, the LTL_{-u}. The timed navigation problem and the task sequencing can thus be decoupled, providing simple and efficient solutions.
- The proposed methodology provides to a *correct by construction* solution for automatic synthesis of motion tasks problem with real-time objectives.

The concept of Navigation Transformation [71] which provides an analytically guaranteed time abstracted solution to the motion planning problem, provides the required machinery, to enable automatic controller synthesis approaches like [73], after appropriate adaptation, to tackle the problem of controller synthesis of motion tasks, with real-time objectives. The rest of this chapter is organized as follows: Section 2.2 presents preliminary notions and definitions while section 2.3 introduces the approach. Section 2.4 presents the simulation results and finally Conclusions are provided in section 2.5.

2.2 Preliminaries

In this section, we will introduce the necessary terminology and definitions for the development of the proposed methodology.

2.2.1 Fragment of Linear Temporal Logic (LTL_{-u})

A fragment of LTL that the "*Until*" $\mathscr{U}$ operator is not considered, is presented below. Note that $\phi_1 \mathscr{U} \phi_2$ guarantees that the ϕ_2 will eventually occur. In our case a weaker property is needed, which states that ϕ_1 holds continuously either until the next occurrence of ϕ_2 or throughout the sequence.

Definition 1 *An LTL_{-u} formula over the set P of atomic proposition is formed according to the following grammar:*

$$\phi \ ::= \ true \ | \ p \ | \ \phi_1 \wedge \phi_2 \ | \neg\phi \ | \bigcirc \ \phi \ | \ \phi_1 \mathscr{W} \phi_2$$

where $p \in P$. The Boolean connectors "conjunction" $\wedge$ and "negation" $\neg$, and two basic temporal modalities $\bigcirc$ (next) and $\mathscr{W}$ (Unless). The atomic proposition $p \in P$ stands for the state label p in a transition system.

From the basic LTL operators we can derive additional standard Boolean and temporal operators. More specific, we can derive the *"always"* $\square$ operator as follows:

$$\square\phi = \phi\mathscr{W}\mathbf{false}$$

Additionally, other operators such as $\vee$ "disjunction" and $\Rightarrow$ "implication" can be derived. From the set of atomic proposition P we can define the well formed formulas (wff) as follows:

- **true, false**, $p, \neg p$ are wff for all $p \in P$;
- if ϕ_1 and ϕ_2 are wff, then $\phi_1 \wedge \phi_2$ and $\phi_1 \vee \phi_2$ are wff;
- if ϕ_1 and ϕ_2 are wff, then $\bigcirc\ \phi_1$, and $\phi_1\mathscr{W}\phi_2$ are wff formulas;

The wff are interpreted over sequences of states $\sigma : \mathbb{N} \to 2^P$ and position j. For any $p \in P$, wff formulas $\phi_1,\ \phi_2$ and $i \in \mathbb{N}$:

- $(\sigma, j) \models p \quad \text{iff} \quad (\sigma, 0) \models p$
- $(\sigma, j) \models \neg p \quad \text{iff} \quad (\sigma, j) \not\models p$
- $(\sigma, j) \models \phi_1 \wedge \phi_2 \quad \text{iff} \quad (\sigma, j) \models \phi_1$ and $(\sigma, j) \models \phi_2$
- $(\sigma, j) \models \phi_1 \vee \phi_2 \quad \text{iff} \quad (\sigma, j) \models \phi_1$ or $(\sigma, j) \models \phi_2$
- $(\sigma, j) \models \bigcirc\ \phi_1 \quad \text{iff} \quad (\sigma, j+1) \models \phi_1$
- $(\sigma, j) \models\ \phi_1\mathscr{W}\phi_2$ iff $(\sigma, j)\ \models \phi_1\mathscr{U}\phi_2$ or $(\sigma, j)\ \models\ \square\phi_1$

The full syntax and semantics of the LTL formula can be found in App. A.1.

2.2.2 System Model

In this work we assume that the system is described by the first order holonomic kinematic model as:

$$\dot{x} = u \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the robot's position, and $u \in \mathbb{R}^n$ is the control input. Let W be the robot's workspace. The initial configuration of the robot is denoted as $x_0 \in \overset{\circ}{W}$, and the destination configuration as $x_d \in \overset{\circ}{W}$.

2.2.3 The Navigation Transformation

We will need some definitions from [71]. The Navigation Transformation is defined as follows.

Definition 2 *[71] A Navigation Transformation is a diffeomorphism* $\Phi : \overset{\circ}{W} \rightarrow \mathscr{P}^n$ *(* $\Phi : \overset{\circ}{W} \rightarrow \tilde{\mathscr{P}}^n$*), that maps the interior of the workspace to a point-world (with spherical boundary).*

Definition 3 *[71] (Time Abstraction of the Motion Planning Problem): Given a static workspace and a finite duration* $T > 0$*, determine (if it exists) the set of control actions* $u(t)$*,* $t \in [t_0, t_0 + T]$ *that will drive system (2.1) from any initial configuration* $x(t_0) = x_0$ *to a given destination configuration* $x(t_0 + T) = x_d$*, avoiding collisions.*

We will be using the time abstracting controller

$$u = u_{NT}(x, x_d, t, T)$$

that is proposed in Proposition 3 of [71] with the scheduling function $s_T(t)$ as defined in Definition 8 of [71]. This controller analytically guarantees that for almost all initial conditions, a time abstracted solution to the motion planning problem with duration T will be provided.

2.2.4 Hybrid Automaton

A Hybrid Automation is a dynamical system that describes the evolution in time of the values of a set of discrete and continuous variables [3], [16]. We will need the following definition:

Definition 4 *(Hybrid Automaton) A hybrid automaton* H *[16] is an eleven tuple* $H = (Q, X, E, U, f, \delta, Inv, guard, \rho, q_0, x_0)$ *where,*

- Q *is a set of discrete states or modes;*
- X *is a set of continuous state space (normally* $\mathbb{R}^n$*);*
- E *is a finite set of events;*
- U *is a set of admissible controls (normally* $U \subseteq \mathbb{R}^m$*);*
- f *is a vector field,* $f : Q \times X \times U \rightarrow X$*;*
- δ *is a discrete state transition function,* $\delta : Q \times X \times E \rightarrow Q$*;*

- *Inv is a set defining an invariant condition (also called domain), Inv* $\subseteq Q \times X$*;*
- *guard is a set defining a guard condition, guard* $\subseteq Q \times Q \times X$*;*
- ρ *is reset function,* $\rho : Q \times Q \times X \times E \to X$*;*
- q_0 *is an initial discrete state;*
- x_0 *is an initial continuous state.*

2.3 Approach

To better demonstrate the proposed approach, consider the following simple example: Assume a robot that has to navigate from its current location to x_d^i in T^i time units, in an obstacle cluttered environment. Let predicate $\mathscr{C}_{x_d^i,T^i}$ be active when controller $u = u_{NT}(x, x_d^i, t, T^i)$ is fulfilling the above task. The set of predicates P can be defined as :

$$P = \mathscr{C}_{x_d^i,T^i} \tag{2.2}$$

where $\mathscr{C}_{x_d^i,T^i}$ is the controller predicates parameterized with the goal x_d^i and time constraints T^i.

2.3.1 The Time Abstracting Hybrid Automaton (TAHA)

From *Def.* 4 we can derive the TAHA $\mathscr{T}$ as illustrated in Fig. 2.1. Let $q \in Q$, $\mathbf{z} = \left[x(t)^\top,\ \tau(t)\right]^\top \in X$ where $\tau(t)$ denotes a real-time clock. In the starting state, $q = 0$ and $\mathbf{z}$ remains unchanged since $\dot{\mathbf{z}} = f(0, \mathbf{z}, \mathbf{0}) = \mathbf{0}$. A transition to $q = 1$ can only occur when an exogenous event $\alpha \in E$ is activated. The transition function δ at this location is defined as:

$$\delta(0, \mathbf{z}, e) = \begin{cases} 1 & \text{if } e = \alpha \\ 0 & \text{otherwise} \end{cases}$$

to indicate that a transition from state 0 to 1 takes place when event α occurs independent of the values of $\mathbf{z}$. When in state $q = 1$, the real-time clock is started with $\dot{\tau} = 1$ and $\dot{x} = u$, i.e. $\dot{\mathbf{z}} = f(1, \mathbf{z}, u) = [u^\top\ 1]^\top$, where the time abstracting controller u is parameterized with the destination configuration and the duration T, i.e. $u = u(x_d, T)$. The invariant condition $[\tau < T]$ associated with $q = 1$ is satisfied while the system is in this state. The guard condition $\tau = T$ is activated when the time is out in state $q = 1$. The guard indicates that the operation is finished at the pre-defined time and the system has to move in another state. This acts as an endogenous time out event which causes the transition to state 2. The navigation task completion at the specific time instant is guaranteed by the underlying time abstracting controllers u. The state transition function in this location takes the form:

$$\delta(1,\mathbf{z},e) = \begin{cases} 2 & \text{if } \tau = T \\ 1 & \tau < T \end{cases}$$

The reset condition is $\tau' = 0$ while the real-time clock is stopped in state 2.

In state 2, the system is waiting an exogenous event $\beta \in E$. When the event β occurs then the automaton goes back to the state 0 and it is waiting for event α to repeat the procedure.

$$\delta(2,\mathbf{z},e) = \begin{cases} 0 & \text{if } e = \beta \\ 2 & \text{otherwise} \end{cases}$$

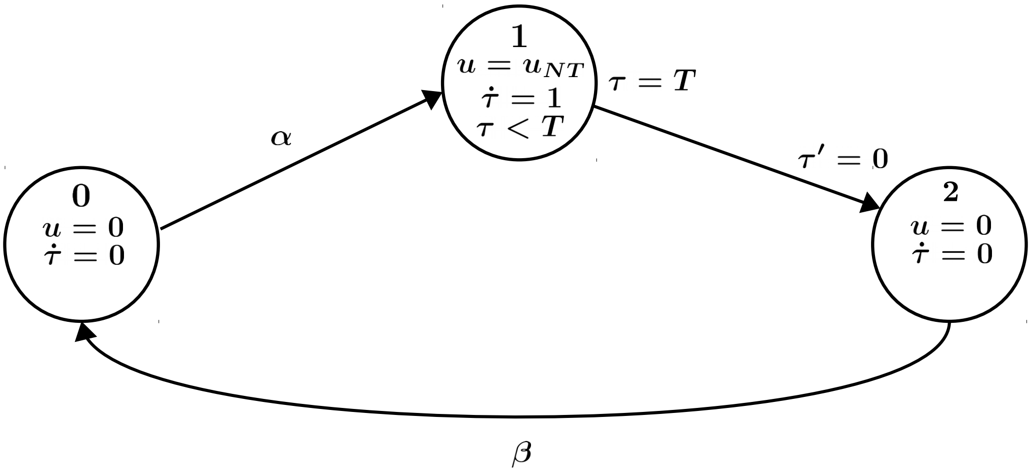


Figure 2.1: The Time Abstracting Hybrid Automaton (TAHA) $\mathscr{T}$

2.3.2 Predicates and Formulas

Since the current work deals with single robot motion task planning, a predicate p_i signifying motion to a specific destination configuration (within a specific time) should exclude any other destination configuration predicate being true at the same time. Hence we define the individual destination predicate P_i as:

$$P_i = p_i \bigwedge_{j \neq i} \neg p_j \tag{2.3}$$

Let $\mathbb{P} = \{P_1, \ldots, P_n\}$ be the individual destination predicate set. The following formula is useful for defining sequences:

$$Seq(\mathbb{P}) = P_1 \wedge \bigcirc(P_2 \wedge \bigcirc(\ldots \wedge \bigcirc P_n)_{\ldots})$$

Define the set rotation operator:

$$rot_i(\mathbb{P}) = \{P_i, \ldots, P_n, P_1, \ldots, P_{i-1}\}$$

where $|rot_i(\mathbb{P})| = |\mathbb{P}|$, where $|\cdot|$ denotes the cardinality of the set. The following formula is useful for defining cycles:

$$Cycle(\mathbb{P}) = \bigvee_{i \in \{1 \ldots |\mathbb{P}|\}} Seq(rot_i(\mathbb{P})) \tag{2.4}$$

2.3.3 LTL Specifications to Büchi Automaton

Definition 5 *A Büchi Automaton (BA) is a tuple $A = (S, \Sigma, \delta_b, S_0, F)$ where:*

- *S is the finite set of states,*
- *Σ is a finite alphabet,*
- *$S_0 \subseteq S$ is a set of initial states,*
- *$\delta_b : S \times \Sigma \to S$ gives the set of transitions and,*
- *$F \subseteq S$ is a set of accepting states*

The language accepted by such automaton is $L(A) = \{\sigma \mid$ there is a run ρ over σ such that $inf(\rho) \cap F \neq \emptyset\}$. A language $L \subseteq \Sigma^\omega$ is said to be $\omega-regular$ if it is accepted by some Büchi automaton.

Several LTL to Büchi automaton translators are available in the literature, e.g. [32], [41]. Following similar arguments and analysis as in [73], define the driving function:

$$\Delta \ : \ \Sigma \times Q \to \mathbb{R}^n \times \mathbb{R}^+ \times E$$

where $E = \{\alpha, \beta\}$ is the TAHA driving component. We can state the following: The driving function here is used to activate transitions in the BA and TAHA based on the outgoing labels available at the current state of BA and the current state of TAHA. The system architecture is depicted in Fig. 2.2. In each state $i \in S$ of a BA, let $l_{s,i} \subset \Sigma$ be the set of available labels (since BA can be non-deterministic). Let $l_i = l_{s,i}[1]$ be the first element of $l_{s,i}$. Each label l_i corresponds to a destination predicate $p_i = \mathscr{C}_{x_d^i, T^i}$. Let $X_d(l_i) = x_d^i$, and $T(l_i) = T^i$. Then the transition function is defined as follows:

$$\Delta(l_i, \mathbf{0}) = \left[\mathbf{x_d^i},\ \mathbf{T^i},\ \alpha\right]$$

$$\Delta(l_i, \mathbf{2}) = \left[\mathbf{x_d^i},\ \mathbf{T^i},\ \beta\right]$$

The system starts with all controller predicates being false. In each state the sequence generator, using the driving function provides the controller parameters and appropriate events to TAHA. Every time the driving function returns a β event the sequence generator converts the corresponding transition label to truth enabling the transition to the next state of the BA.

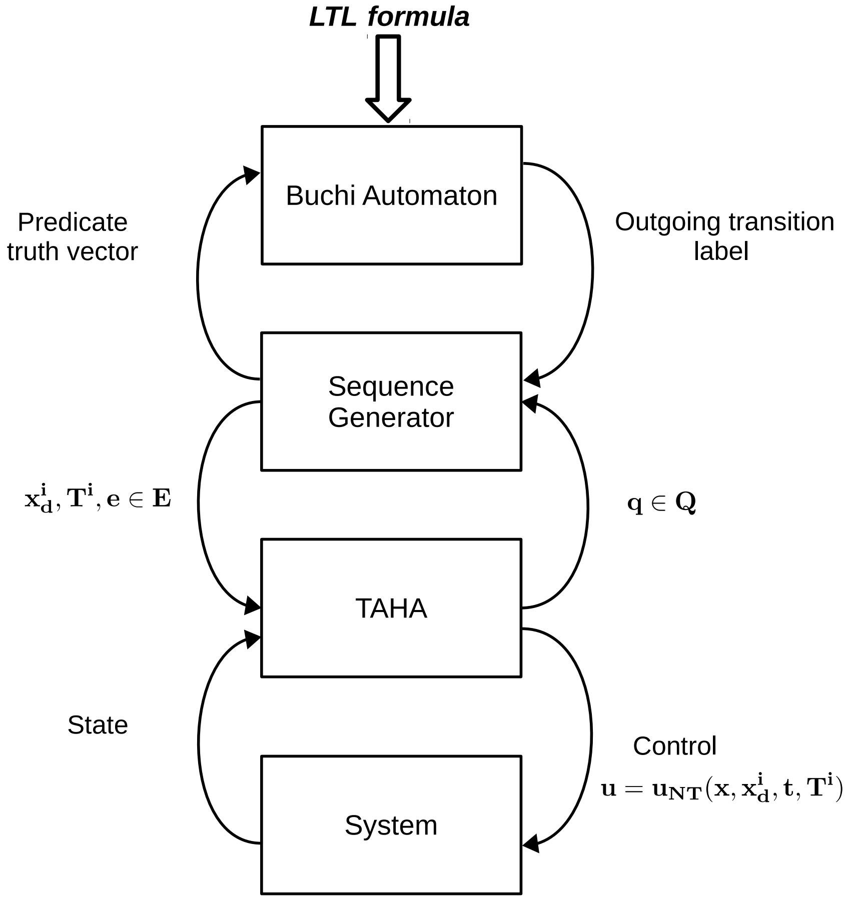


Figure 2.2: Information flow diagram for the proposed architecture

2.4 Simulation Results

In order to demonstrate the effectiveness of our methodology, we have set up a simulation. In the following case study, let us consider the environment with obstacles as shown in Fig. 2.4.

Assume we have a surveillance robot that has to continuously cycle through a motion task that requires going to location A from anywhere on the workspace after exactly 5 seconds, and then visiting location B after exactly 8 seconds, and then going to location C in 3 seconds and finally returning to location A in exactly 5 seconds. Note that kinematic constraints are beyond the scope of the current work.

The table I shows the destination configuration and the duration tasks. Consider the following LTL_{-u} formula :

	$X_d(P_i)$	$T^i(P_i)(sec)$
p_1	[-4, 1] (location A)	5
p_2	[3, -1] (location B)	8
p_3	[2, 1] (location C)	3

Table 2.1: Destination configuration and the duration

$$\phi = P_1 \wedge \Box\, Cycle(\mathbb{P}) \tag{2.5}$$

The resulting Büchi automaton from the formula above according Eqs. 2.3 and 2.4 and the tasks table 2.1 is shown in Fig. 2.3.

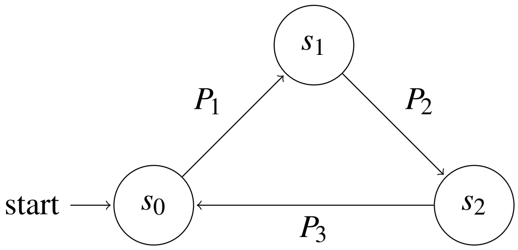


Figure 2.3: The resulting Büchi automaton under ϕ

where P_1, P_2, P_3 can be derived from Eq. 2.3 as follow:

$$P_1 = (p_1) \wedge \neg(p_2) \wedge \neg(p_3)$$
$$P_2 = (p_2) \wedge \neg(p_1) \wedge \neg(p_3)$$
$$P_3 = (p_3) \wedge \neg(p_1) \wedge \neg(p_2)$$

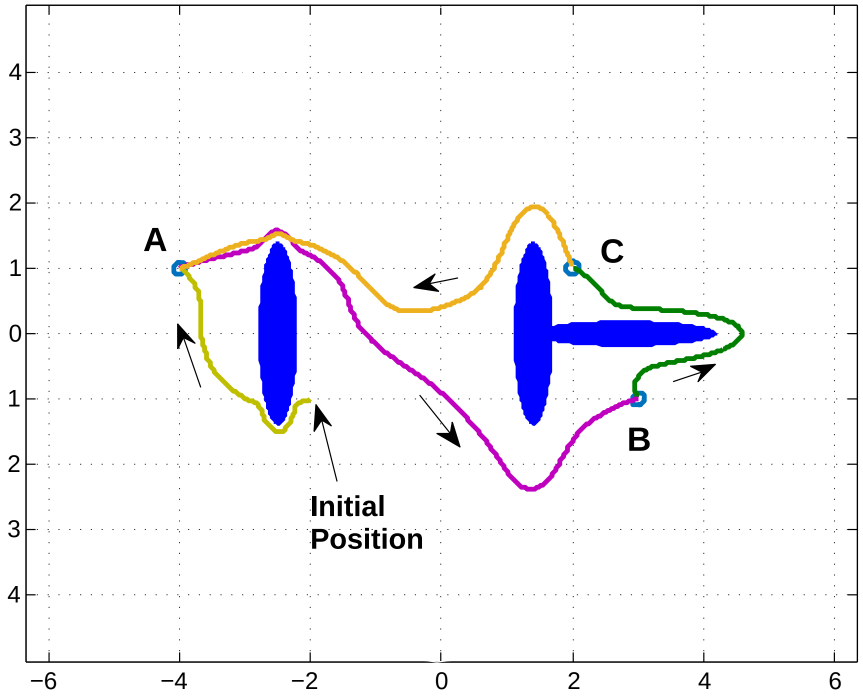


Figure 2.4: The complete trajectory of robot

The Fig. 2.4 shows the trajectory of the robot avoiding the obstacles in the workspace. The arrows denote the direction of the robot and the A, B and C are the goals location that robot cycles through in specific time durations. The blue ellipsoidal shapes illustrate the obstacles. The simulation results show that the robot could successfully complete the tasks under real time specifications.

Figs. 2.5-2.8 illustrate the time that the robot needs to move to each location. More precisely, Figs. 2.5-2.8 show that the robot can successfully satisfy the motion task with real-time constraints i.e. (move from a random initial position in workspace to the goal destination A, avoiding obstacles in specific time).

2.5 Conclusions

In this chapter, we propose a novel methodology for synthesizing motion tasks with real-time objectives. The main advantages of the presented methodology is a correct by construction solution and the utilization of the Navigation Transformation by a fragment of LTL. This approach decouples the time abstracted navigation problem from the task sequencing problem, simplifying the solution. The proposed methodology was verified by

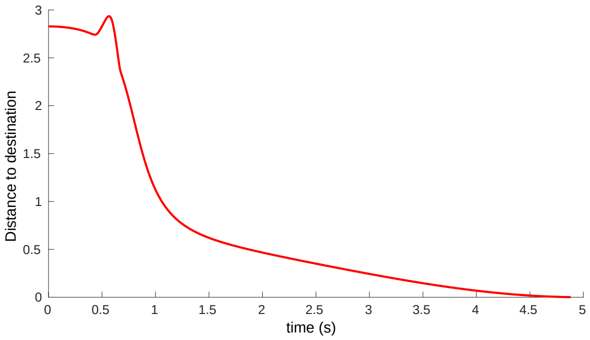


Figure 2.5: Initial position to A

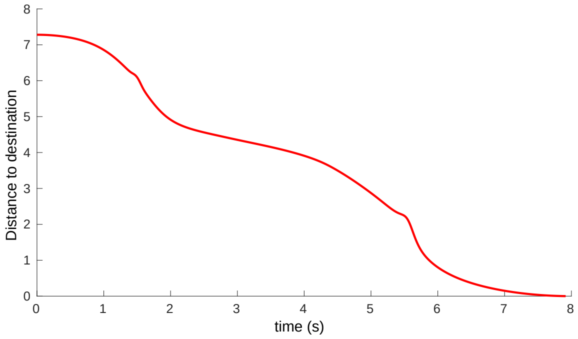


Figure 2.6: A to B

non-trivial simulation results.

Further research issues include addition of allowed and forbidden regions in the set of predicates, agents with input constraints, and real world hardware experiments. Moreover, cooperative tasks with multiple agents are under investigation.

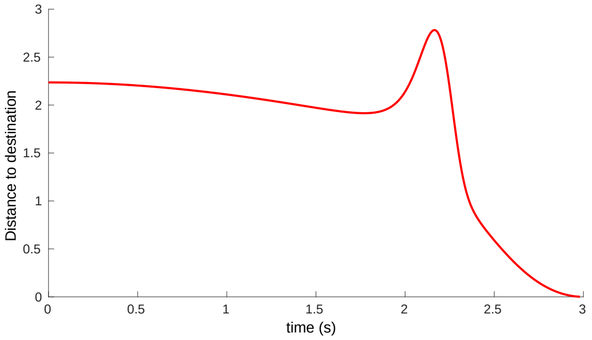


Figure 2.7: B to C

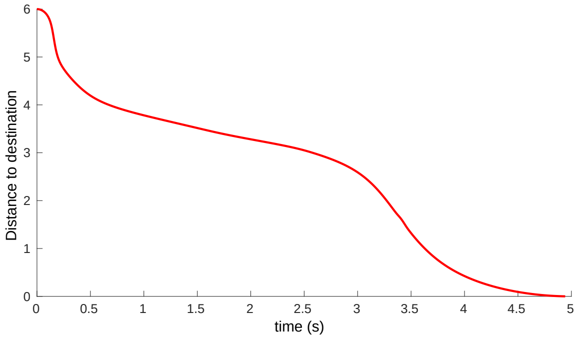


Figure 2.8: C to A

Part III

Underwater Robotic Systems

Chapter 3

A Laser Vision System for Relative 3-D Posture Estimation of an Underwater Vehicle with Hemispherical Optics

This chapter presents a novel Laser Vision System (LVS) for underwater vehicles. It is suitable for measuring the relative posture from both mesh-like and solid targets in underwater environments. The system was developed in the framework of the AQUABOT project [6][1], a research project dedicated to the development of an underwater robotic system for inspection of offshore aquaculture installations. The work describes the development of an analytical model for three-medium refraction, that takes into account the non-linear hemispherical optics for image rectification. It also describes the development of a refractive index estimation method, for the medium external to the underwater vehicle's dome. The proposed system uses three line-lasers within the field of view of the camera, thus ensuring consistency in the reflections from mesh-like targets. The algorithms developed in this work provide appropriately filtered point-cloud data sets from each laser, as well as high-level information like distance and relative orientation of the target with respect to the ROV in real-time. In addition, an automatic calibration procedure, along with the accompanying hardware for the system has been developed in order to reduce the calibration overhead required by regular maintenance operations for underwater robots operating in sea-water. Furthermore, an image spatial filter was developed for discriminating between mesh and non-mesh like targets in the LVS measurements. Finally, a set of experimental results in controlled laboratory environment as well as in real conditions at offshore aquaculture installations demonstrate the performance and robustness of the system. The results in this chapter were originally presented by the author in [26], [25] and [22].

---

[1] This work was supported by the European Regional Development Fund and the Republic of Cyprus through the Research Promotion Foundation under research grant ΑΕΙΦΟΡΙΑ/ΓΕΩΡΓΟ/0311(ΒΙΕ)/08.

3.1 Introduction

Underwater Robotics have received an increasing interest from research and industry during the last years. Currently, underwater robotic vehicles are used in a wide range of applications ranging from exploration and mapping of underwater environments to monitoring and inspection of undersea infrastructures such as pipes and ships [105], [102] and [12]. Underwater operations typically raise more challenges than equivalent ground or air operations. Typical examples of such challenges relate to communications, localization, multiple mediums inside and outside the robot and also medium variations with the depth, temperature and presence of pollutants, mechanical strength and operation issues, low visibility, corrosive environment, etc.

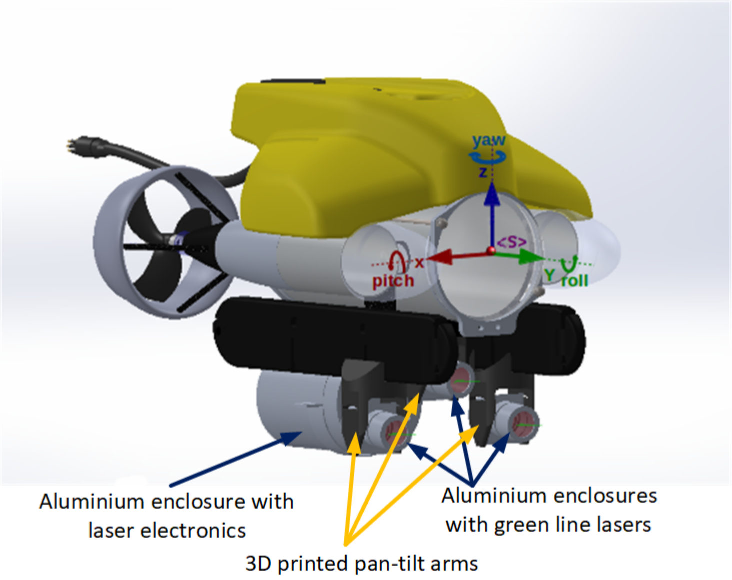
The motivation of this work was the development of one of the sensing modalities of a robotic visual inspection system for offshore aquaculture installations. The particular modality, that is based on visual information, enables the relative posture estimation of the underwater robot with respect to the aquaculture fish-net without having to introduce any modifications on the aquaculture's structure.

The main contributions of this work are:

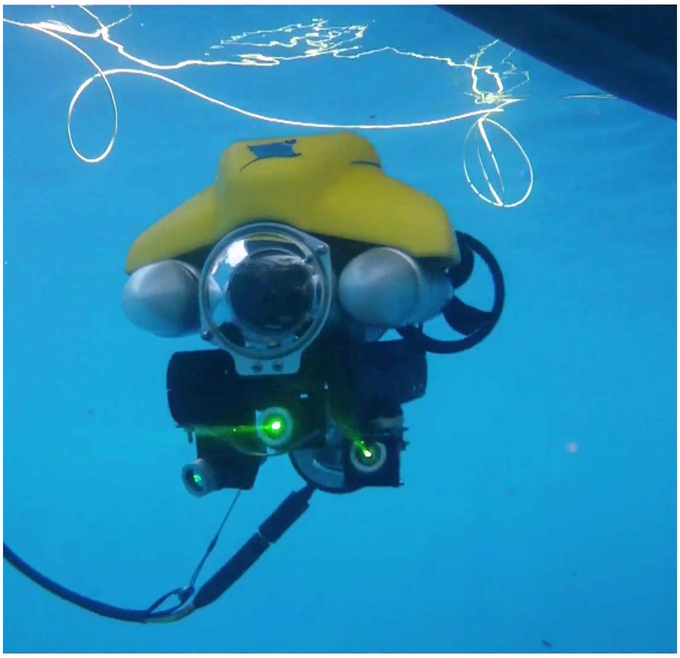
1. An analytical model for a three-medium refraction that takes into account the non-linear hemispherical optics for image rectification and refractive index estimation of the external medium.
2. An automatically calibrated Laser Vision System (LVS) suitable for measuring the relative posture from both solid and mesh-like targets in underwater environments.
3. A spatial filter for discriminating LVS measurements from mesh-like structures and measurements from other artifacts in the environment.

The problem of the camera calibration in air has been widely investigated in several research works [97], [48]. In underwater environments, implementation of these techniques is non-trivial due to multi-medium refraction, particularly when cameras are housed. An examination of the flat refractive geometry for underwater applications is presented in [96]. The refractive index determines how much light is bent, or refracted, when light moves from one medium to another. In [55] a camera calibration method is presented, including a parameterization of a flat port underwater housing. The method used an analysis by synthesis approach in order to become independent from errors in corner detection for the geometric parameters. However, underwater robots are typically designed with hemispherical domes to be able to mechanically withstand high pressures in deep waters. In addition, this design mitigates the effects of hydrodynamics by reducing the drag coefficient. In [66] it is analyzed why the perspective model is not valid for hemispherical optics and a comparative study of the errors induced by refraction is provided, when cameras are mounted behind hemispherical or planar air/water interfaces.

Structured light systems are widely used in vision-based systems to perform a wide range of applications such as 3D reconstruction, scanning and range measurements [93] and [94]. Previous studies describe underwater ranging systems which project a laser stripe or line [29], [58], [100]. In [58] a methodology for defining the position vector of an ROV is proposed, using its own camera signal and the information provided by two laser pointers. In [29] a methodology of orientation estimation is also introduced, projecting a laser stripe on the image plane. A low cost underwater laser range-finder based upon a simple camera and parallel laser line setup is proposed by [15], where the distance calculation is based on the pinhole camera model. In [46] a solution that utilizes each laser independently is presented, thus a range estimate is achieved for each laser. A stereo structured light system is proposed in [74] for underwater inspection operations. More specifically, two methods for calibrating a stereo structured light system for perception in dry or underwater environments are presented. An underwater LVS is proposed by [81] using a single laser pointer when the camera is housed by a dome. Nevertheless, the authors do not consider the refractive index problem with the restrictive assumptions on the setup, that the camera lens must be mounted at the center of the dome and no camera tilting is accounted for.



(a) CAD Model of ROV.



(b) Integrated System with Laser Vision System.

Figure 3.1: Design and Integration of the LVS on the Videoay Pro-4 platform.

This work tackles the development of an analytical model for the hemispherical optics physics, that describes the path of light rays which are refracted through three different interfaces; air, acrylic(dome) and water. This is required for appropriately interpreting the laser reflection images from an array of line lasers and producing the relative posture of the robot with respect to a mesh-like target, that can then be utilized for underwater localization, tracking and navigation tasks. The developed methodology provides the capability of on-line estimation of the refractive index of the external medium and on-line adaptation of the model to the estimated refractive index. This is particularly useful in operational scenarios where the robot is operating close to aquacultures (due to the presence of dissolved/liquid biomass in their proximity) or for the detection of leaks or pollutants that affect the refractive index of sea water. Utilizing the analytical model, a methodology is proposed to automate the LVS calibration process in air, a task that is typically required after every maintenance cycle. Determining the posture with respect to mesh-like structures is non-trivial in environments where additional artifacts cause laser reflections (e.g. fish, air-bubbles, undissolved waste products, etc.). To this extend we developed a spatial filter for discriminating reflections from mesh and non mesh-like structures in order to correctly determine the relative posture of the ROV.

The developments presented in this work were carried out for partially fulfilling the needs of the AQUABOT project [6]. A VideoRay Pro-4 ROV was used as the base platform for the experiments and a custom laser system was designed, constructed and integrated with the VideoRay Pro-4 platform (see Fig. 3.1 (a) and (b)).

The rest of the chapter is organized as follows: Section 3.2 provides the analytical model for a light ray that it is refracted in three different mediums before it reaches the camera sensor, Section 3.3 provides a novel LVS that considers the hemispherical optics suitable for measuring the relative posture for solid and mesh-like structures in underwater environments. In Section 3.4 a spatial filter for mesh-like targets identification is presented while Section 3.5 demonstrates the proposed system in controlled laboratory and real applications experiments. Finally, Section 3.6 concludes with a summary discussion, conclusions, and future work.

3.2 Three-medium refractive model, calibration and adaptation

3.2.1 Analytical Model

In this section we describe the camera calibration procedure for a system with hemispherical optics. The intrinsic camera matrix [47] is described below:

$$K = \begin{pmatrix} f_x & \alpha & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

where f_x and f_y are the focal length, c_x and c_y are the image center in pixels and α is a skew factor.

Assuming that the camera has been calibrated in air (single medium) to determine its intrinsic and extrinsic parameters (e.g. following [48]) with respect to the $< S' >$ coordinate system, we can substitute the camera optics in the air with the pinhole model. Hereafter when we are referring to the image plane it will understood as the image plane corresponding to the pinhole model after the camera has been calibrated.

Assume a light ray traveling through a certain route through three different mediums until it reaches the image plane. From Snell's law it is known that every time a light ray changes medium, it is refracted with a certain angle. Fig. 3.2 shows the propagation of a light ray from the point U that is located in the water until the point P that lies in the image plane. The red line represents the actual light ray path. The purple dashed lines represent the internal and external radii of the hemispherical dome. $< S' >$ indicates the origin of the dome coordinate system and the point $L' = [X_l', Y_l', Z_l']^T$ is the center of the lens of the camera. Point U' is the point viewed outside the dome, and corresponds to the point P' in the image plane, that has coordinates with respect to the $< S' >$ coordinate system as follows:

$$P' = L' + R \cdot \begin{bmatrix} u \\ -f \\ v \end{bmatrix} \tag{3.1}$$

where R is image-plane rotation matrix (camera extrinsic parameter), u and v are in the pixel positions in image-plane attached coordinate system, and f is the focal length of the camera. Let $< S >$ be the coordinate system resulting from rotating $< S' >$, according to R. Then

$$P = L + \begin{bmatrix} u \\ -f \\ v \end{bmatrix} \tag{3.2}$$

where $P = [X_c, Y_c, Z_c]^T$ and $L = [X_l, Y_l, Z_l]^T$ are the pixel location in the image-plane and lens position in the $< S >$ coordinate system. For the rest of the analysis in this subsection will assume that the reference coordinate system is $< S >$ unless otherwise stated.

Points A and $p_{\mathscr{D}}^S$ are the intersection points of the light ray with the medium's boundaries. The position vector of a point on the radius of the internal dome $\bar{r}$ with spherical coordinates (r, θ, ϕ) is given in Cartesian coordinates as:

$$x = r\cos\theta\cos\phi \tag{3.3a}$$

$$y = r\sin\theta\cos\phi \tag{3.3b}$$

$$z = r\sin\phi \tag{3.3c}$$

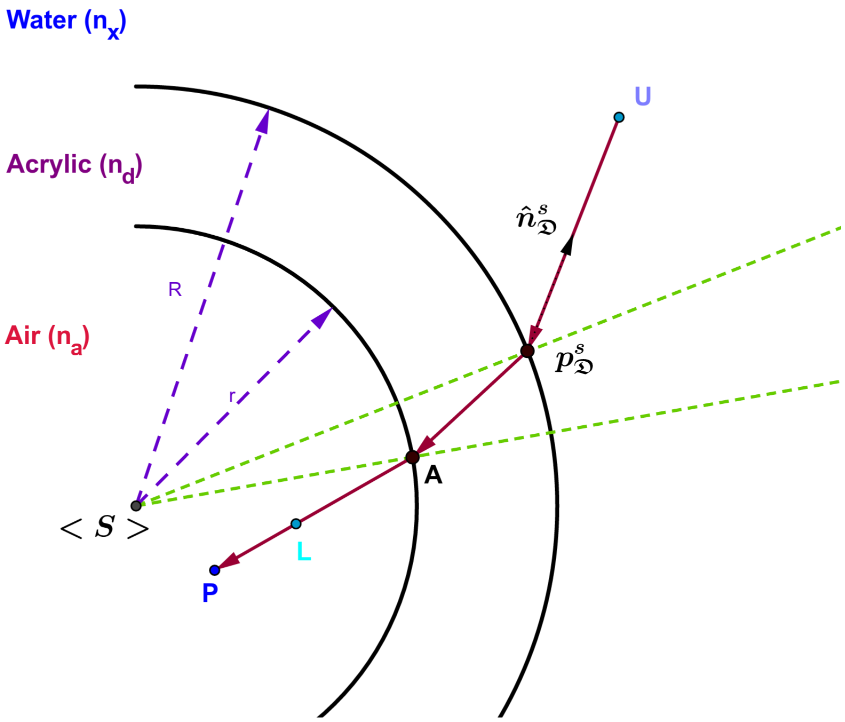


Figure 3.2: Geometric setup of the hemispherical refraction problem. The angles are exaggerated for demonstration purposes.

The position of point A can be expressed with the position vector $\bar{d_{i1}}(\lambda_1)$ as follows:

$$\bar{d_{i1}}(\lambda_1) = \bar{P} + \lambda_1(\bar{L} - \bar{P}) \qquad \lambda_1 \geq 0 \tag{3.4}$$

Using the Eqns. (3.2)-(3.4) we arrive at the following system:

$$X_c + u + \lambda_1(X_l - X_c - u) = r\cos\theta\cos\phi \tag{3.5a}$$

$$Y_l - f + \lambda_1 f = r\sin\theta\cos\phi \tag{3.5b}$$

$$Z_c + v + \lambda_1(Z_l - Z_c - v) = r\sin\phi \tag{3.5c}$$

The solution of the system gives the expression of the unknown λ_1. Substituting λ_1 we evaluate the angles ϕ and θ as shown below:

$$\phi = \sin^{-1}\left(\frac{Z_c + v + \lambda_1(Z_l - Z_c - v)}{r}\right) \tag{3.6}$$

$$\theta = \sin^{-1}\left(\frac{Y_l - f + \lambda_1 f}{r\cos\phi}\right) \tag{3.7}$$

The refraction is governed by the Snell's law to relate the light paths of incident light and refracted light with respect to the surface normal of the refractive plane:

$$\frac{\sin\delta_1}{\sin\delta_2} = \frac{u_1}{u_2} = \frac{n_d}{n_a}$$

Applying Snell's Law we determine the refractive angle:

$$\delta_2 = \sin^{-1}\left(\frac{n_a}{n_d}\right)\sin(\delta_1) \tag{3.8}$$

where

$$\delta_1 = \cos^{-1}\left(\frac{\bar{d_{i1}}\cdot\bar{r}}{|\bar{d_{i1}}||\bar{r}|}\right) \tag{3.9}$$

We introduce the perpendicular vector C_1 on the plane that is defined by the vectors $\bar{r}$ and $\bar{d_{i1}}$,

$$C_1 = \bar{r}\times\bar{d_{i1}} \tag{3.10}$$

where $(\times)$ denotes the vectors cross product. We assume that C_1 is also perpendicular on a unit vector $\hat{\rho_2}$ that lies in the refracted light ray. Therefore,

$$C_1\cdot\hat{\rho_2} = 0 \tag{3.11}$$

$$\bar{r}\cdot\hat{\rho_2} = cos\delta_2 \tag{3.12}$$

where $(\cdot)$ denotes the vectors dot product.

In order to complete the picture we need to enter the equation of the unit vectors,

$$||(\hat{\rho_2})|| = 1 \tag{3.13}$$

The solution of the system above determine the vector $\hat{\rho_2}$ which means that the only unknown to evaluate the second vector is λ_2, which is satisfied the equation below:

$$\bar{d_{i2}}(\lambda_2) = A + \lambda_2\hat{\rho_2} \qquad \lambda_2 \geq 0 \tag{3.14}$$

We deal this problem in a similar manner of the first point on the domain of the exterior hemisphere and we arrive at the equation that calculates the scalar function λ_2,

$$\lambda_2^2 + 2\lambda_2(A\cdot\hat{\rho_2}) + |A^2| - R^2 = 0 \tag{3.15}$$

Since we get the value of λ_2 we determine through the Snell's law the angles of refraction δ_3 and δ_4 and continue with the third point U outside of the dome. The equation below describes the light ray from point $p^S_{\mathscr{D}}$ to U:

$$\bar{d}_{i3}(\lambda_3) = p^S_{\mathscr{D}} + \lambda_3 \hat{\rho}_3 \qquad \lambda_3 \geq 0 \tag{3.16}$$

Note that the point $p^S_{\mathscr{D}}$ is calculated by the Eq. 3.14. Now we can provide a generic function for the hemispherical dome model of the form:

$$\mathscr{D}(u,v) \rightarrow (\hat{n}^{S'}_{\mathscr{D}}, p^{S'}_{\mathscr{D}}) \tag{3.17}$$

where (u,v) is the pixel coordinates on the image-plane, $n^{S'}_{\mathscr{D}}$ is a unit vector directed to the light source U' rooted at the point $p^{S'}_{\mathscr{D}}$ on the dome surface, all in the $< S' >$ coordinate system.

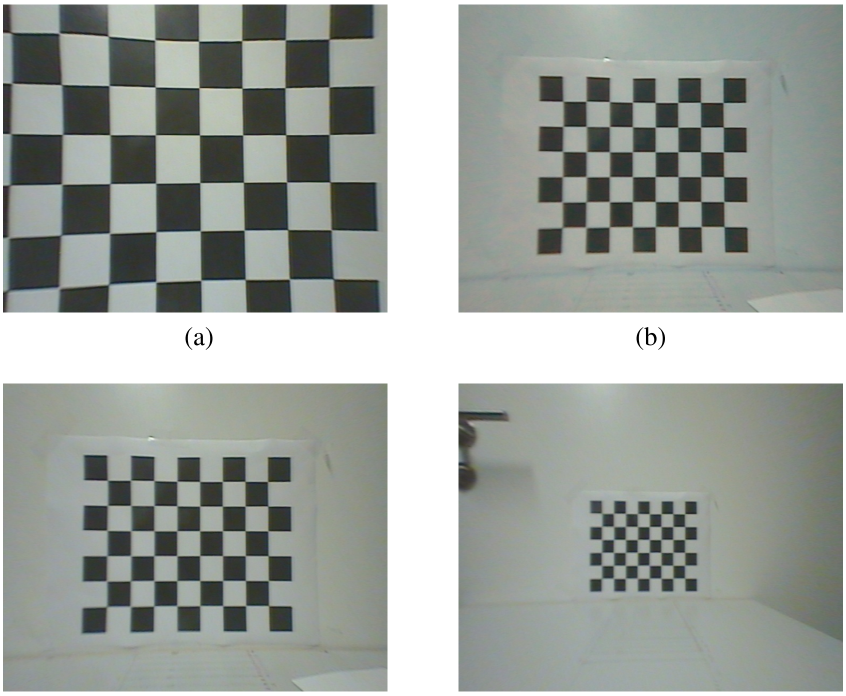
3.2.2 Model Calibration and Adaptive Refractive Index

The accuracy of the model, depends heavily on the accuracy of the various dome parameters used. For this reason a procedure for calibration of the analytical dome model was devised and is described below. The procedure consists of two steps. In the first step a series of images of chessboards were taken from different, known locations with respect to $< S' >$ as shown in Fig. 3.3.

The images were analyzed using ROS and OpenCV, and a file was produced containing a series of image-plane points matched with their corresponding $(x,\ y,\ z)$ coordinates in 3D space. This file was used for the second step of the dome calibration procedure. For the second step, a function of the dome projecting each image-plane point on the corresponding plane of known distance from the dome was developed. The function contains best known values for all dome parameters including image-plane position and orientation, lens position with respect to image-plane, dome size and refractive index, and air refractive index as it is described in previous Section 3.2.1. In order to determine each point U and perform the camera calibration, each ccd point is traced towards a plane of known position from the origin of the dome $< S' >$. The plane (chessboard placement) is perpendicularly placed along the y-axis of the ROV as shown in Fig. 3.1(a) .

The chessboard plane $\mathscr{P}_{\mathbf{cb}}$ in the $< S' >$ coordinate system is provided by:

$$r_{b,i} \hat{n}^{S'}_{b,i} = d^{S'}_{b} \tag{3.18}$$



(a) (b) (c) (d)

Figure 3.3: Chess board patterns at know locations for the calibration of the dome model.

where $r_{b,i}$ is any point living on $\mathscr{P}_{\mathbf{cb}}$ and $d_b^{S'}$, $\hat{n}_b^{S'}$ can be determined since $\mathscr{P}_{\mathbf{cb}}$ is known. The combination of Eq. 3.16 and Eq. 3.18 eliminates the λ_3 parameter and gives the expression below:

$$p_{b,i}^{S'} = p_{\mathscr{D},i}^{S'} + \left(\frac{d_b^{S'} - p_{\mathscr{D},i}^{S'} \hat{n}_b^{S'}}{\hat{n}_{\mathscr{D},i}^{S'} \hat{n}_b^{S'}} \right) \hat{n}_{\mathscr{D},i}^{S'} \tag{3.19}$$

where $p_{b,i}^{S'}$ is the intersection point between the line d_{i3} and the $\mathscr{P}_{\mathbf{cb}}$. As the chessboard and its squares are of known dimensions and placement, the actual point p_{act} is of known $(\,x,\ y,\ z)$ coordinate.

An error function was developed to represent the difference between the actual and the estimated 3D points $(p_{act} - p_{b,i})$. The Levenberg-Marquardt algorithm [75] was used to minimize the error in the function parameter values, thus fine tunning the parameters, getting the minimum error values based on the input points. In order to determine the refractive index of the medium during a normal operation of the ROV, a target is fixed at a known location in view of the image-plane (fixed on ROV) as shown in Fig. 3.4. A similar algorithm to the one used for dome calibration above was developed with the only variable parameter being the refractive index of the medium (outside the dome). A background code runs the algorithm while the ROV is operated, indicating the estimated refractive index on screen in real time.

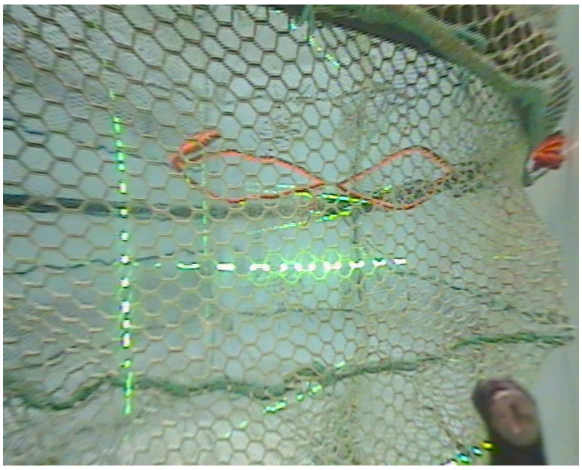


Figure 3.4: The fixed target for refractive index calculation.

3.3 Laser Vision System (LVS)

3.3.1 Approach

In this section we describe the LVS as it shows in Fig. 3.1. The lasers housed by aluminium cases, were custom designed and built by the authors for the needs of the underwater operations. Each laser is mounted in a different direction for covering a wide range of applications such as the seabed, the sea surface, obstacles avoidance, object geometry recovery and range measurements of particular targets. The OpenCV library [13] was used for the laser light detection by the camera sensor. More particular, we use a threshold filter on the rectified image as shown in Fig. 3.11(b). Since we have three lasers, we have to classify which points belong to which laser. For this reason we partition the thresholded image into three partitions and apply a mask to throw away pixels that belong to intersecting laser images in the same region.

The procedure for recovering the distance measurements form the laser images is as follows:

Let $\mathbf{p}_{\mathscr{P}_\mathrm{i}}$ be the i'th image-plane pixel corresponding a laser reflection on the target. Then from the dome model (Eq. 3.17) we have:

$$\mathscr{D}(\mathbf{p}_{\mathscr{P}_i}) \rightarrow (\hat{n}^{S'}_{\mathscr{D}i}, p^{S'}_{\mathscr{D}i}) \tag{3.20}$$

Hence the line L_i connecting point $p^{S'}_{\mathscr{D}i}$ with the reflection target U is provided by:

$$L_i(\lambda) = p^{S'}_{\mathscr{D}i} + \lambda \hat{n}^{S'}_{\mathscr{D}i}, \quad \lambda \geq 0 \tag{3.21}$$

The laser plane for laser ℓ in the $< S' >$ coordinate system is provided by:

$$r_\ell \hat{n}^{S'}_\ell = d^{S'}_\ell \tag{3.22}$$

where r_ℓ is any point living on the laser ℓ plane and parameters $\hat{n}_\ell^{S'}$ and $d_\ell^{S'}$ are the parameters of each laser determined via the calibration procedure. Then the target position U in $< S' >$, denoted by $p_{w,i}^{S'}$ is determined by taking the intersection of line L_i with the laser ℓ plane by eliminating λ as:

$$p_{w,i}^{S'} = p_{\mathscr{D},i}^{S'} + \left(\frac{d_\ell^{S'} - p_{\mathscr{D},i}^{S'} \hat{n}_\ell^{S'}}{\hat{n}_{\mathscr{D},i}^{S'} \hat{n}_\ell^{S'}} \right) \hat{n}_{\mathscr{D},i}^{S'} \tag{3.23}$$

where $d_\ell^{S'}$, $n_\ell^{S'}$ are laser calibration parameters for each laser ℓ. Eq. (3.17) combined with eq. (3.23) provide a target relative localization function of the form:

$$\mathscr{T}_\ell(p_x, p_y) \to p_w^{S'} \tag{3.24}$$

Table 3.1: Laser Specifications

Wavelength	Power	Supply	Dimensions
532nm	< 20 mW,cw	3 VDC	Ø(12 x 33.5 x 51.5) mm

3.3.2 Relative 3-D Posture Estimation to Mesh-Like Targets

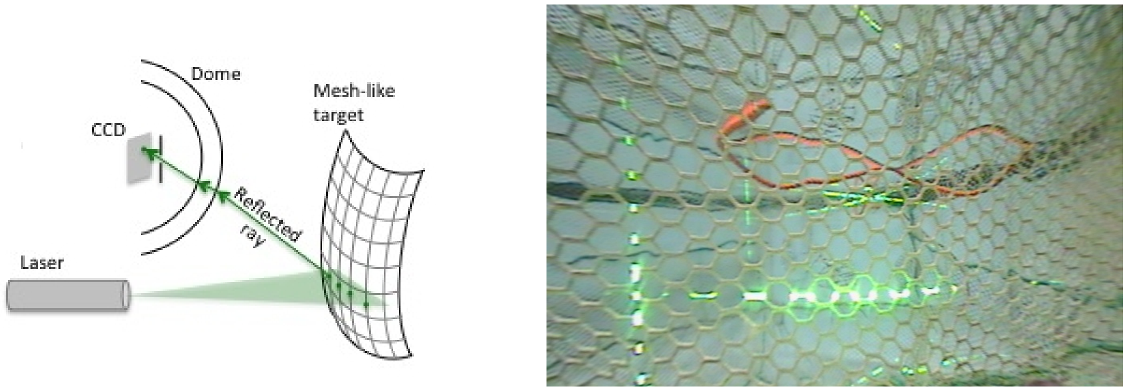
In order to estimate the relative posture to a mesh-like target as shown in Fig. 3.5 the adopted approach is to fit the LVS measurements to a 3-D surface and then deduce the relative posture to the surface. In the current work we will fit the LVS measurements from at least two line-lasers' target reflections to a 3-D plane.

Assume laser ℓ induces n_ℓ of image-plane pixels to light up by its reflection on the target. Denote by $\mathbf{P}^\ell$ the $n_\ell \times 2$ matrix containing the pixel coordinates of the n_ℓ pixels. Then from eq. (3.24) and with a slight abuse of notation we have that $\mathscr{T}_\ell(\mathbf{P}^\ell) = \mathbf{P}_{w,\ell}^{S'}$ are the 3-D positions of the laser reflections in $< S' >$. Given k non co-planar line lasers $\ell_1, \ldots \ell_k$ we can produce the augmented matrix:

$$\mathbf{P}_w^{S'} = \begin{bmatrix} \mathbf{P}_{w,\ell_1}^{S'} \\ \vdots \\ \mathbf{P}_{w,\ell_k}^{S'} \end{bmatrix} \tag{3.25}$$

Since the target plane cannot physically pass from the origin, the plane equation in the $< S' >$ frame can be written as:

$$\bar{r}^{S'} \cdot \frac{\hat{n}_p^{S'}}{d^{S'}} = 1 \tag{3.26}$$



(a) Mesh-like target.

(b) Fishnet Structure.

(c) Aquaculture Fishnet.

Figure 3.5: Using a line-laser based LVS to determine the relative posture to a mesh-like target.

where $\bar{r}^{S'}$ is any vector connecting $< S' >$ with a point on the target. Then

$$\frac{\hat{n}_p^{S'}}{d^{S'}} = (\mathbf{P}_w^{S'})^{\dagger} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \tag{3.27}$$

where $(\mathbf{P}_w^{S'})^{\dagger}$ is the left pseudo-inverse of $\mathbf{P}_w^{S'}$ that is guaranteed to be non-singular as long as our lasers are non co-planar and have non co-linear target projections. From this we can immediately extract the relative posture to the mesh like target that comprise the plane's normal $\hat{n}_p^{S'}$ and the distance to the plane $d^{S'} > 0$, by noting that $\left\|\hat{n}_p^{S'}\right\| = 1$. We have to note here that due to the planar symmetry assumption, only relative pitch and relative yaw can be determined i.e. rotation of the ROV along the plane's normal cannot be evaluated. This can be achieved only in the case of asymmetric (non-planar and non-spherical) targets.

However, this is beyond the scope of the current work.

Now, using the unit direction vectors $\hat{x}_{S'}, \hat{y}_{S'}, \hat{z}_{S'}$ of the coordinate system $< S' >$, the relative pitch can be extracted by projecting the plane's normal vector along the $y-z$ plane, as:

$$\theta_{S'} = \frac{\pi}{2} - acos\left[\frac{(\hat{n}_p^{S'})^T \cdot (I - \hat{x}_{S'} \cdot \hat{x}_{S'}^T) \cdot \hat{z}_{S'}}{\|(\hat{n}_p^{S'})^T \cdot (I - \hat{x}_{S'} \cdot \hat{x}_{S'}^T)\|}\right] \tag{3.28}$$

and the relative yaw can be extracted by projecting the plane's normal vector along the $y-x$ plane, as:

$$\psi_{S'} = \frac{\pi}{2} - acos\left[\frac{(\hat{n}_p^{S'})^T \cdot (I - \hat{z}_{S'} \cdot \hat{z}_{S'}^T) \cdot \hat{x}_{S'}}{\|(\hat{n}_p^{S'})^T \cdot (I - \hat{z}_{S'} \cdot \hat{z}_{S'}^T)\|}\right] \tag{3.29}$$

3.3.3 Automatic Calibration

Underwater vehicles are often exposed in harsh conditions i.e sea water (salinity), sun and overworked by pressure. To this end, system and components require regular maintenance after operations such as cleaning and sealing tests. Maintenance procedures require regular disassembly of the LVS as well, which in turn requires re-calibration to ensure accurate readings. In this subsection we present an automatic calibration procedure developed for the LVS. Fig. 3.6 shows the developed calibration box which is used for the LVS re-calibration procedure.

A coordinate system $< G >$ is defined for the box. The four box planes ((L)eft, (R)ight, (F)orward, (D)own) are defined as:

$$r_b \cdot \hat{n}_b^G = d_b, \quad b \in \{L, R, F, D\} \tag{3.30}$$

The ROV is fixated in an initially unknown position S^G and with an initially unknown orientation R_S^G in the calibration box. A set of n_p patterns are applied at known positions $\mathscr{P}_i^G$, $i \in \{1, \ldots n_p\}$ in the box. Let $\mathbf{p}_{\mathscr{P}_\mathrm{i}}$ be the pixel corresponding to pattern i. Then from the dome model we have:

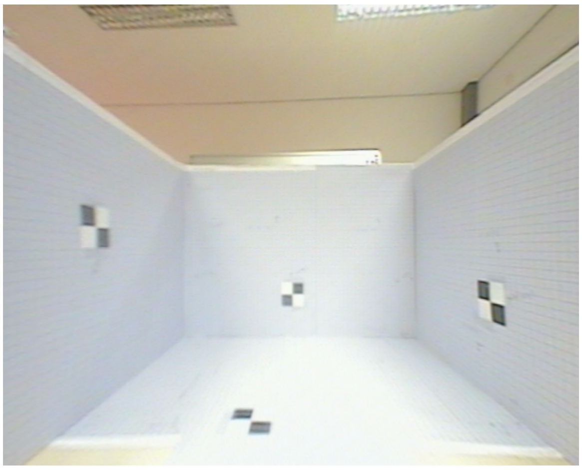
$$\mathscr{D}(\mathbf{p}_{\mathscr{P}_i}) \rightarrow (\hat{n}_{\mathscr{D}i}^{S'}, p_{\mathscr{D}i}^{S'}), \quad i \in \{1, \ldots n_p\} \tag{3.31}$$

Hence the line L_i connecting point $p_{\mathscr{D}i}^{S'}$ with $\mathscr{P}_i$ is provided by:

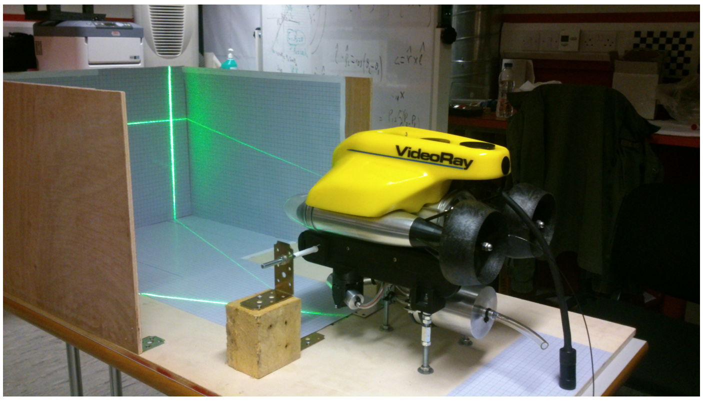
$$L_i(\lambda) = p_{\mathscr{D}i}^{S'} + \lambda_i \hat{n}_{\mathscr{D}i}^{S'}, \quad \lambda \geq 0 \tag{3.32}$$

The laser plane for laser ℓ in the $< S' >$ coordinate system is provided by:

$$r_\ell \hat{n}_\ell^{S'} = d_\ell^{S'} \tag{3.33}$$



(a) Laser's Calibration Box as seen by the ROV camera.



(b) ROV with Laser Vision System.

Figure 3.6: Calibration box for the LVS.

The automatic laser calibration then consists of the procedure to determine the parameters $\hat{n}_\ell^{S'}$ and $d_\ell^{S'}$ of each laser ℓ. To achieve this however first we need to determine S^G and R_S^G. For each pattern we have that:

$$\mathscr{P}_i^G = S^G + R_S^G(p_{\mathscr{D}_i}^{S'} + \lambda_i \hat{n}_{\mathscr{D}_i}^{S'}), \quad i \in \{1, \ldots n_p\} \tag{3.34}$$

Since the rotation matrix R_S^G is a function of three Euler angles, the above equations form a system of $3n_p$ equations with $n_p + 6$ unknowns. This implies that we need at least 3 patterns to solve this problem. However additional patterns will increase the accuracy of the calibration, also removing ambiguity issues. Hence we form a potential function as:

$$V(S^G, R_S^G, \lambda_1, \ldots \lambda_{n_p}) = \sum_{i=1}^{n_p} \left\| -\mathscr{P}_i^G + S^G + R_S^G(p_{\mathscr{D}_i}^{S'} + \lambda_i \hat{n}_{\mathscr{D}_i}^{S'}) \right\|^2 \tag{3.35}$$

Hence we can get $S^G, R_S^G, \lambda_1, \ldots \lambda_{n_p}$ as the solution of the non-linear minimization problem:

$$(S^G, R_S^G, \lambda_1, \dots \lambda_{n_p}) = \arg\min \left\{ V(S^G, R_S^G, \lambda_1, \dots \lambda_{n_p}) \right\} \tag{3.36}$$

The next step requires the projection of each laser on at least two box planes. From S^G, R_S^G that were calculated in the previous step, and using the plane equations, for each reflected target point j_b from plane b, we have:

$$(S^G + R_S^G (p_{\mathscr{D}_{j_b}}^{S'} + \lambda_{j_b} \hat{n}_{\mathscr{D}_{j_b}}^{S'}) \cdot \hat{n}_b^G = d_b \tag{3.37}$$

However, since by construction $d_\ell \neq 0$, λ_{j_b} should also satisfy:

$$(p_{\mathscr{D}_{j_b}}^{S'} + \lambda_{j_b} \hat{n}_{\mathscr{D}_{j_b}}^{S'}) \cdot \frac{\hat{n}_\ell^{S'}}{d_\ell^{S'}} = 1 \tag{3.38}$$

Solve eq. (3.37) for λ_{j_b} and substitute all λ_{j_b}'s in eq. (3.38). Now write each result as a row of the matrix Y. Then:

$$\frac{\hat{n}_\ell^{S'}}{d_\ell^{S'}} = Y^\dagger \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \tag{3.39}$$

where $Y^\dagger$ is the left pseudo-inverse of Y. From this we can immediately extract the laser calibration parameters ($\hat{n}_\ell^{S'}$ and $d_\ell^{S'}$) by noting that $\left\| \hat{n}_\ell^{S'} \right\| = 1$.

3.4 A Filter for Mesh-Like Structures

3.4.1 Preliminaries

In this section a spatial filter is developed to discriminate mesh-like structures from other artifacts in the LVS measurements. This is useful in the case that the relative posture is sought with respect to mesh-like structures like fish-nets Fig. 3.5, where the LVS should be able to discern the difference between fish-net, fish and air bubble reflections. The aim of the proposed filter is to only allow reflections from the mesh-like spatial structure that is being observed as target. Fig. 3.11 shows the laser lines' reflections from the target (green dots) and the water tank (solid green lines). Also, note the additional (faint) reflections from the water tank surface to the water tank wall. Spatial filtering using Fourier Transformation has been reported for the identification of fabric structures in images in [104]. In our approach, we exploit the point cloud produced by the LVS in the $< S' >$ coordinate system. We assume that multi-path reflections (e.g. light beams that reflect on the water surface and then hit a target) either do not exist, or that they are weaker than single path reflections. In our lab and sea experiments we have seen that this is a valid assumption. Now observing that all LVS measurements (excluding multi-path reflections) are co-planar, i.e. they reside in the laser plane, we create a binary image containing the laser-plane reflections which is co-planar with the laser-plane. We can thus develop an appropriate filter to discern between reflections that belong to the mesh-like structure and ones that do not.

3.4.2 Approach

Mesh-like structure have the characteristic of being periodic in 2-dimensions. Intersecting such structures with a straight line yields intersection points that appear with predetermined regularity. Depending on the mesh geometry a maximum and a minimum distance can be derived for the intersection points as we traverse the intersecting line. Thus, the distance between consecutive intersections will lie between a maximum and a minimum range. This observation is applicable for almost all positions of the intersecting line. Singular[2] positions occur only when the intersecting line is collinear with edges on the target. Assuming a laser plane hitting such a target, we can define a minimum and a maximum distance between consecutive reflections from the target in the 3D space, irrespective of the target's orientation. In the development of the filter we will ignore reflections from target features that are collinear with the laser line, as such features are considered to be either singularities or non mesh-like targets. For example, in Fig. 3.7(a) the maximum distance between consecutive reflections is the diameter of the hexagon and the minimum distance is the side of the hexagon. In Fig. 3.7(b) the maximum distance is the diagonal of the rectangle and the minimum distance is half the diagonal. Hence every reflection from such targets will always fall between the minimum and the maximum period with one of its neighboring reflections.

Laser plane image

The first step is to create the binary image of the LVS reflections in the laser plane. To do this we rotate the point cloud $p_w^{S'}$ that lives in the laser plane, to the $x-y$ plane. First let us fix an ortho-canonical coordinate system $<\ell>$ on the laser plane, with direction unit vectors expressed in the $<S'>$ frame as $e_{x,\ell}, e_{y,\ell}, e_{z,\ell}$:

Let

$$e_{z,\ell} = \hat{n}_\ell^{S'}$$

Then

$$e_{y,\ell} = \frac{e_{y,S'}\left(I - e_{z,\ell}e_{z,\ell}^T\right)}{\left\|e_{y,S'}\left(I - e_{z,\ell}e_{z,\ell}^T\right)\right\|} \tag{3.40}$$

---

[2] i.e. in the sense that such an occurrence has zero probability when randomly placing the intersecting line

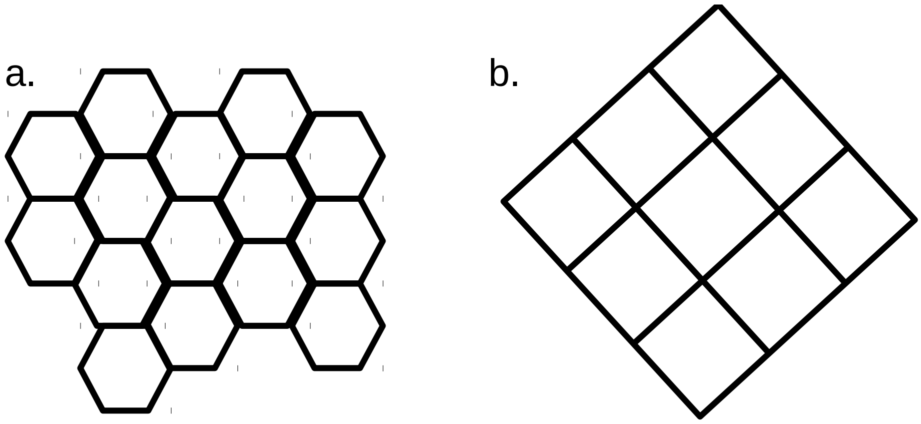


Figure 3.7: Mesh-like geometries: a. hexagonal mesh and b. rectangular mesh.

where $e_{y,S'}$ is the y-direction in the $< S' >$ frame (forward laser looking direction) and,

$$e_{y,\ell} = e_{z,\ell} \times e_{x,\ell} \tag{3.41}$$

Now create the rotation matrix from frame $< \ell >$ to frame $< S' >$ as:

$$R_{\ell}^{S'} = \begin{bmatrix} e_{x,\ell} & \vdots & e_{y,\ell} & \vdots & e_{z,\ell} \end{bmatrix}^T \tag{3.42}$$

Then the 3-D positions of the laser reflections $\mathbf{P}_w^{S'}$ can be translated and rotated to fall in the $x-y$ plane of $< S' >$ as follows:

$$p_{w,i}^{S',S'} = R_{\ell}^{S'}(p_{w,i}^{S'} - e_{z,\ell} \cdot d_{\ell}^{S'}) \tag{3.43}$$

Notice that the z-component of $p_{w,i}^{S',S'}$ is zero, and $\mathbf{P}_w^{S',S'}$ is a 2D point-cloud. Hence $\mathbf{P}_w^{S',S'}$ can be represented with a binary image.

Define $q_i^x \triangleq p_{w,i}^{S',S'} \cdot [1\ 0\ 0]^T$ and $q_i^y \triangleq p_{w,i}^{S',S'} \cdot [0\ 1\ 0]^T$. Let $x_L = \min_i q_i^x, x_R = \max_i q_i^x$, $y_U = \max_i q_i^y$, $y_D = \min_i q_i^y$ represent the four boundaries of the laser plane. Let dx, dy represent the spatial discretization of the binary image (this depends primarily on the characteristics of the target- i.e. thread diameter, assuming the LVS has adequate resolution).

Assume a grid with discretization dx, dy with $n_x = \lceil \frac{x_R - x_L}{dx} \rceil$ columns and $n_y = \lceil \frac{y_U - y_D}{dy} \rceil$ rows. Create an array of index sets S_{idx} as follows:

$$S_{idx}(m,n) = \left\{ i \;\middle|\; m \cdot dx \leq q_i^x - x_L < m \cdot dx + 1 \;\wedge\; n \cdot dy \leq q_i^y - y_L < n \cdot dy + 1 \right\} \tag{3.44}$$

Now define the binary image as follows:

$$Im(m,n) = \begin{cases} 1 & |S_{idx}(m,n)| > 0 \\ 0 & |S_{idx}(m,n)| = 0 \end{cases} \tag{3.45}$$

where $|\cdot|$ denotes the set cardinality.

Binary Image Processing Filter

The aim of the proposed filter is to identify points that belong to the mesh-like structure. Let d_θ be the thread diameter of the mesh-like structure. Let r_{min} be the minimum distance between consecutive threads (center-to-center) as these are intersected in the laser plane and r_{max} the corresponding maximum distance. Then a reflection from a target thread will have a size of $rd_\theta = \frac{d_\theta}{cos\theta_{lt}}$ where θ_{lt} the angle between the laser plane and the target thread. The choice of θ_{lt} should be such that $rd_\theta < r_{max}$ otherwise the target periodicity assumption will not be valid. In practice we want to have $rd_\theta << r_{min}$ which is achievable for typical mesh-like targets assuming appropriate configuration of the laser-plane roll angle. We can now start constructing the structuring elements of our filter (see Fig. 3.8) by defining three regions via three concentric circles: Region **A** with diameter rd_θ, region **B** with radius $r_{\min} - \frac{rd_\theta}{2}$ that excludes region **A**, and region **C** with radius $r_{\max} + \frac{rd_\theta}{2}$ that excludes regions **B** and **A**. Regions **A**, **B** and **C** define three binary valued structuring elements Str_A, Str_B and Str_C, with the same discretization as the binary image, that are 1 in their respective regions and 0 outside.

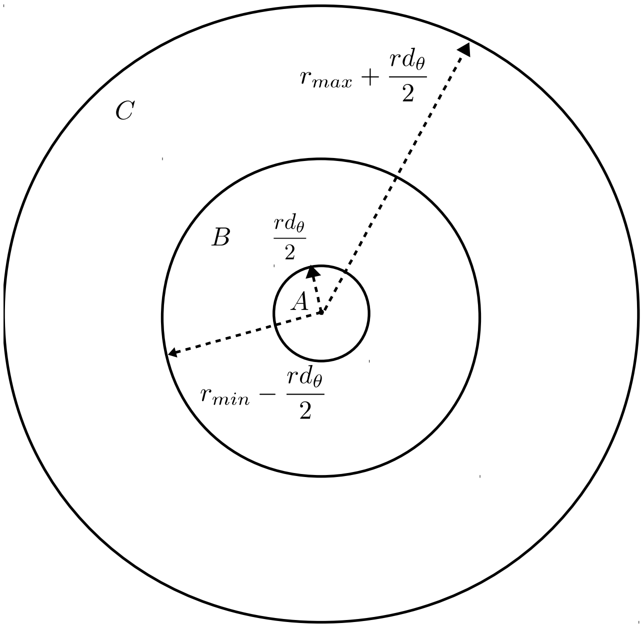


Figure 3.8: Structuring element regions.

We can now start defining our filtering algorithm. We pad the binary image Im with $\rho_{pad} = \lceil r_{\max} + \frac{rd_\theta}{2} \rceil$ cells to get the image Im'. We will now establish the limits required to determine the morphological operations.

Algorithm 1 Mesh Filter Algorithm

Require: Im', Str_A, Str_B, Str_C
Ensure: Mesh Reflections from mesh-like structure

```
loop $(m,n) \in \left\{1 \ldots n_x \times 1 \ldots n_y\right\}$
    $(m',n') \leftarrow (m+\rho_{pad}, n+\rho_{pad})$
    if $\lceil \frac{dg_\theta}{2} \rceil \leq |Im' \cap Str_A(m',n')| \leq dg_\theta$ then
        if $0 \leq |Im' \cap Str_B(m',n')| \leq dg_\theta - |Im' \cap Str_A(m',n')|$ then
            if $\lceil \frac{3dg_\theta}{2} \rceil \leq |Im' \cap Str_C(m',n')| \leq 2dg_\theta$ then
                $Im''(m,n) \leftarrow 1$
            end if
        end if
    end if
end loop
$k \leftarrow 1$
loop $(m,n) \in \left\{1 \ldots n_x \times 1 \ldots n_y\right\}$
    if $Im''(m,n)$ then
        $Idx \leftarrow S_{idx}(m,n)$,
        $L \leftarrow |Idx|$
        loop $j \in \{1 \ldots L\}$
            $\mathbf{P}^{S^*}_{w,k} \leftarrow p^{S'}_{w,j}$
            $k \leftarrow k+1$
        end loop
    end if
end loop
return $\mathbf{P}^{S^*}_{w}$
```

Assume now that we have a laser reflection from the center of a thread of a mesh like target centered at the m,n pixel of Im' and we place Str_A at that location, i.e. $Str_A(m,n)$. Considering the grid discretization, and assuming $dx = dy = dd$ the thread reflection will encompass at most $dg_\theta = \lceil \frac{rd_\theta}{dd} \rceil + 1$ grid cells and this is the maximum number of cells we expect to be active in $Im' \cap Str_A(m,n)$. Then $Im' \cap Str_B(m,n)$ will be empty and $Im' \cap Str_C(m,n)$ will contain at most $2dg_\theta$ cells.

Assume now that we have a laser reflection from the edge of a thread of a mesh like target centered at the m,n pixel of Im' and we place Str_A at that location. Then only from $\lceil \frac{dg_\theta}{2} \rceil$ up to dg_θ cells will be active in $Im' \cap Str_A(m,n)$. In $Im' \cap Str_B(m,n)$ we can have at most $\lceil \frac{dg_\theta}{2} \rceil$ cells active, and in $Im' \cap Str_C(m,n)$ will contain at least $\lceil \frac{3dg_\theta}{2} \rceil + 1$ cells active.

Almost all reflections (excluding singularities) from the mesh-like target will fall within the ranges provided above. A new binary image Im'' can now be created that only contains pixels that belong to the mesh-like structure. The set of actual points that correspond to the mesh-like structure can be recovered by feeding into $S_{idx}(m,n)$ the pixels of Im'' that are active. The filtering algorithm provided by Algorithm 1, returns the 3-D positions of the laser reflections only from the mesh-like structure, $\mathbf{P}_w^{S^*}$.

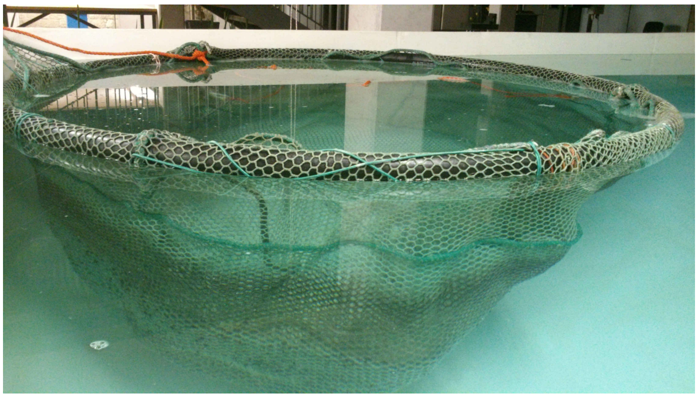
3.5 Experiments

3.5.1 Experimental Setup

In this section, we describe the experimental setup for the system and the experimental results obtained during performance evaluation of the system. The validity of the proposed system is verified both in a laboratory-controlled environment setting Fig. 3.9, as well as in real-condition at an offshore aquaculture installation Fig. 3.10.

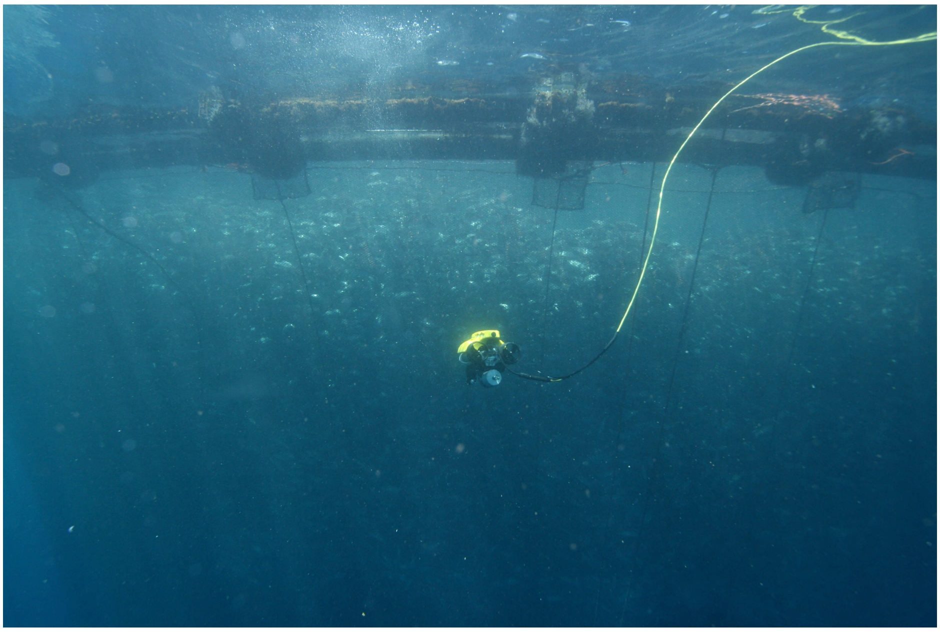
A Videoray Pro IV (ROV) was used for the experiments. The robot is equipped with a CCD camera, a Tilt Compensated Compass (TCC) sensor capable of providing roll, pitch and yaw measurements, the LVS that was analyzed in this work and a custom developed control station. Furthermore, the robot was retrofitted with an inertial measurement unit (IMU) providing it's acceleration and angular rates. The ROV camera resolution is 720 x 576p and it is an 160° wide angle camera with a frame-rate of 25 fps. The system software was developed in C/C++, using the Robotic Operating System (ROS) [91] and OpenCV [13] on a 12.04 Ubuntu Linux OS. The PC that was used for the experiments was an Intel i7, dual-core laptop, with 8GB RAM memory. The ROV is powered by a boat-mounted generator through a tether whereas the laser system is powered by a dedicated battery.

(a) Laboratory water tank.

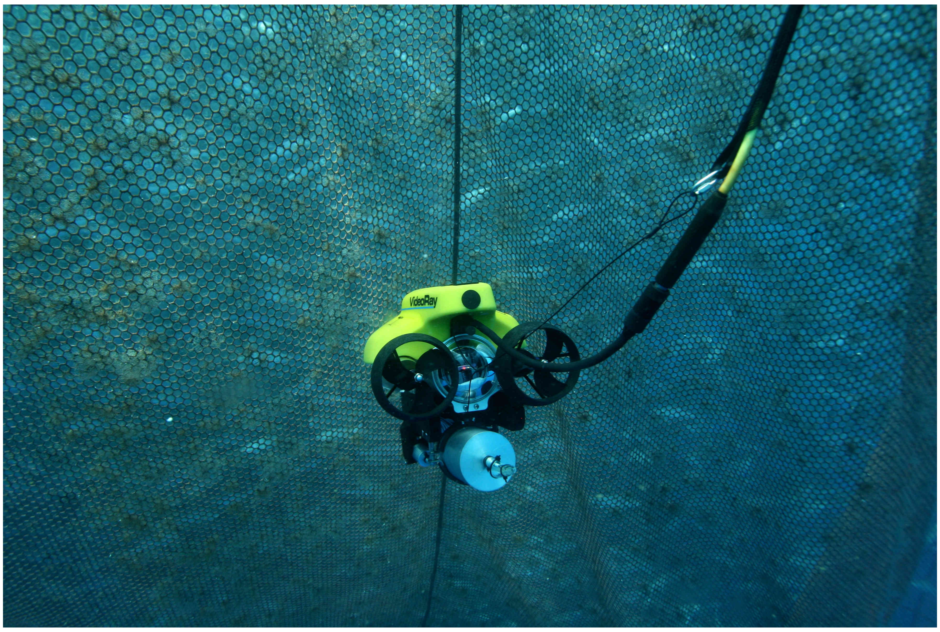


(b) Mock-up environment.

Figure 3.9: Laboratory mock-up experiment.



(a) The ROV operates an inspection task at aquaculture installation.

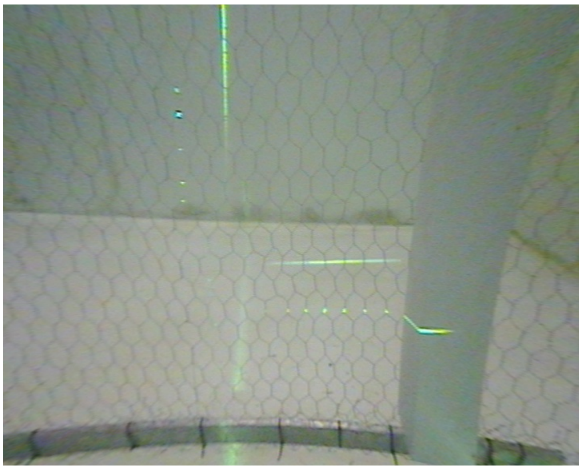


(b) Aquaculture inspection operation by the proposed system.

Figure 3.10: Offshore aquaculture installations experiment.

3.5.2 Experimental Evaluation of the Mesh Filter Algorithm

In this section the experimental validation of the methodology that is proposed in Section 3.4 is presented. In Fig. 3.11(a) a mesh like target as seen from the ROV camera is depicted, while Fig. 3.11(b) depicts the corresponding thresholded image. Using the hemispherical dome model 3.17, the laser point clouds are extracted. The point cloud corresponding to the horizontal laser is depicted in Fig. 3.12.



(a) Mesh-like tarket as seen by the ROV camera.

(b) Thresholded Image.

Figure 3.11: Mesh-like target.

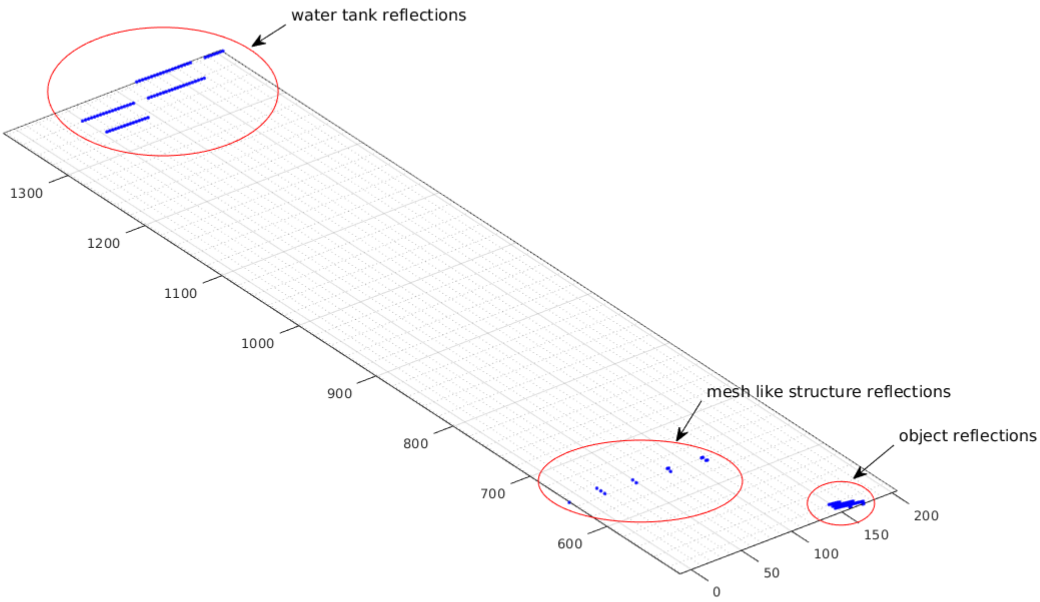


Figure 3.12: Laser point-cloud data (only for the horizontal laser).

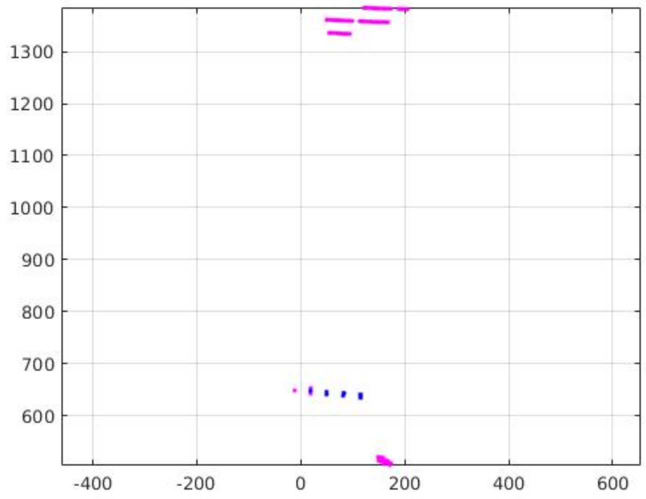
As can be seen in Fig. 3.12, the laser point-cloud contains reflections from the mesh-like structure, the water tank floor and walls and the solid (non-mesh) object. From Fig. 3.12, we can distinguish the different reflections from the the laser point cloud. Considering only the horizontal laser line for this experiment, the reflections from the water tank floor can be easily recognized and eliminated because they are beyond the limits of the operational range zone. Note that the operational range zone is between 100mm - 1000mm since beyond this range the accuracy provided by the camera (pixels/cm^2 on the target) and, more importantly, the laser beam scattering, cause the algorithm to fail, since the laser (thin) plane assumption is no longer valid. The operational range can be improved using better laser optics for the same medium and camera analysis. Hence, reflections that appear beyond these limits, are not taken into account in the mesh filter algorithm. In the operational range the algorithm should be able to detect the mesh like structure and eliminate other artifacts.

In order to identify which of the laser points belong to the mesh-like target, the Mesh Filter algorithm (Algorithm 1) was used. For the mesh-like structure we used we had the following parameters:

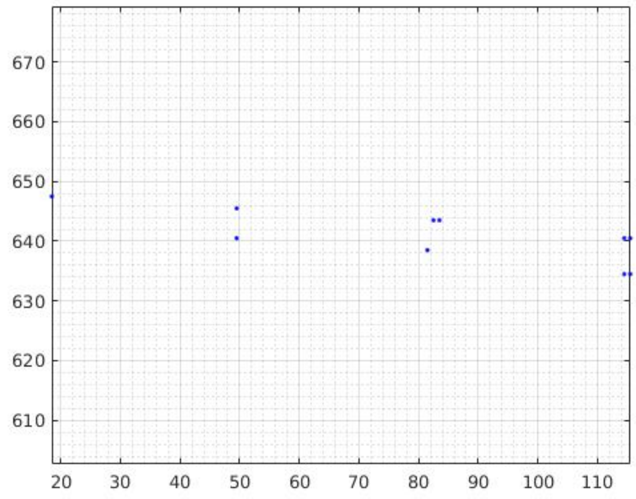
Fig. 3.13 depicts the result after applying the Mesh Filter algorithm. As can be seen the algorithm has rejected all reflections that did not belong to the mesh-like structure.

Table 3.2: Parameters

d_θ	rd_θ	dd	r_{min}	r_{max}
2mm	10mm	1mm	15mm	35mm



(a) The pink points denote all the reflections while the blue points indicate the mesh like structure points after the algorithm has been applied.



(b) Mesh-like Structure Points. Notice the ±2.5mm separation of the same point target reflection, caused due to the scattering of the laser beam e.g. at (50mm,643mm).

Figure 3.13: Mesh filter algorithm's results.

3.5.3 Adaptive Calibration Experiments

In this section we will address the issue of on-line adaptation of the robot vision system to variations in the refractive index of the ambient fluid. Such variations may indicate different cases like leaks from oil pipes, fish excretions at the vicinity of aquacultures or the salt-fresh water interface. Depending on the task, the AUV vision system has to be able to adapt its operation to correctly interpret the received image stream primarily for navigation purposes. An on-line adaptive algorithm was developed to dynamically adjust the vision system of the ROV to changes in the refractive index of the fluid. This algorithm utilizes the calibration model described earlier and is based on an iterative procedure. In this approach we assumed that the ROV and the environment (aquaculture) are static and the only refraction index of the third medium (water) has been changed. Since the ROV is observing the same underwater features all the time, the image on the CCD plane should remain the same as well. However, the variations of the fluid refraction index may lead to changes in the CCD image as well. Considering the identity of the images as a basis, we can estimate the new refraction index of the ambient fluid. The developed adaptive calibration algorithm makes use of a discrepancy of the coordinates of the point P as a residual function required to be vanished:

$$f = P^{old} - P^{new}, \tag{3.46}$$

where P^{old} is the position of the point P observed earlier for the known refraction index n_3^{old}, P^{new} is the new guess position of the point P. The input to the algorithm is two images of the same underwater scene but taken in the fluids with different refraction indices. The refractive index of one fluid is known and, as a consequence, the coordinates of the points P and S are known as well. The coordinates of the point P may serve as an initial guess $P^{new}\left(n_3^{new}\right)$ for the algorithm.

The resulting problem is non-linear with three unknowns and may be solved using an iterative numerical method. We applied the Levenberg-Marquardt [75] algorithm as one of the common schemes for finding solutions of non-linear systems. However, the convergence issues should be taken into account since the problem is very sensitive to the choice of the image features. Usually, the variations of water refraction index of the water are small but lead to significant distortion of the image. When only one point (pixel) of the environment image is taken as an input to the algorithm, the issues of inaccuracy of the estimation or divergence of the algorithm may arise. In order to increase the accuracy, several pixels that form a sub-pixel may be used as initial conditions. However, in this case, the complexity of the problem will increase as well.

As one of our future work directions, we consider to use a subpixel accuracy that enable to significantly increase the precision of the calibration process. This algorithm is generally based on different interpolation techniques for interpolation intensity of the image or an error function of the image matching. The sub-pixel accuracy algorithm may also be useful for the tracking problem as well. Although we assume the static problem for this case, in the real-world conditions the small relative movements between the ROV and the aquaculture structure can take place as well.

3.5.3.1 Simulation Results

The developed model was fitted with the known parameters of the dome and camera. The model was set to calculate the displacement in pixels of a target positioned on a projection plane 82mm from the dome base line, 25mm off the dome vertical centerline and sitting on the horizontal centerline .

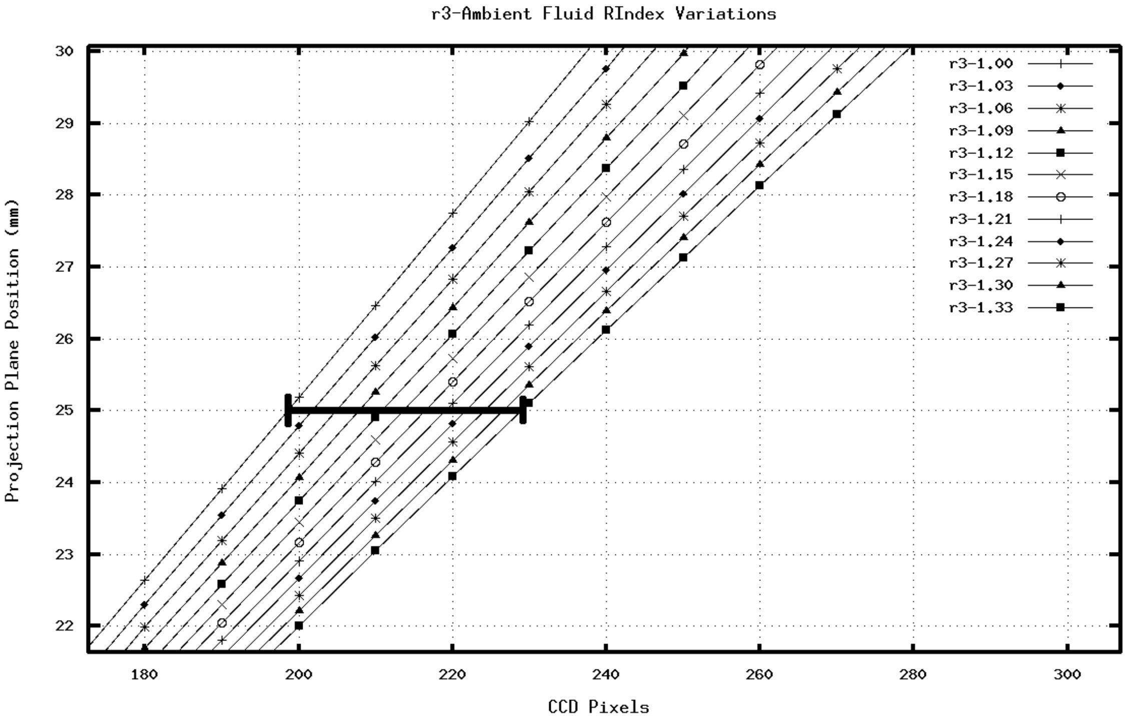


Figure 3.14: Variations of the Ambient Fluid Refractive Index

Fig. 3.14 shows the results from a simulation where the refractive index of the ambient fluid was increased from 1 to 1.33, that is from air to freshwater. The line on the figure shows the displacement in pixels for the given target to be about 30.6 pixels (198.4-229). From the figure, it is also obvious that the change in pixels is larger at the ends of the picture. Had the target been on the 30mm mark on the same normal projection plane the movement in pixels due to the change in the ambient fluid refractive index would be larger. This would give a higher resolution in real applications and the ability to record smaller changes of fluid refractive index.

3.5.3.2 Experimental Setup

The methodology that was followed is described herebelow: A Videoray Pro 4 (ROV) was used for the experiment 3.15. The dome was dismount from the ROV and the calibration of camera took place [48], [97], [107], where the internal parameters of the camera, were calculated in air. The physical model explains the course of a light ray as it cross the three mediums. A parameter identification procedure is carried out to determine the parameters of the model.

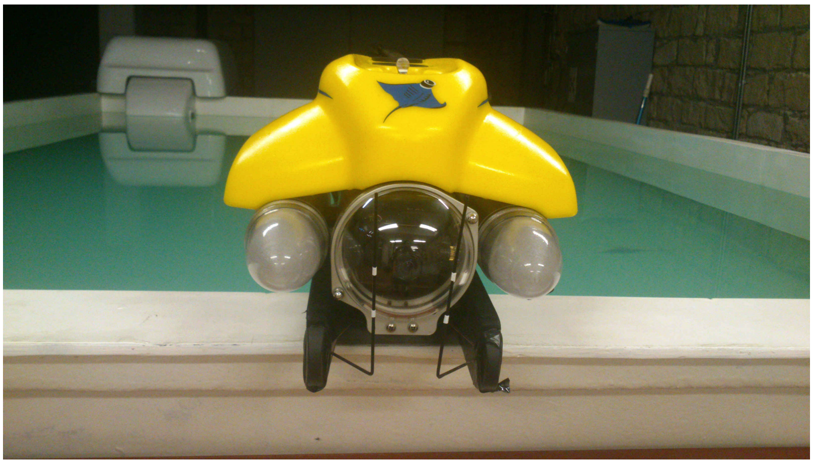


Figure 3.15: Experimental Setup

For the experiments two targets were fitted on either side of the dome as shown in Fig 3.16 and 3.17 . These targets consist of two wires running the vertical distance of the camera view. They are both at 25mm and on either side of the dome center making a total distance of 50mm between their external edges.Two targets were used in order to have double the pixel difference and the possible resolution. Both targets are fixed on the AUV on a plane 32mm from the dome tip, or 82mm from the dome base. Images of the targets were taken with the AUV camera in air and water. The images have shown the targets to move an average of 36.15 pixels towards the outside of the image, as predicted by the simulation. The magnitude difference in target displacement between simulation and experiment is 5.55 pixels (30.6 - 36.15). The difference between the predicted and the experimental results can be attributed to two basic factors : (a) camera rotation and (b) dome imperfections. These two aspects were not taken into account in the current work and are being consider as a future work.

Figure 3.16: Targets fitted at 50mm apart on a plane 82mm from the dome base(Air)

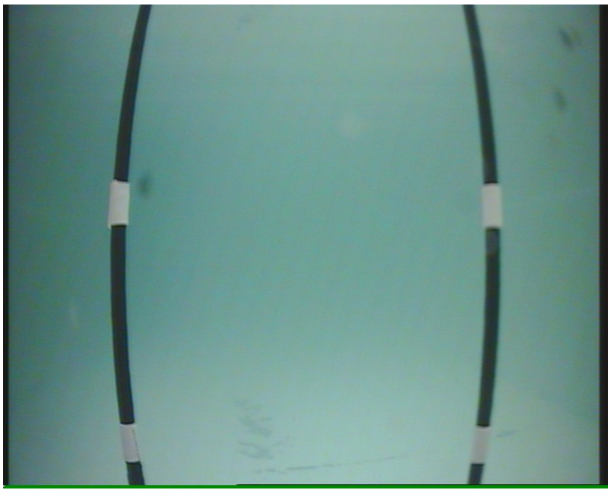


Figure 3.17: Targets fitted at 50mm apart on a plane 82mm from the dome base(Underwater)

Using the on-line adaptive estimation algorithm we can now on-the-fly estimate the refractive index of the medium ROV is submerged in.

3.5.4 Experimental evaluation of the LVS in the laboratory

Two series of tests were performed in controlled environment in order to evaluate the performance of the LVS system in the laboratory. The tests validated the performance of the LVS system in linear distance and angular measurements from a fixed target in front of the

ROV. The first series of experiments included positioning and aligning the ROV at specific positions in the water tank and comparing the LVS measurements against the known posture of the ROV. The ROV was aligned perpendicular to the water tank wall and was fixed underwater at 374*mm*, 438*mm*, 525*mm* and 659*mm* from the wall. At each position an array of measurements were performed to determine the distance $d^{S'}$ reported by the LVS. Fig. 3.18 depicts the error in the distance reported by the LVS. As can be seen the LVS accuracy is within 2.3% of the measured distance.

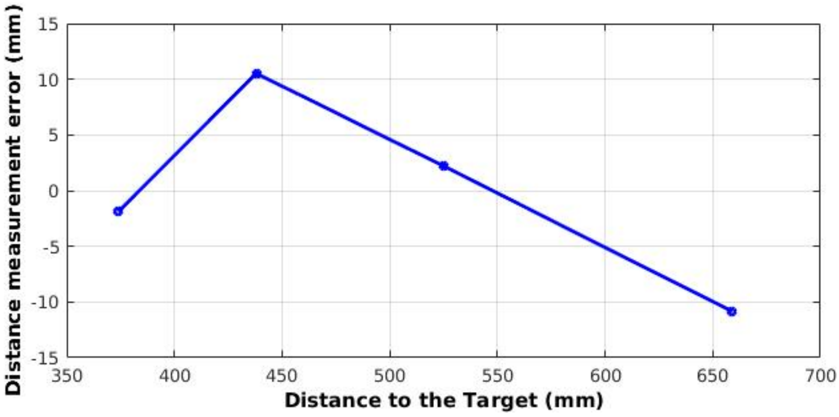


Figure 3.18: d^S error versus distance.

The second series of experiments included positioning the ROV at a fixed distance from the water tank wall and changing the pitch and then the yaw angle of the ROV. The obtained measurements were compared against the relative orientation obtained through the TCC sensor. The results are depicted in Fig. 3.19 for the relative pitch validation experiment and in Fig. 3.20 for the relative yaw validation experiment.

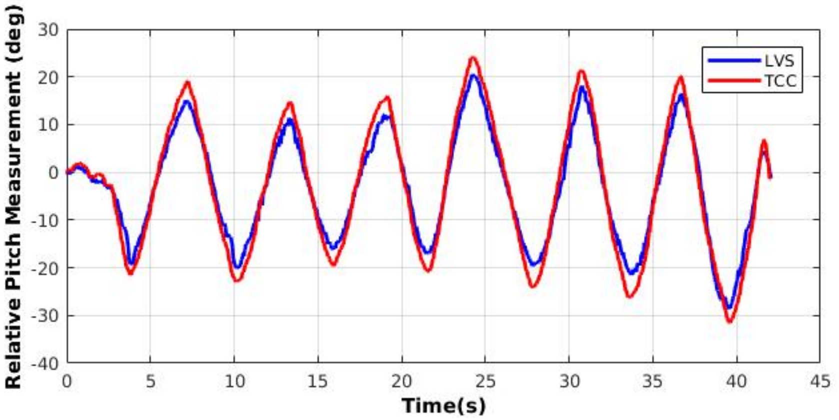


Figure 3.19: Relative pitch validation experiment in the laboratory water tank. The blue line denotes the measurements of the LVS while the red line is the measurement from TCC.

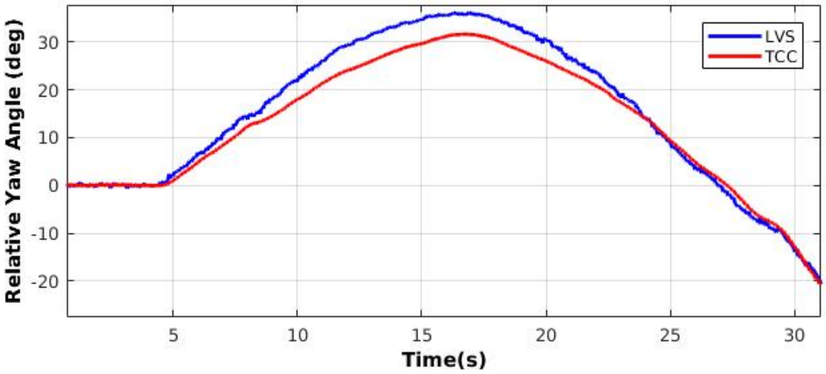
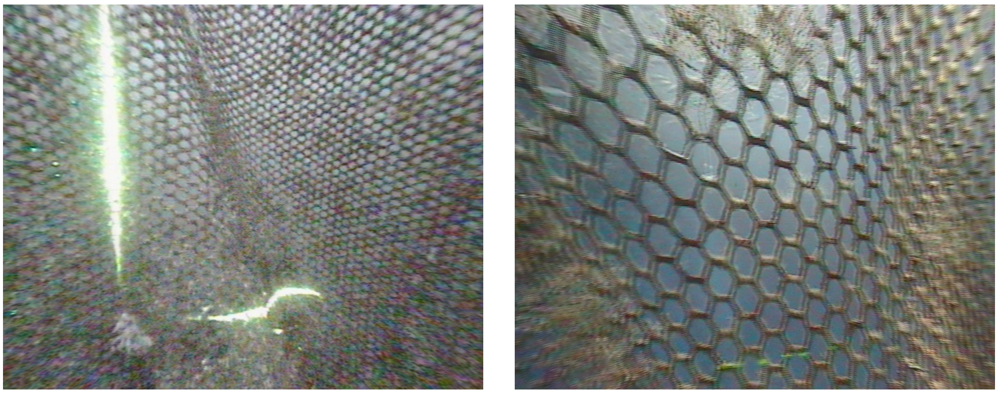


Figure 3.20: Relative yaw validation experiment in the laboratory water tank. The blue line denotes the measurements of the LVS while the red line is the measurement from TCC.

As can be seen from both experiments the orientation error is within 14% of the orientation reported by the TCC sensor. The main source of error in the experiments was caused by the scattering of the line laser beam in the transverse to the laser plane direction. This can be rectified by appropriately tuning the focus of the line laser beams or using better laser line optics.

3.5.5 Experimental evaluation at an offshore aquaculture installation

In order to test the behavior of the LVS system in a real environment, experimental tests of the LVS were performed at an offshore aquaculture installation as shown in Figs. 3.10 and 3.21.



(a) The mesh filter algorithm was not applicable to some portions of the fish-net.

(b) The LVS was not applicable under intense sunlight.

Figure 3.21: Experimental evaluation at an offshore aquaculture installation.

The experiment, demonstrating the operation of the LVS system while measuring the relative ROV posture to a mesh-like structure such as the aquaculture fish net. Fig. 3.22 shows the LVS measurements of distance, of the ROV to the fish-net as the ROV hovers near the fish-net. Fig. 3.23 depicts the relative pitch of the ROV to the fish-net and Fig. 3.24 depicts the relative yaw of the ROV, as measured by the LVS system during the experiment. The LVS interprets the variations and foldings of the fish net using the methodology presented in section 3.3.2. Since this experiment was performed on an operating aquaculture fish net, with non-planar, time-varying geometry due to the sea currents and net flexibility and foldings, a comparative analysis of the measurements to ground truth was not feasible to provide an estimation of its performance. The way that we chose to evaluate the performance of the LVS was by implementing a PID controller on the ROV to stabilize its distance and yaw angle with respect to the fish net using LVS feedback. The distance set-point was set at $550mm$ and the relative yaw angle set-point was set to 0^o. The ROV pitch angle is passively stable and non-actuated. Due to algae growth and increased scattering due to higher biomass concentration in the proximity of the fish-net Fig. 3.21(a), the mesh filter algorithm was not applicable to some portions of the fish-net, while the LVS was not applicable under intense sunlight Fig. 3.21(b). From the experiments performed, as can be seen from Fig. 3.22 the reported distance indeed varies around $550mm$ as expected. From Fig. 3.24 the reported yaw also varies around 0^o as expected. From $1s$ to $2.5s$ the performance of the controller is much better since the LVS was reflecting on a more ”flat” region of the fish net during that time. Regarding the relative pitch angle as can be seen from Fig. 3.23, it varies around 25^o which is also as expected.

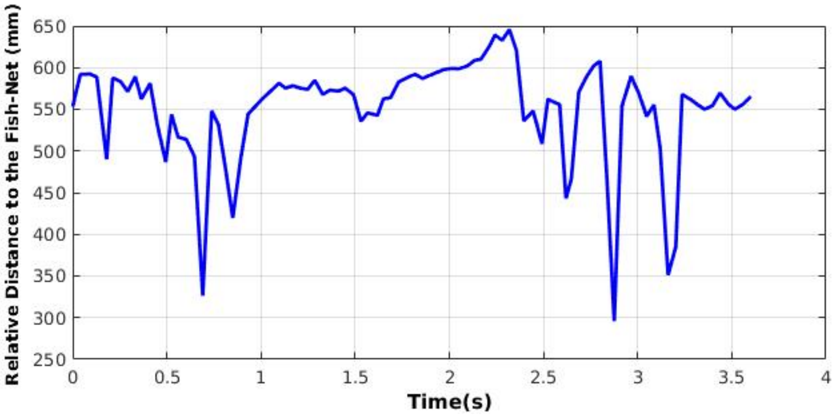


Figure 3.22: Relative distance from the ROV to mesh-like structure(fishnet) at sea.

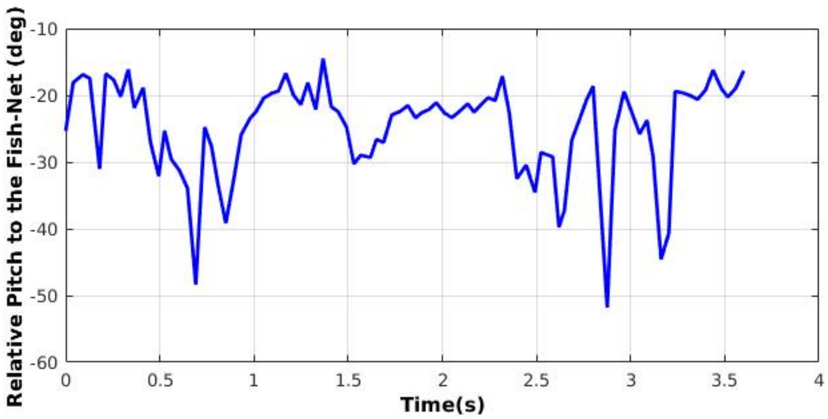


Figure 3.23: Relative pitch from the ROV to the target(fishnet) at sea.

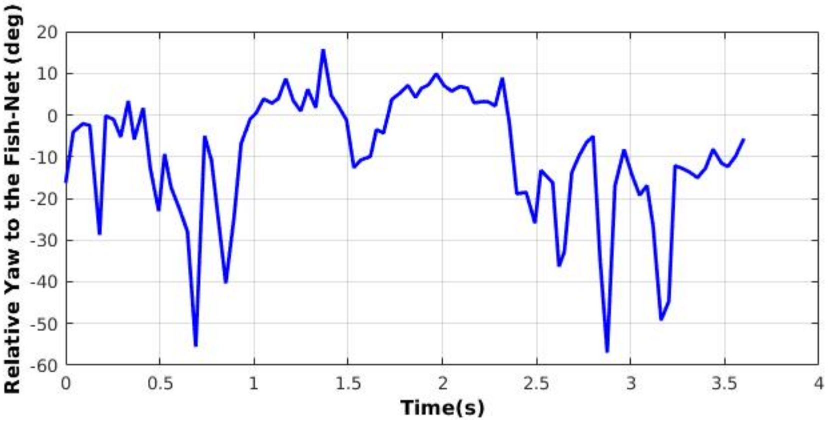


Figure 3.24: Relative yaw from the ROV to the target(fishnet) at sea.

3.6 Conclusion

The work presented, details the development and experimental validation of algorithms for a novel Laser Vision System (LVS) for measuring an ROV's posture with respect to both solid and mesh like targets in underwater environments. An analytical model for three-medium refraction (air, acrylic, water) that takes into account the non-linear hemispherical optics was developed. The system development was motivated by the need for underwater ROV localization in close proximity to aquaculture fish-nets, and due to the analytic nature of the solution, it is applicable to operation to mediums with varying refractive index, extending its applicable range. The LVS is capable of generating information as point-cloud sets from each laser and, by utilizing the proposed algorithms, high level information like distance and relative orientation of the target with respect to the ROV can be recovered. Due to the regular maintenance required by the system (typical for underwater vehicles and their components), an automatic calibration technique was developed. Furthermore, a spatial filter was developed and demonstrated, in order to discriminate the mesh like targets from other artifacts in the LVS measurements. The algorithms developed in this work are appropriate for real time operation.

Experimental results both in the laboratory and field experiments at an offshore aquaculture installation, demonstrate the performance of the system. During field experiments, the proposed system demonstrated the expected performance that was recorded during laboratory testing, while the limitations of the system's performance when operating near the water surface during intense sunlight and when operating in an environment with increased biomass were recorded.

Further development of this work includes utilizing the developed localization system to close the feedback loop in motion task planning strategies for autonomous underwater structure inspection and maintenance tasks for operations that include but are not limited to aquaculture fishnet fault detection and repair. Future work also includes cooperative object transportation by multiple underwater robots. More specifically, we are investigating the collaborative underwater object transportation problem with only implicit communication by underwater robots. The problem addresses the transportation of a solid object from an initial configuration to a destination configuration by two autonomous underwater agents. The underwater agents cannot communicate between themselves and the only means of communicating their intentions are implicit, by observing the motion of the transported object on their end.

Underwater robot experiments are very difficult to implement due to the underwater environment and are also very costly. Fig. 3.25 and Fig. 3.26 shows preliminary realistic scenarios developed in ROS/Gazebo platform based on the Unmanned Underwater Vehicle (UUV) simulation with Gazebo [78].

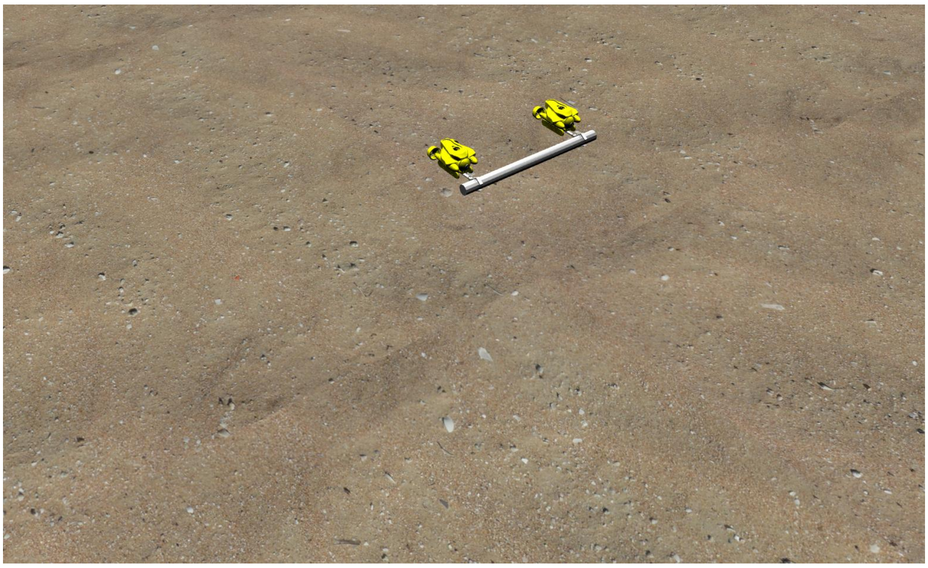


Figure 3.25: Cooperative object transportation simulation in ROS/Gazebo.

The simulations were developed in order to test the developed algorithms in realistic scenarios.

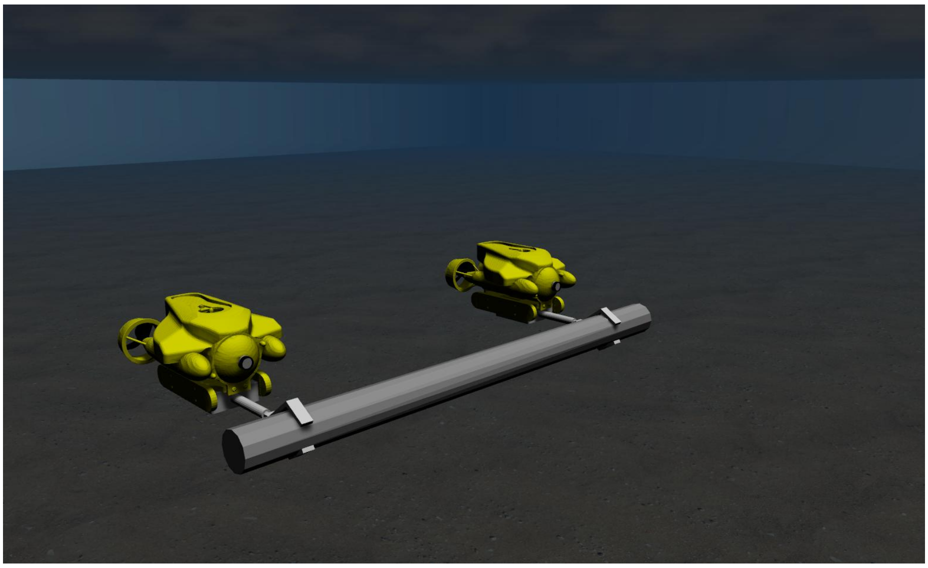


Figure 3.26: Cooperative manipulation task simulation in ROS/Gazebo.

Chapter 4

Towards Multi-Sensor UKF-based Localization of an Underwater Robotic Vehicle for Underwater Inspection Operations

This chapter presents a methodology for localizing an autonomous tethered underwater robotic vehicle deployed for inspection operations in an underwater environment for the AQUABOT project [6][1]. The developed methodology is based on an Unscented Kalman Filer (UKF) to fuse information from onboard sensors along with a'priori knowledge about the aquaculture's geometry and the underwater robot's hydrodynamics. The proposed algorithm incorporates several modifications that take into account the sensors' measurements that are available asynchronously and with varying frequency. The performance of the proposed methodology is assessed through accuracy and consistency metrics generated from simulation benchmarks. The results in this chapter were originally presented with the author contribution in [86] and [23].

4.1 Introduction

Underwater robotics has experienced a dramatic growing in the last years due to its potentials in different undersea applications including gathering data for oceanographic needs, pipeline and harbor inspection, and monitoring aquacultures. In the report of the European Commission [44], aquaculture is called to be the fastest growing food producing sector in the world. In addition, aquaculture facilities and tasks required for their maintenance represent a great application field for underwater robots. In aquaculture industry, the main

---

[1] This work was supported by the European Regional Development Fund and the Republic of Cyprus through the Research Promotion Foundation under research grant ΑΕΙΦΟΡΙΑ/ΓΕΩΡΓΟ/0311(ΒΙΕ)/08.

problem is considered to be escapes of fish from the fish net cages. According to the Norwegian report of escape incidents [37], more than 60% of escapes are caused by structural failures of fish farm equipment. The main reasons for such faults are abrasion and tearing of mooring ropes, anchor chains, or holes in the fish nets from fish biting. In order to prevent such incidents, the regular inspection of all subsystems and early detection of net holes and other types of system faults are mandatory tasks in the fish producing industry.

Currently, such operations are provided by human divers. However, this task is associated with risks of human health and lives. These risks have been increasing recently because of the tendency to push fish cage systems to far-shore sites with more severe waves and currents [61, 109]. In this case, underwater unmanned vehicles appear to be a safe and cost-efficient alternative to the human diving inspection by avoiding the health and human life risks. The current work presents certain results of a research project aimed to develop an autonomous system for visual inspection of fish farm nets and moorings by using a tethered autonomous underwater vehicle (AUV). The GPS-denied undersea environments make the autonomous operation of underwater vehicles to be one of the most challenging problem. The effective and accurate localization and control is enabled by a fusion of data from onboard sensors and a-priory knowledge about dynamic behavior of the aquaculture.

The underwater vehicle used in this project is a commercially available tethered underwater vehicle a VideoRay Pro IV. Although the tether could introduce some limitations during the underwater operation of the robot, the main benefit of its usage is power regularity. In addition, it is used to transfer video information about aquaculture structure being under inspection. The basic sensor suite of the vehicle includes a front facing camera, inertial measurement unit (IMU) for measuring angular rates and body accelerations, and a depth sensor. In order to improve the accuracy of localization and control, three line lasers were added to the sensor suite making the vehicle be able to measure the distance to the underwater objects and obstacles.

To localize a vehicle in an underwater environment where the signals from the GPS are unavailable, a sensor fusion approach based on Unscented Kalman Filter (UKF) was applied. In this approach, we fused information from the sources of two kinds: the hardware navigation sensors described earlier and aquaculture structures' geometry modeled a'priory. In most cases, the aquaculture underwater constructions include arrays of fish net cages connecting between each other by ropes and chains. Such an aquaculture represents a very flexible structure that depends on the velocity of underwater currents and has strong nonlinearities.

The rest of the chapter is organized as follows. Section 4.2 provides the description of the current state of the art and the progress beyond it contributed by this work. Section 4.3

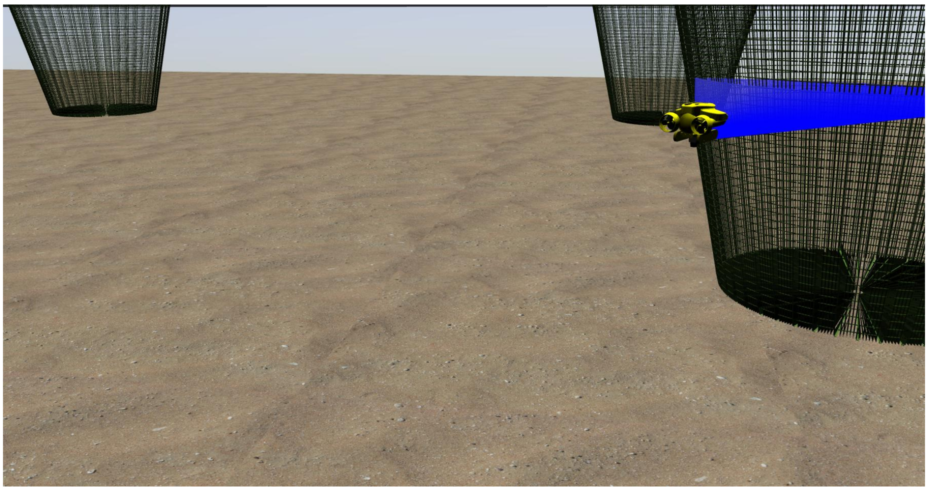


Figure 4.1: 3D simulation of Rov with line laser in the proximity of the aquaculture fishnets.

presents the sensor models where is used for the sensor fusion while Section 4.4 propose an underwater localization based on an asynchronous UKF. Finally, Section 4.5 conclude the chapter with the simulation results and discussion.

4.2 State of the Art and Progress Beyond It

This work focuses on underwater GPS-denied localization of an autonomous vehicle and covers several topics that are under the great interest of the research community.

The task of underwater localization is considered to be a very challenging task due to the sensor usage limitation (pure visibility, unavailability of the GPS signals, etc.) and high disturbances undersea. The popular choice for the estimating the vehicle's state are an Extended Kalman filter (EKF) or an Unscented Kalman filter (UKF). Kottas and Roumeliotis in [63] investigated the observability properties of a navigation system based on camera and IMU sensors and proposed an algorithm utilizing this analysis to improve the EKF state estimator. Bloesh et al [11] considered a measurement fusion problem from an inertial sensor and a monocular camera in form of optical flow by using an UKF to estimate a robot pose over time.

This work contributes to the field of underwater localization by proposing an asynchronous UKF originated from the work of Karras et al. [57]. The asynchronous UKF takes into account two main features of the information and measurement sources used for localization: the varying frequency of the measurements from the sensors suite of the vehicle and their asynchronous availability for the filter. The asynchronous UKF methodology does not required the rearrangement of the filter algorithm and utilizes a special "speed-up" implementation of the vehicle's state prediction step.

Several research works have been conducted recently to model a sensor suite of underwater vehicles. Lynen et al. [76] presented measurement models of a camera-IMU system consisting of an IMU and a spherical camera along with the optical flow method for the Extended Kalman Filter (EKF)-based state estimation. Karras et al. [59] described a model of a laser-based vision system consisting of two laser pointers and a single camera mounted on a underwater vehicle. They presented modeling and calibration methodology of such a system along with the target tracking and steering control algorithms. Wang and Clark [101] presented a dynamic model of the VideoRay Pro III Remotely Operated Vehicle (ROV). They conducted a series of intensive experiments for parameter identification of the vehicle's hydrodynamic derivatives and thruster coefficients based on the assumption of decoupled vehicle's dynamics. The theoretical estimation of the parameters along with the model verification showed the assumption is reasonability of this assumption for typical operating conditions of underwater vehicles.

In this work, we present a model of the vehicle's sensor suite and of the underwater vehicle that are used in the localization scheme. The main contribution of the work in the field of sensor modeling includes modeling of a novel Laser-Vision System (LVS) that combines a CCD camera and three line lasers. Such a system is used to determine the distance from the current vehicle position to a target (aquaculture underwater structures). In addition, we propose a model and an algorithm of the optical flow concept that is used as one of the sources for the information fusion. The model and the algorithm utilize the data extracted from camera's images and lasers' measurements. The VideoRay Pro IV vehicle used in this project was modeled as well by applying dynamic parameter identification procedure to determine the vehicle's parameter and derivatives. To the best of our knowledge, no such modeling of the VideoRay Pro IV vehicle has appeared in the literature.

In order to improve the accuracy of the proposed localization approach, we use a'priory knowledge about the aquaculture geometry that can be modeled for a certain velocity of the ocean current. This problem has been extensively investigated recently along with the development of different numerical models of the fish net dynamics. Aarsnes et al. [1] were among the first that conducted theoretical studies to calculate forces acting on the net panel and experiments to determine the net-current interaction and velocity reduction within net cage systems. Tsukrov et al. [38] proposed a consistent finite element method to model the hydrodynamic response of net panels. They considered a one-dimensional finite element with unit length that is under the action of hydrodynamic, buoyancy, inertia and elastic forces. After the integration, the system of nonlinear equations describing a system model in a moving fluid was constructed. The model was verified through experiments in the steady flow. Lader et al. [67] developed a dynamic model for 3D net structures exposed to waves and current. They divided the net structure into square super-elements for which the hydrodynamic forces were calculated. Each node of the super-element was connected to others by nonlinear springs that produced the structural forces. The hydrodynamical forces acting on each super-element were equally distributed to four nodes of the element, and the system of motion equations for each node could be solved. Zhao et al. [108] proposed a numerical model based on a lumped-mass method where the hydrodynamic drag and other external forces where applied for the net twines and lumped masses. They assumed that there are lumped masses at each knot of the net and at the center of the mesh bar. They conducted experiments and provided comparison with other approaches as well. Huang et al. [52] developed a numerical model to analyze the dynamic behavior of a net-cage system in currents. The net cage was divided into plane surface elements on which the hydrodynamic forces were calculated. The comparison between the numerical and experimental results were provided to estimate the net volume reduction.

The main contribution of this work in the modeling of aquaculture structures is twofold. First, the calculation of the hydrodynamic forces (drag, inertia and buoyancy) in the lumped-mass method applied in this work was done with respect to the net meshes' bar that are massless while the gravity force was associated only with the net knots. In addition, the proposed methodology for estimating the aquaculture geometry is considered to be a part of the online sensor fusion process for the localization task. For this purpose, the geometry estimator consists of offline and online steps. In offline, the aquaculture geometry is calculated for a set of ocean current velocities and stored in the vehicle's memory. During online operation, the geometry is updated based on in-situ measurements of the current's velocity.

4.3 Vehicle and Sensor Models

4.3.1 Aquaculture Model

In this work, we assume a "gravity" net cage system that consists of a floating collar, a netting system, weights, mooring ropes and anchors. The schematic diagram of such a system is shown in 4.2. The "lumped-mass method" concept is used here to model a fish net structure. According to this concept, the net is divided into discrete net elements with lumped masses at the element nodes connected by massless springs. In this case, each net mesh represents a structure of four interconnected bars that are subject to the external forces applied at the center of the element [86]. The external forces include the drag force

$\mathbf{F}_D$, the inertia force $\mathbf{F}_I$, the gravity $\mathbf{F}_G$ and the buoyancy $\mathbf{F}_B$ that can be calculated as:

$$\begin{aligned} \mathbf{F}_D &= \frac{1}{2}\rho C_D A \mathbf{U}_R \left|\mathbf{U}_R\right|, \\ \mathbf{F}_I &= \rho C_a V \dot{\mathbf{U}}_R + \rho V \dot{\mathbf{U}}, \\ \mathbf{F}_G &= M\mathbf{g}, \\ \mathbf{F}_B &= (\rho - \rho_b) V \mathbf{g}, \end{aligned} \tag{4.1}$$

where ρ is the water density, C_D is the drag coefficient, A is the reference area (an area of the body perpendicular to the flow), $\mathbf{U}_R = \mathbf{U} - \mathbf{V}$ is the relative flow velocity, $\mathbf{U}$ is the flow velocity, $\mathbf{V}$ is the body velocity, C_a is the added mass coefficient, V is the body volume, M is the body mass, $\mathbf{g}$ is the gravitational constant vector, ρ_b is the body density.

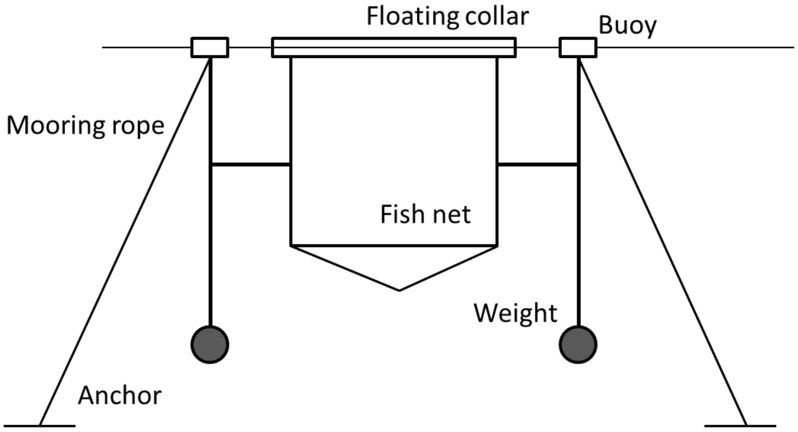


Figure 4.2: FishNetCage.

We assume the bar elements as of cylindrical form. The drag force acting on a such element may be divided into two components: the normal component and the tangential component, relative to the bar [90]. These forces can be represented as follows:

$$\begin{aligned} \mathbf{F}_n &= \frac{1}{2}\rho C_n A \left|\mathbf{U}_{Rn}\right|^2 \mathbf{e}_n, \\ \mathbf{F}_t &= \frac{1}{2}\rho C_t A \left|\mathbf{U}_{Rt}\right|^2 \mathbf{e}_t, \end{aligned} \tag{4.2}$$

where $A = dl$ is a cross-sectional area of the bar element, d is the element diameter which is the same as the twine thickness, l is the length of the bar element, C_n and C_n are the normal and tangential drag coefficients, $\mathbf{U}_{Rn}$ and $\mathbf{U}_{Rt}$ are the normal and tangential relative velocity components, $\mathbf{e}_n$ and $\mathbf{e}_t$ are the unit vectors in the normal and tangential directions, respectively. The drag coefficients are functions of the Reynolds number and calculated according to [18].

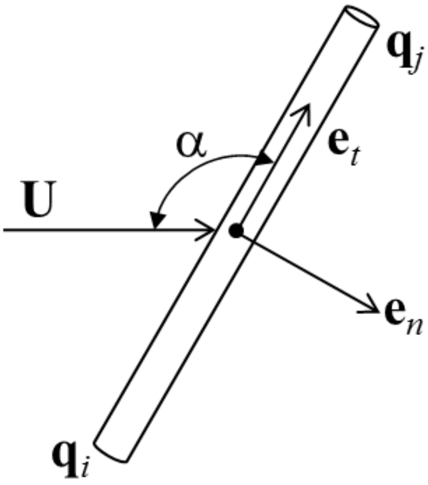


Figure 4.3: Normal and tangential unit vectors and angle of attack definitions.

The normal and tangential velocity components are calculated as:

$$\begin{aligned} \mathbf{U}_{Rn} &= \mathbf{U}_R \sin\alpha, \\ \mathbf{U}_{Rt} &= \mathbf{U}_R \cos\alpha, \end{aligned} \tag{4.3}$$

where α is the angle of attack defined as an angle between the flow velocity direction and the normal to the net element as shown on 4.3.

The flexibility of the net structure can be described by elastic (tension) forces of the twines. We assumed that the net element nodes are connected through massless springs, so each node is subject to four elastic forces from the neighboring nodes. The force in the spring between nodes i and j is given by (4.4):

$$\mathbf{F}_E^{ij} = \sigma_{ij}\mathbf{e}_{bar}^{ij}, \tag{4.4}$$

where σ_{ij} is the force magnitude, and $\mathbf{e}_{bar}^{ij}$ is the bar unit vector.

The force magnitude (tension) may be expressed as:

$$\sigma_{ij} = \begin{cases} C_1\varepsilon_{ij} & \text{if } \varepsilon_{ij} > 0, \\ 0 & \text{if } \varepsilon_{ij} \le 0, \end{cases} \tag{4.5}$$

where C_1 is the elastic coefficient of the twine, ε_{ij} is the elongation of the spring ij.

The motion of a net structure may be represented by the motion of its mass points under external and structural forces. By applying the Newton's law for the equations 4.1, 4.2 and 4.4, we get the following motion equation for each lumped mass point i:

$$(m_i + \Delta m_i)\frac{\partial^2 \mathbf{q}_i}{\partial t^2} = \mathbf{F}_i^{ext} + \mathbf{F}_i^{int}. \tag{4.6}$$

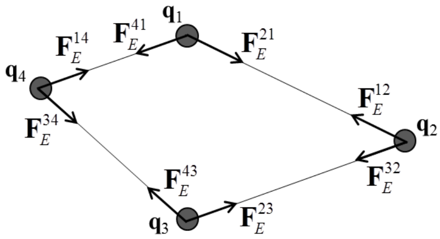


Figure 4.4: The structural forces of mesh bar elements.

We assume a steady current flow, so the inertial force is equal to zero: $\mathbf{F}_I = 0$. The external forces $\mathbf{F}^{ext}$ and the internal force $\mathbf{F}^{int}$ are calculated for each bar k and then uniformly distributed to the corresponding mass points. For each point, the applied forces will be as follows:

$$\begin{gathered}\mathbf{F}_i^{ext} = \frac{1}{2}\sum_{k=1}^{4}\left(\mathbf{F}_n^k + \mathbf{F}_t^k\right) + \mathbf{F}_{Gi} + \mathbf{F}_{Bi}, \\ \mathbf{F}_i^{int} = \mathbf{F}_E^{i,i-1} + \mathbf{F}_E^{i,i+1},\end{gathered} \tag{4.7}$$

where m_i is the mass of the point i, $\Delta m_i = \rho C_a V$ is the added mass of the point i, k the index of corresponding bar elements. The distribution of the forces acting on the point i is shown in 4.5.

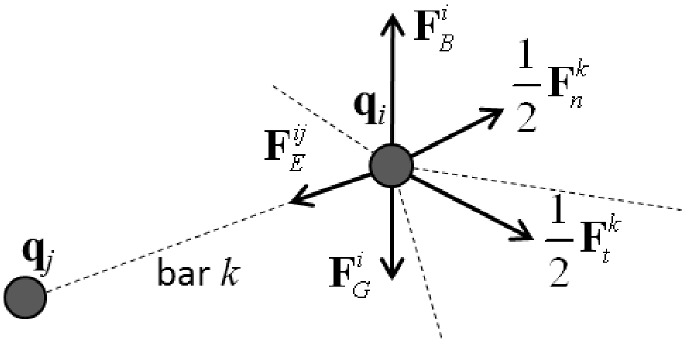


Figure 4.5: The force distribution for a mass point i.

The resulting system of equations is non-linear with $N \times 3$ unknown mass point coordinates and may be solved using an iterative numerical method.

4.3.2 Vehicle Model

We consider the six degree of freedom (DOF) dynamical model for the ROV that is given by the following system of equations [35]:

$$\begin{gathered}\mathbf{M}\dot{\nu} + \mathbf{C}(\nu)\nu + \mathbf{D}(\nu)\nu + \mathbf{g}(\eta) = \tau, \\ \dot{\eta} = \mathbf{J}(\eta)\nu,\end{gathered} \tag{4.8}$$

where **M** is the inertia matrix, **C** is the coriolis and centripetal matrix, **D** is the drag matrix, **g** is the force vector, τ is the trust vector, ν is the body velocities vector, η is the position and Euler angles vector, **J** is the transformation matrix from the body-fixed frame $\{B\}$ to the inertial frame $\{I\}$.

The matrices **M**, **C** and **D** depend on the hydrodynamic parameters of the vehicle that represent the derivatives of the hydrodynamic forces and moments acting on the vehicle with respect to the vehicle velocities and accelerations:

$$\begin{aligned} \mathbf{M} &\sim \frac{\partial X}{\partial \dot{u}}, \frac{\partial Y}{\partial \dot{v}}, \cdots, \frac{\partial N}{\partial \dot{r}}, \\ \mathbf{C} &\sim \frac{\partial X}{\partial \dot{u}}, \frac{\partial Y}{\partial \dot{v}}, \cdots, \frac{\partial N}{\partial \dot{r}}, \\ \mathbf{D} &\sim \frac{\partial X}{\partial u}, \cdots, \frac{\partial N}{\partial r}, \frac{\partial X}{\partial u\,|u|}, \cdots, \frac{\partial N}{\partial r\,|r|}. \end{aligned} \tag{4.9}$$

For using the dynamic model 4.8 in the fusion algorithm, the model was discretized and augmented with the linear accelerations in the body frame $\mathbf{a} = [a_x, a_y, a_z]^T$. The acceleration model was derived based on the approach proposed in [57]. The final discrete model of the vehicle can be written as:

$$\begin{aligned} \nu_{k+1} &= \nu_k + \mathbf{M}^{-1}\left(\tau_k - \mathbf{C}(\nu_k)\,\nu_k - \mathbf{D}(\nu_k)\,\nu_k \right. \\ &\quad \left. - \mathbf{g}(\nu_k)\right)\Delta t, \\ \eta_{k+1} &= \eta_k + \mathbf{J}(\nu_k)\,\eta_k \Delta t, \\ \mathbf{a}_{k+1} &= -\mathbf{a}_k + 2\mathbf{M}^{-1}\left(\tau_k - \mathbf{C}(\nu_k)\,\nu_k - \mathbf{D}(\nu_k)\,\nu_k \right. \\ &\quad \left. - \mathbf{g}(\nu_k)\right). \end{aligned} \tag{4.10}$$

The ROV process model we use in the UKF algorithm can be written as:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) + \mathbf{w}_{ROV}, \tag{4.11}$$

where $\mathbf{x} = \left[\nu_1^T, \nu_2^T, \eta_1^T, \eta_2^T, \mathbf{a}^T\right]^T$ is the state vector, **u** is the control (thrust) vector, $\mathbf{w}_{ROV} \sim (0, \mathbf{R}_{ROV})$ is the white noise with zero mean and the covariance matrix $\mathbf{R}_{ROV}$.

4.3.3 Accelerometer and Gyroscope Models

We assume that the accelerometer outputs the linear accelerations **a** in the body frame, while the gyroscope measures the angular velocity ν_2 in the body frame of the ROV. These two sensors are modeled as follows:

$$\mathbf{z}_{IMU} = h_{IMU}(\mathbf{x}, \mathbf{w}_{IMU}), \tag{4.12}$$

where $\mathbf{w}_{IMU} \sim N(0, \mathbf{R}_{IMU})$ is the white noise with zero mean and covariance matrix $\mathbf{R}_{IMU}$.

In this equation, the vector $\mathbf{z}_g$ consists of gyroscope and accelerometer measurements as follows: $\mathbf{z}_{IMU} = \left[\mathbf{z}_g^T, \mathbf{z}_a^T\right]^T$, where the measurements from the gyroscope are defined as:

$$\mathbf{z}_g = [p, q, r]^T + \mathbf{w}_g, \tag{4.13}$$

and the measurements from the accelerometer can be written as:

$$\mathbf{z}_a = \mathbf{a} + \mathbf{w}_a, \tag{4.14}$$

where $\mathbf{w}_g$ and $\mathbf{w}_a$ are the white zero mean noises.

4.3.4 Tilt-Compensated Compass and Depth Pressure Sensor Models

The tilt-compensated compass and the depth pressure sensor used in this work provide measurements of the current depth of the vehicle and the earth magnetic vector direction.

The depth pressure sensor is modeled based on the Pascal's Law for the hydrostatic pressure that is the force exerted on an object due to the weight of water above it:

$$\mathbf{z}_{PS} = h_{PS}(\mathbf{x}, \mathbf{w}_{PS}) = \rho g z + p_a + \mathbf{w}_{PS}, \tag{4.15}$$

where ρ is the water density (1000 $\mathrm{kg\,m^{-3}}$ for fresh water), g is the gravitational constant, z is the depth below the water surface, p_a is the atmosperic pressure, and $\mathbf{w}_{PS} \sim N(0, \mathbf{R}_{PS})$ is the white noise with zero mean and covariance matrix $\mathbf{R}_{PS}$.

The tilt-compensated compass sensor provide the Euler angles of the vehicle and is modeled with the following equation:

$$\mathbf{z}_{MS} = \mathbf{V}_m + \mathbf{w}_{MS}, \tag{4.16}$$

where $\mathbf{V}_m = [\phi, \theta, \cos\psi, \sin\psi]$ is the Euler angles while the yaw angle is represented by its trigonometric functions to avoid the discontinuity at 0^o; $\mathbf{w}_{MS}$ is the white noise with zero mean and covariance matrix $\mathbf{R}_{PS}$.

4.3.5 Laser Vision System Model

The LVS sensor [25] consists of a CCD camera and a set of three lasers projecting the lines in the form of a triangle on the target as shown on 5.1. The CCD camera is enclosed by a hemispherical dome. Since the acrylic dome and an underwater environment affect the vision system of the ROV, a special camera calibration process was used [26]. The LVS provides the distance and orientation of a target with respect to the vehicle [87].

The model of the fish net geometry can be considered as an additional information source and is used in the LVS measurement model as follows. The vehicle navigates around a fish cage inspecting and monitoring faults and failures of the fish net. The camera of the robot is pointing towards a target while the lasers create a triangle on the surface of the target. Since such a triangle defines a plane on the targets's surface, a relative distance and a relative orientation of the target with respect to the vehicle body frame can be calculated. The fish net geometry $\mathscr{F}$ is represented as a quad mesh with $V_{\mathscr{F}}$ vertices and $F_{\mathscr{F}}$ faces while the coordinates of the vertices are known a'priory as shown on 4.6. In order to compute the distance d from the vehicle position $v_1 = [x,y,z]^T$ to a mesh face, we first triangulate the mesh faces by splitting each quad cell into two triangles. Then, we draw a ray $\mathbf{p}$ from the origin of the body frame along the X_B-axis of the vehicle:

$$\mathbf{p} = \mathbf{O}_B + \Delta\mathbf{p} \cdot \varepsilon, \quad \varepsilon > 0, \tag{4.17}$$

where $\mathbf{O}_B$ is the origin of the body frame, $\Delta\mathbf{p}$ is the X-column of the rotational matrix $\mathbf{J}_2$. The intersection point $\mathbf{P}$ of this ray with a mesh face containing vertices $\mathbf{V}_0$, $\mathbf{V}_1$ and $\mathbf{V}_2$ can be found as:

$$\mathbf{P} = \frac{-\mathbf{n} \cdot \mathbf{w}}{\mathbf{n} \cdot \Delta\mathbf{p}}, \tag{4.18}$$

where $\mathbf{n}$ is the normal to the current face, $\mathbf{w} = \mathbf{O}_B - \mathbf{V}_0$ is the vector between the ray origin $\mathbf{O}_B$ and the face vertex $\mathbf{V}_0$. By testing whether the intersection point is located inside the corresponding triangle face, the distance from the body frame origin to that face can be found. We proceed with the described procedure for all $2F_{\mathscr{F}}$ faces of the fish net and take the smallest distance. The orientation of the face can be calculated as a normal vector $\mathbf{n}$ to this face:

$$\mathbf{n} = \frac{\mathbf{e}_1 \times \mathbf{e}_2}{|\mathbf{e}_1 \times \mathbf{e}_2|}, \tag{4.19}$$

where $\mathbf{e}_1 = \mathbf{V}_0 - \mathbf{V}_1$, $\mathbf{e}_2 = \mathbf{V}_0 - \mathbf{V}_2$ are the edge vectors.

Thus, the measurement model of the LVS can be described as follows:

$$[d, \mathbf{n}]^T = h_{LVS}(\mathbf{x}, \mathscr{F}) + \mathbf{w}_{LVS}, \tag{4.20}$$

where $\mathbf{w}_{LVS} \sim N(0, \mathbf{R}_{LVS})$ is the zero mean white noise with covariance matrix $\mathbf{R}_{LVS}$.

4.4 Underwater Localization Based on an Asynchronous Unscented Kalman Filter

In this work, we applied an Unscented Kalman Filter (UKF) scheme to estimate the states of the vehicle. Since the measurements from the onboard sensor could arrive with differ-

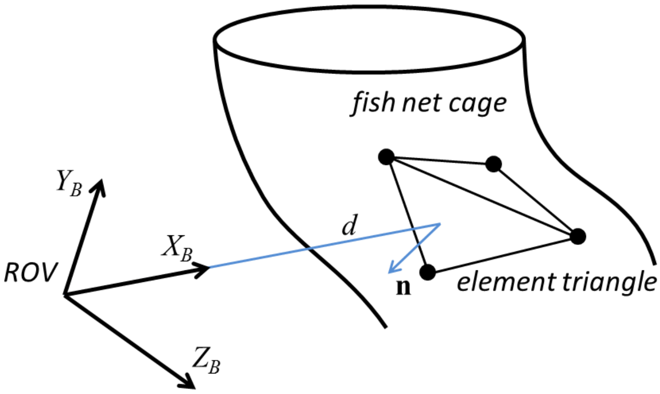


Figure 4.6: The measurement model of the combined Laser Vision System.

ent frequencies and asynchronously, the UKF algorithm was modified to incorporate this feature using the approach similar to [17,57].

The asynchronous UKF algorithm is described in 2 and can be summarized as follows. At the initialization step, a set of weights for the state mean $\mathscr{W}_m = \left\{W_m^{(i)}\right\}$ and state covariance $\mathscr{W}_c = \left\{W_c^{(i)}\right\}$, $i = 0, \cdots, 2n$, where n is the dimension of the state vector, is calculated. Three parameters are used in the weight calculation: the constant α that determines the spread of the sigma points around the state mean, the constant β that represents the a'priory knowledge about the state vector distribution, and the scaling parameter $\lambda = \alpha^2 (n + \kappa) - n$, where κ is the secondary scaling parameters. The exact values of these parameters depend on the properties of a certain dynamic system and usually are set as: $1e-4 \leq \alpha \leq 1$, $\beta = 2$ (for Gaussian distributions), $\kappa = n - 3$ for systems of order $n \leq 3$ or $\kappa = 0$ for high-order systems [31,56,80,99].

The vehicle's onboard sensor suite can be described as a set S of N elements, $N = 5$: $S = \{\mathrm{IMU}, \mathrm{MS}, \mathrm{PS}, \mathrm{LVS}, \mathrm{OF}\}$. Thus, the power set $\mathscr{P}(S)$ of the set S represents all combinations of the simultaneously available sensor measurements. Since the power set includes also an empty set $\varnothing$ that does not have meaning in our case (the state updating starts only when a new measurement from either of the sensors is available), thus the following description of the asynchronous UKF algorithm will be given for the power set of the sensors excluding empty set: $\mathscr{P}_w(S) = \mathscr{P}(S) \setminus \{\varnothing\}$. The indexing set $J = 0, \ldots, N$ of the power set $\mathscr{P}_w(S)$ represents the indexes of the measurements' combinations: $\mathscr{P}_w(S) = \cup_{j \in J} \mathscr{P}_w^j(S)$.

In order to speed up the state estimation process, an additional modification of the standard UKF algorithm was applied in this work. Since one of the most time-consuming steps in the UKF procedure is the sigma points propagation through a system process model, it was decided to take this step out of the procedure and to implement it as a separately running integration process that will be described later on.

The asynchronous UKF algorithm ASYNCHRONOUSUKF starts with the initialization of the filter variables such as the weights for the mean and covariance of the sigma points 4.21 and calculation of the initial sigma points 4.22 according to [80]. The initial sigma points is passed to the integration procedure running in parallel with the filter. When a new measurement $\mathbf{Z}^j$ from either of the sensors' combination $\mathscr{P}_w(S)$ is available, a procedure STATECOVESTIMATION for estimating state and covariance of the vehicle for the current time instance is called.

The state and covariance estimation algorithm STATECOVESTIMATION is described in 3 and is divided into two steps. At the prediction step of the algorithm, the predicted state mean $\hat{\mathbf{x}}$ and the covariance $\mathbf{P_x}$ are calculated as in 4.25 based on the sigma points from the integration process in order to redraw a new set of $2n+1$ sigma points $\mathscr{X}$ as in 4.26. Then at the update step of the algorithm, a new set of the sigma points is calculated and propagated through the corresponding measurement equation to compute the predicted mean $\hat{\mathbf{y}}$ and covariance $\mathbf{P_{yy}}$ of the measurement according to 4.28. The measurement model 4.27 depends on the sensors' combination at the current time instance. In the case, when the measurement was received from only one sensor, the measurement model for the UKF algorithm is formulated in the form of the corresponding sensor model (4.12 is for the IMU measurements, 4.15 is for the pressure sensor measurements, etc.). However, for a certain sensors' combination j of $\mathscr{P}_w(S)$, a combined sensor model h^j is created as a stack of the individual sensor models. For example, the threefold sensor model for the combination of IMU, the magnetometer and the LVS ($j=16$) will be formulated as:

$$h^{16} = \begin{bmatrix} h_{IMU}(\mathbf{x}) + \mathbf{w}_{IMU} \\ h_{MS}(\mathbf{x}) + \mathbf{w}_{MS} \\ h_{LVS}(\mathbf{x}) + \mathbf{w}_{LVS} \end{bmatrix} \tag{4.23}$$

Note that the sensor noise covariance matrix $\mathbf{R}^j$ should be rewritten as well in accordance to the applied sensor model. For example, for the threefold combined sensor model mentioned above it will be as follows:

$$\mathbf{R}^{16} = \begin{bmatrix} \mathbf{R}_{IMU} & \mathbf{0}_{IMU\times MS} & \mathbf{0}_{IMU\times LVS} \\ \mathbf{0}_{MS\times IMU} & \mathbf{R}_{MS} & \mathbf{0}_{MS\times LVS} \\ \mathbf{0}_{LVS\times IMU} & \mathbf{0}_{LVS\times MS} & \mathbf{R}_{LVS} \end{bmatrix}. \tag{4.24}$$

Then, the cross-covariance of the state and measurement $\mathbf{P_{xy}}$ and the filter gain $\mathbf{K}$ are calculated as in 4.29 and in 4.30. At the end of the algorithm for the current step k the state mean and state covariance are updated 4.31.

The sigma points integration process is described in 4 and runs as a separate process parallel to the primary UKF algorithm. The main reason to exclude this step from the

Algorithm 2 Multisensor Asynchronous UKF.

```
procedure AsynchronousUKF($\hat{\mathbf{x}}_0, \mathbf{P}_{\mathbf{x}_0}, \alpha, \beta, \lambda, n, \{\hat{\mathbf{x}}_k\}, \{\mathbf{P}_{\mathbf{x}_k}\}$)
Input: initial state mean $\hat{\mathbf{x}}_0$; initial state covariance $\mathbf{P}_{\mathbf{x}_0}$ , filter parameters $\alpha$, $\beta$ and $\lambda$; the dimensions of the state vector $n$.
Output: state and covariance estimates for each time instance $k$.
    Compute a set of $2n+1$ weights for the mean and covariance of sigma points:
        $$\begin{aligned} W_m^{(0)} &= \frac{\lambda}{n+\lambda} \\ W_c^{(0)} &= \frac{\lambda}{n+\lambda} + \left(1-\alpha^2+\beta\right) \\ W_m^{(i)} = W_m^{(i)} &= \frac{\lambda}{2\left(n+\lambda\right)}, \qquad i = 1,\cdots,2n. \end{aligned} \tag{4.21}$$
    Calculate $2n+1$ sigma points:
        $$\begin{aligned} \mathscr{X}_0 = [&\hat{\mathbf{x}}_0, \\ &\hat{\mathbf{x}}_0 + \sqrt{(n+\lambda)\,\mathbf{P}_{\mathbf{x}_0}}, \\ &\hat{\mathbf{x}}_0 - \sqrt{(n+\lambda)\,\mathbf{P}_{\mathbf{x}_0}}\Big]. \end{aligned} \tag{4.22}$$
    Start the parallel sigma points' integration process SigmaPointsIntegration($\mathscr{X}_0, t_0$).
    Current time step $k = 1$.
    while the vehicle is activated do
        Get measurements $\mathbf{Z}_k^j \in \mathbb{R}^{n_z^j}$, where $n_z^j$ is the dimension of the measurement from the sensor combination $\mathscr{P}_w^j$.
        Get current values of the sigma points from the sigma points' integration process for the time instance $k$:
        $\mathscr{X}_{k|k-1} \leftarrow$ SigmaPointsIntegration($t_k$).
        Run the estimation procedure to get the estimates of the vehicle's state and covariance matrix for the current step:
        $(\hat{\mathbf{x}}_k, \mathbf{P}_{\mathbf{x}_k}) \leftarrow$ StateCovEstimation($\mathscr{X}_{k|k-1}, \mathbf{Z}_k^j$).
    end while
end procedure
```

Algorithm 3 System State and Covariance Estimation for a given time instance.

```
procedure StateCovEstimation($\mathcal{X}_{k|k-1}, \mathbf{Z}_k^j, \hat{\mathbf{x}}_k, \mathbf{P}_{\mathbf{x}_k}$)
    Input: set of sigma points $mathcalX_{k|k-1}$; current measurement $\mathbf{Z}_k^j$ for the time instance $k$.
    Output: state $\hat{\mathbf{x}}_k$ and covariance $\mathbf{P}_{\mathbf{x}_k}$ estimates for the time instance $k$.
    Calculate the weighted mean and covariance estimates for $\mathcal{X}_{k|k-1}$ as:
        $$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \sum_{i=0}^{2n} W_m^{(i)} \mathcal{X}_{k|k-1}^{(i)}, \\ \mathbf{P}_{\mathbf{x}_{k|k-1}} &= \sum_{i=0}^{2n} W_c^{(i)} \left( \mathcal{X}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1} \right) \left( \mathcal{X}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1} \right)^T + \mathbf{Q}, \end{aligned} \tag{4.25}$$
    where $\mathbf{Q}$ is the system process noise covariance matrix.
    Redraw a new set of the sigma points based on updated mean and covariance estimates:
        $$\begin{aligned} \mathcal{X}_{k|k-1}^{*} = \Big[ & \hat{\mathbf{x}}_{k|k-1}, \\ & \hat{\mathbf{x}}_{k|k-1} + \sqrt{(n+\lambda)\,\mathbf{P}_{\mathbf{x}_{k|k-1}}}, \\ & \hat{\mathbf{x}}_{k|k-1} - \sqrt{(n+\lambda)\,\mathbf{P}_{\mathbf{x}_{k|k-1}}} \Big]. \end{aligned} \tag{4.26}$$
    Propagate the new sigma points through the corresponding measurement equation:
        $$\mathcal{Y}_{k|k-1} = h^j \left( \mathcal{X}_{k-1}^{*} \right), \tag{4.27}$$
    Calculate the weighted mean and covariance estimates for the measurements' prediction:
        $$\begin{aligned} \hat{\mathbf{y}}_{k|k-1} &= \sum_{i=0}^{2n} W_m^{(i)} \mathcal{Y}_{k|k-1}^{(i)}, \\ \mathbf{P}_{\mathbf{yy}_{k|k-1}} &= \sum_{i=0}^{2n} W_c^{(i)} \left( \mathcal{Y}_{k|k-1}^{(i)} - \mathbf{y}\hat{\mathbf{y}}_{k|k-1} \right) \left( \mathcal{Y}_{k|k-1}^{(i)} - \hat{\mathbf{y}}_{k|k-1} \right)^T + \mathbf{R}^j, \end{aligned} \tag{4.28}$$
    where $\mathbf{R}^j$ is the measurement noise covariance matrix of the corresponding sensor.
```

```
    Estimate the cross-covariance between the state and measurement predictions:
        $\mathbf{P}_{\mathbf{xy}_{k|k-1}} = \sum_{i=0}^{2n} W_c^{(i)} \left(\mathscr{X}^{*(i)}_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)\left(\mathscr{Y}^{(i)}_{k|k-1} - \hat{\mathbf{y}}_{k|k-1}\right)^T.$   (4.29)
    Calculate the filter gain:
        $\mathbf{K}_k = \mathbf{P}_{\mathbf{xy}_{k|k-1}} \mathbf{P}^{-1}_{\mathbf{yy}_{k|k-1}}.$   (4.30)
    Update the state mean and covariance matrix:
        $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_k - \hat{\mathbf{y}}_{k|k-1}\right),$
        $\mathbf{P}_{\mathbf{x}_k} = \mathbf{P}_{\mathbf{x}_{k|k-1}} - \mathbf{K}_k \mathbf{P}_{\mathbf{yy}_{k|k-1}} \mathbf{K}_k^T.$   (4.31)
end procedure
```

filter is to reduce the computational load and speed up the filter. Basically, the step of propagation the sigma points set through the system model is the most time consuming step of the UKF algorithm. In general case, we have $2n+1$ sigma points each of which is of $\mathbb{R}^n$. For our case $n = 15$, thus there are 465 equations that should be integrated at each time step of the filter.

With the proposed modification, the numeric integration of the nonlinear equations of the system states for all sigma points is running all the time. When a new measurement $\mathbf{Z}^j$ arrives, this process outputs a set of sigma points for the current time that will be used afterwards in AsynchronousUKF. During the UKF execution, the sigma point integration process is slowing down to the "idle" mode. When new estimates of the system state mean and covariance are available, they are passed back to SigmaPointsIntegration that resumes the integration process. In order to "catch up" the time, the speed of the integration is increasing until the inner tim of the integration will be equal to the real time of the system. This process is schematically shown on 4.7.

4.5 Results and Discussion

4.5.1 Simulation Results

The developed localization algorithm was validated through various simulations under realistic levels of noises and disturbances.

The simulation scenario was to navigate the vehicle downwards along the aquaculture height with the constant vertical velocity $w = 0.1\,\mathrm{m\,s^{-1}}$. Simple controllers were applied in order to keep the vehicle along the predefined trajectory and the aquaculture in the field of

Algorithm 4 Sigma Points Integration Process.

```
procedure SigmaPointsIntegration($\hat{\mathbf{x}}_0, \mathbf{P}_{\mathbf{x}_0}, \alpha, \beta, \lambda, n, \{\hat{\mathbf{x}}_k\}, \{\mathbf{P}_{\mathbf{x}_k}\}$)
Input: initial state mean $\hat{\mathbf{x}}_0$; initial state covariance $\mathbf{P}_{\mathbf{x}_0}$ , filter parameters $\alpha$, $\beta$ and $\lambda$; the dimensions of the state vector $n$.
Output: state and covariance estimates for each time instance $k$.
    Compute a set of $2n+1$ weights for the mean and covariance of sigma points:
        $$\begin{aligned} W_m^{(0)} &= \frac{\lambda}{n+\lambda} \\ W_c^{(0)} &= \frac{\lambda}{n+\lambda} + \left(1-\alpha^2+\beta\right) \\ W_m^{(i)} &= W_m^{(i)} = \frac{\lambda}{2\left(n+\lambda\right)}, \qquad i = 1, \cdots, 2n. \end{aligned} \tag{4.32}$$
    Calculate $2n+1$ sigma points:
        $$\begin{aligned} \mathscr{X}_0 = [&\hat{\mathbf{x}}_0, \\ &\hat{\mathbf{x}}_0 + \sqrt{(n+\lambda)\,\mathbf{P}_{\mathbf{x}_0}}, \\ &\hat{\mathbf{x}}_0 - \sqrt{(n+\lambda)\,\mathbf{P}_{\mathbf{x}_0}}\Big]. \end{aligned} \tag{4.33}$$
    Start the parallel sigma points' integration process SigmaPointsIntegration($\mathscr{X}_0, t_0$).
    Current time step $k = 1$.
    while the vehicle is activated do
        Get measurements $\mathbf{Z}_k^j \in \mathbb{R}^{n_z^j}$, where $n_z^j$ is the dimension of the measurement from the sensor combination $\mathscr{P}_w^j$.
        Get current values of the sigma points from the sigma points' integration process for the time instance $k$:
        $\mathscr{X}_{k|k-1} \leftarrow$ SigmaPointsIntegration($t_k$).
        Run the estimation procedure to get the estimates of the vehicle's state and covariance matrix for the current step:
        $(\hat{\mathbf{x}}_k, \mathbf{P}_{\mathbf{x}_k}) \leftarrow$ StateCovEstimation($\mathscr{X}_{k|k-1}, \mathbf{Z}_k^j$).
    end while
end procedure
```

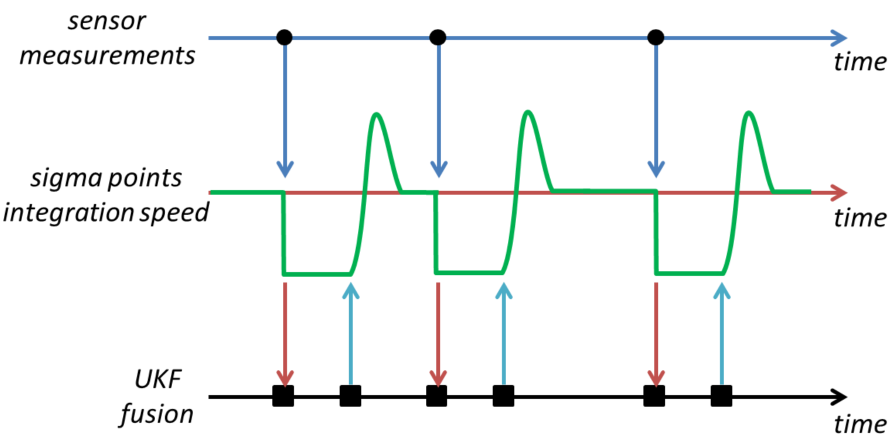


Figure 4.7: Illustration of the sigma points integration process.

view of the ROV. The STD of the gyro measurements was assumed as 0.05 rad/s, the STD of the accelerometers was 0.03 $\mathrm{m\,s^{-1}}$, the STD of the pressure sensor was 0.1 m, the STD of the LVS was 0.1 m. The results of one simulation run for 50 s are shown on 4.8, where the trace of the covariance matrix is demonstrated as a measure of convergence of the filter.

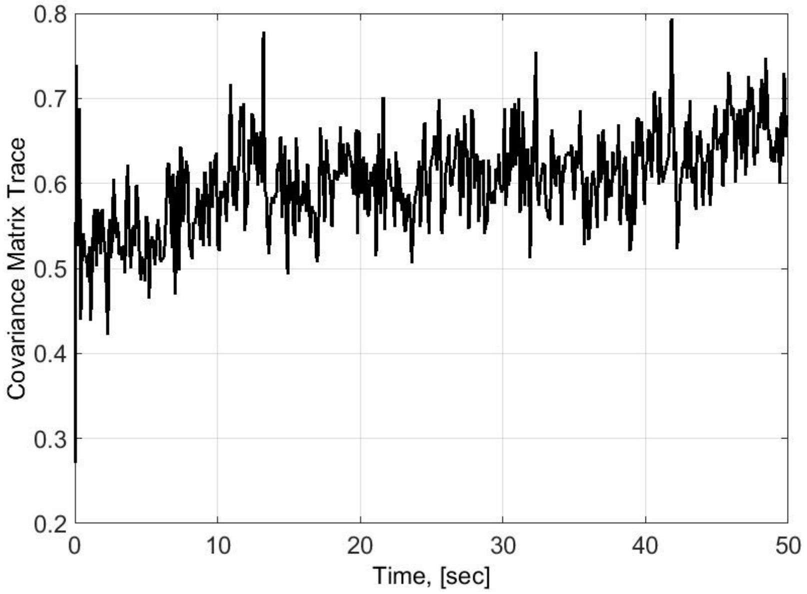


Figure 4.8: Simulation results for one run: the trace of the state estimation covariance matrix.

A series of Monte-Carlo runs were conducted in order to evaluate the performance of the proposed sensor fusion algorithm for the position estimation only. Since we know a "true" state of the vehicle at each time instance, two metrics were used to examine the filter, namely, the root mean squared error (RMSE), to estimate the accuracy of the filter, and the normalized estimation error squared (NEES), to provide a measure of the filter consistency [9].

For each Monte-Carlo run j, $j = 1, \ldots, N_{runs}$ and each time step i, $i = 1, \ldots, N_{steps}$ we computed the NEES as:

$$NEES_i^j = \left(\mathbf{x}_i^j - \tilde{\mathbf{x}}_i^j\right)^T \left(\mathbf{P}_x^{ij}\right)^{-1} \left(\mathbf{x}_i^j - \tilde{\mathbf{x}}_i^j\right). \tag{4.34}$$

The average NEES (ANEES) over 10 Monte-Carlo runs and for each time step i is shown on 4.9a for 100 s simulation run and was computed as:

$$ANEES_i = \frac{1}{N_{runs}} \sum_{j=1}^{N_{runs}} NEES_i^j. \tag{4.35}$$

The average of the NEES over all Monte-Carlo runs and all times steps was 11.16. For a consistent filter, the ANEES tends towards the dimension of the state vector as the number of Monte-Carlo runs approaches infinity [9]. The comprehensive study of the filter performance in terms of ANEES and RMSE for different types of filters was conducted in [53]. According to this study, the performance of the asynchronous UKF applied in this work is close to the performance of the standard UKF.

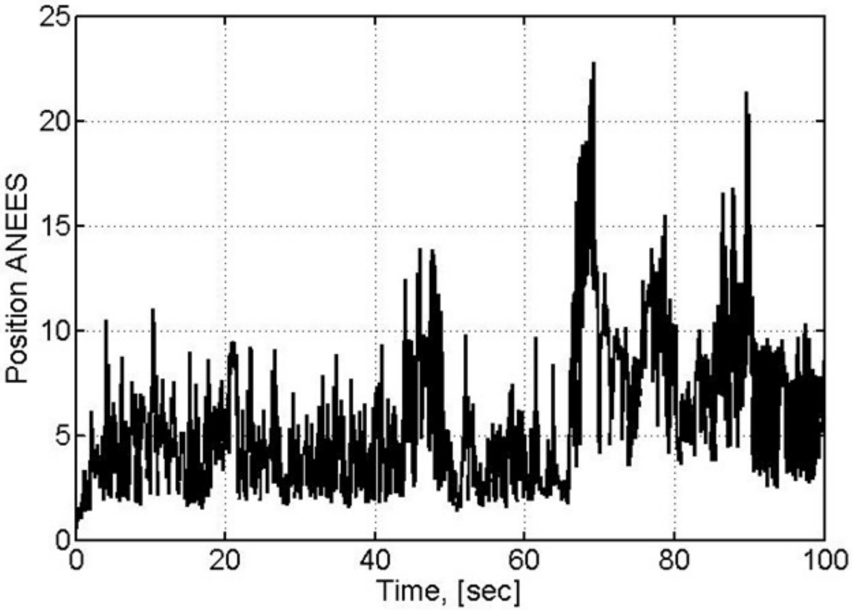
The RMSE shown on 4.9b was computed for each time as well according to:

$$RMSE_i = \sqrt{\frac{1}{N_{runs}} \sum_{j=1}^{N_{runs}} \left(\mathbf{x}_i^j - \tilde{\mathbf{x}}_i^j\right)^T \left(\mathbf{x}_i^j - \tilde{\mathbf{x}}_i^j\right)}, \tag{4.36}$$

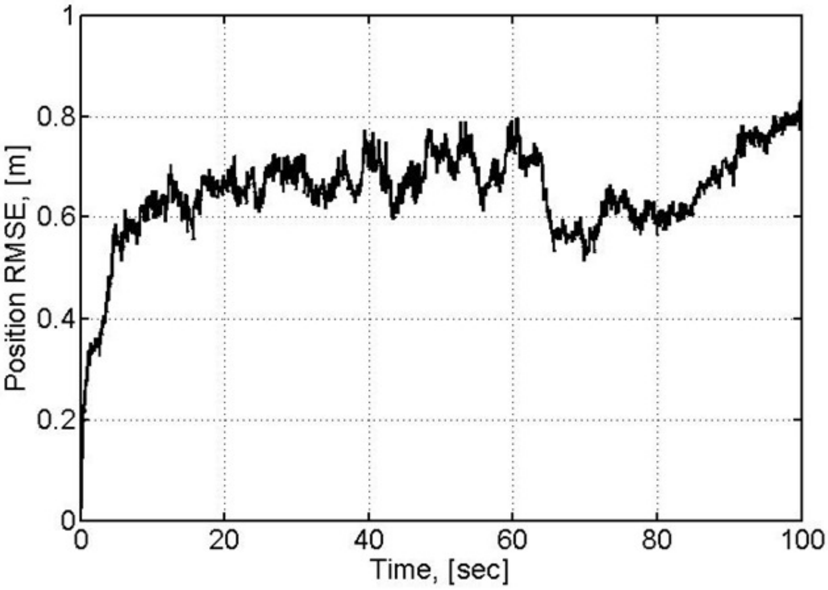
while the average RMSE over all steps for this case was 0.65 m.

4.6 Conclusion

This chapter presented a localization algorithm for an autonomous underwater vehicle for aquaculture inspection. The algorithm can fuse the information from several sensors providing measurements with different and various frequencies. The performance of the algorithm was validated through simulations under real-world conditions and noises.



(a) The position ANEES averaged per time step



(b) The position RMSE averaged per time step.

Figure 4.9: Simulation results for 10 Monte-Carlo runs.

Chapter 5

Trajectory Tracking Control of an Underwater Vehicle for Underwater Inspection Operations

This chapter presents the development of a hybrid controller for the fishnet coverage task by an autonomous underwater robotic system in the framework of the AQUABOT project [6][1]. The sensor fusion technique in the previous chapter for the robot localization uses information from several sources including an onboard inertial sensor, an onboard camera combined with line lasers and a priory knowledge about the aquaculture geometry. The control hybrid architecture includes an online estimator of the flow velocity around aquaculture provided by various sensors to compensate for the drift and necessary controllers required to track the 'virtual' moving target. The proposed algorithms are validated through simulations for small-scale aquaculture under realistic levels of noises and disturbances.

5.1 Introduction

Underwater robotic systems are used for a variety of applications in oceans and seas. Aquaculture and fisheries are agricultural sub-domains where application of robotic technologies is expected to significantly contribute to their future development and sustainability. In aquaculture industry, the main problem is considered to be escapes of fish from the fish net cages due to abrasion and tearing of mooring ropes, anchor chains, or holes in the fish nets from fish biting. In order to prevent such incidents, the regular inspection of all subsystems and early detection of net holes and other types of system faults are mandatory tasks in the fish producing industry.

---

[1] This work was supported by the European Regional Development Fund and the Republic of Cyprus through the Research Promotion Foundation under research grant ΑΕΙΦΟΡΙΑ/ΓΕΩΡΓΟ/0311(ΒΙΕ)/08.

The current work is a part of a research project co-funded by the European Regional Development Fund and the Republic of Cyprus and was aimed at developing an autonomous system for visual inspection of fish farm nets and moorings by using a tethered Remotely Operated Vehicle (ROV) in a shared autonomy mode. The GPS-denied undersea environments make the autonomous operation of underwater ROVs to be one of the most challenging problem. The effective and accurate localization and control is enabled by a fusion of data from onboard sensors and a-priory knowledge about dynamic behaviour of the aquaculture. The ocean currents and flows imply additional difficulties for a ROV control system making a robot to deviate away from its desired state or path. Estimates of the flow velocity provided by various sensors or techniques may be incorporated into the control loop to compensate for the drift.

In this chapter, we address a problem of underwater visual inspection task as a combination of: (i) a problem of localization and state estimation of the ROV with respect to the aquaculture by fusing information from different sources; (ii) a problem of control of an under-actuated underwater vehicle in the proximity to the fish net cages; (iii) a problem of full coverage of fishnet cages. Such problems have been studied intensively for different types of autonomous vehicles. Kottas and Roumeliotis in [63] investigated the observability properties of a navigation system based on camera and IMU sensors and proposed an algorithm utilizing this analysis to improve the Extended Kalman Filter (EKF) state estimator. Bloesh et al [11] considered a measurement fusion problem from an inertial sensor and a monocular camera in form of optical flow by using an UKF to estimate a robot pose over time. Karras et al [57] proposed an online asynchronous Modified Dual UKF algorithm for state and parameter estimation of an underwater vehicle that uses fused data from an IMU and a combined camera-point lasers system. Klebert et al. [61] proposed a comprehensive review of research works on flow hydrodynamics within and around a single net cage and a cage array.

The rest of the chapter is organized as follows: Section 5.2 presents an overview of the proposed robotic system and preliminaries, section 5.3 presents the design of the hybrid controller while section 5.4 provides the simulation results, and section 5.5 concludes the chapter.

5.2 System Overview

In this section, a brief description of a system proposed in the current project will be provided. The ROV used in the project is a VideoRay Pro IV ROV equipped with two horizontal thrusters for surge and yaw motions and one vertical thruster for heave motion. The vehicle is under-actuated, meaning that it cannot perform sway, roll and pitch motions. The basic sensor system includes a front facing camera housed with an acrylic dome, accelerometers, gyroscopes, a tilt-compensated compass and a depth sensor. In addition to these sensors, the ROV was equipped with three line lasers that being combined with the on-board camera concludes a novel Laser Vision System (LVS) shown on 5.1.

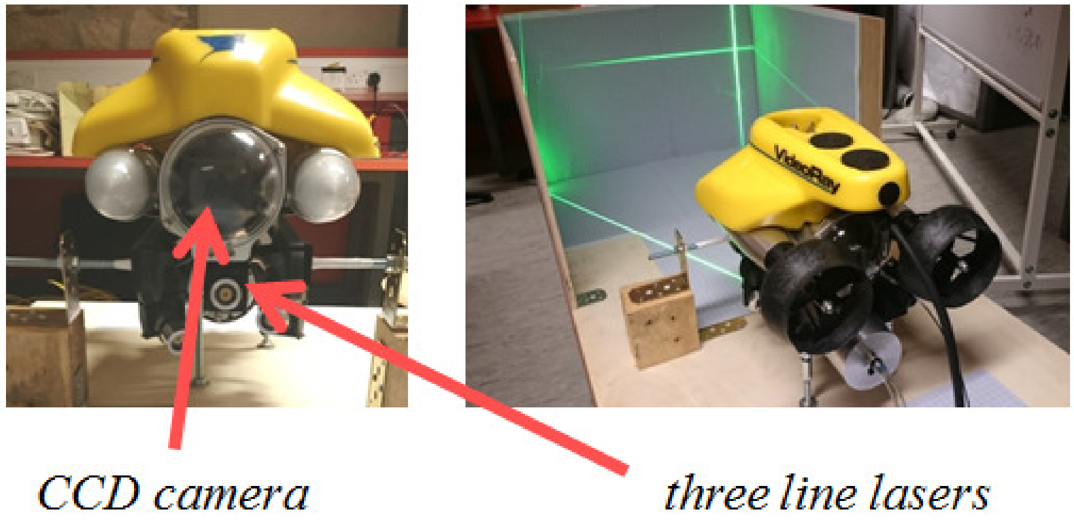


Figure 5.1: Combined Laser Vision System setup.

We consider the ROV as a 6-DOF rigid body that can be described by the following vectors: $\eta_1 = [x, y, z]^T$ is the position vector, $\eta_2 = [\phi, \theta, \psi]^T$ is the orientation (Euler angle) vector (roll, pitch and yaw angles), $\nu_1 = [u, v, w]^T$ is the body linear velocities (surge, sway and heave), $\nu_2 = [p, q, r]^T$ is the body angular velocities. Two coordinate systems are used to define the vectors: the body frame fixed to the ROV and the inertial frame fixed to the aquaculture (fish net cage) as shown on 5.2 along with the forces and moments $\tau = [X, Y, Z, K, M, N]^T$ acting on the vehicle.

5.2.0.1 Vehicle Model

In this work, we assume a full 6-DOF nonlinear dynamic model of the ROV that is given by the following system of equations [35]:

$$\begin{gathered} \mathbf{M}\dot{\nu} + \mathbf{C}(\nu)\,\nu + \mathbf{D}(\nu)\,\nu + \mathbf{g}(\eta) = \tau, \\ \dot{\eta} = \mathbf{J}(\eta)\,\nu, \end{gathered} \tag{5.1}$$

where $\mathbf{M}$ is the inertia matrix, $\mathbf{C}$ is the coriolis and centripetal matrix, $\mathbf{D}$ is the drag matrix, $\mathbf{g}$ is the force vector, τ is the trust vector, ν is the body velocities vector, η is the position and Euler angles vector, $\mathbf{J}$ is the transformation matrix from the body-fixed frame $\{B\}$ to the inertial frame $\{I\}$, where all aforementioned matrices are according to [35].

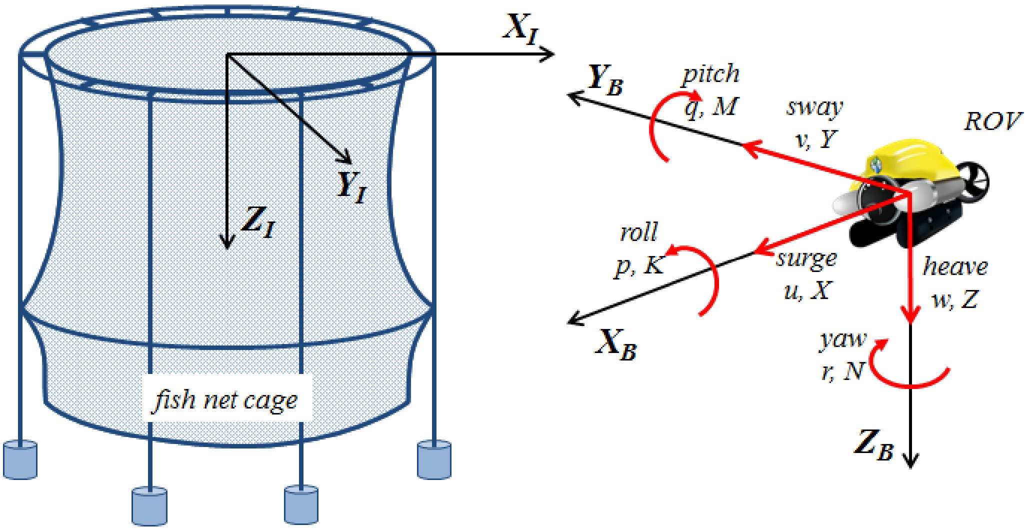


Figure 5.2: Inertial and body-fixed reference frames.

5.3 Hybrid Control Design

In previous chapters, 3 and 4 we presented algorithms for the LVS development and a sensor fusion technique for the underwater robot localization. In this section, we propose a hybrid controller for the coverage of the fishnet. We follow a two-step approach to design a controller for the underwater operation of the ROV: at the first step, a kinematic controller is designed, while at the second stage we backstep the kinematic controller into the system dynamics. Since the ROV used in this work is under-actuated, the controller consists of three independent controllers for each of actuated vehicle states: linear, angular and depth controllers. In addition, the proposed hybrid controller has four modes to ensure the sideward motion of the vehicle with camera pointing to the fish net cage.

To guarantee the full coverage of the aquaculture during the underwater visual inspection task, we assume that there is a 'virtual target' moving on the surface of the fishnet. The trajectory parameters of the moving target are included into the kinematic controller. Note that the navigation and coverage algorithms required for the target trajectory generation are beyond the scope of the current work. The target trajectory is assumed to be known before the underwater operation by the underwater robot.

5.3.1 Controller Design at the Kinematic Level

The kinematics of an underwater vehicle including virtual moving target terms can be described by the following system of equations:

$$
\begin{aligned}
\dot{d} &= \dot{x}_T \cos\theta + \dot{y}_T \sin\theta - u\cos\alpha + v\sin\alpha, \\
\dot{\theta} &= \frac{u\sin\alpha}{d} + \frac{v\cos\alpha}{d} - \frac{\dot{x}_T\sin\theta}{d} + \frac{\dot{y}_T\cos\theta}{d}, \\
\dot{\alpha} &= \dot{\theta} - \dot{\psi} = \frac{u\sin\alpha}{d} + \frac{v\cos\alpha}{d} - \frac{\dot{x}_T\sin\theta}{d} + \frac{\dot{y}_T\cos\theta}{d} - r,
\end{aligned}
\tag{5.2}
$$

where d is the norm of the planar vector **d** between the center of the body coordinate frame O_B and the center of the target frame O_T; θ is the angle between the vector **d** and X-axis of the target frame X_T; α s the angle between the vector **d** and the X-axis of the body frame X_B; x_T, y_T are the target velocity along the X- and Y-axes. Such 2D representation of the ROV's kinematics can be convenient since the motions in the vertical and horizontal directions are decoupled in this case. The frames, vectors and angles used here are shown on 5.3.

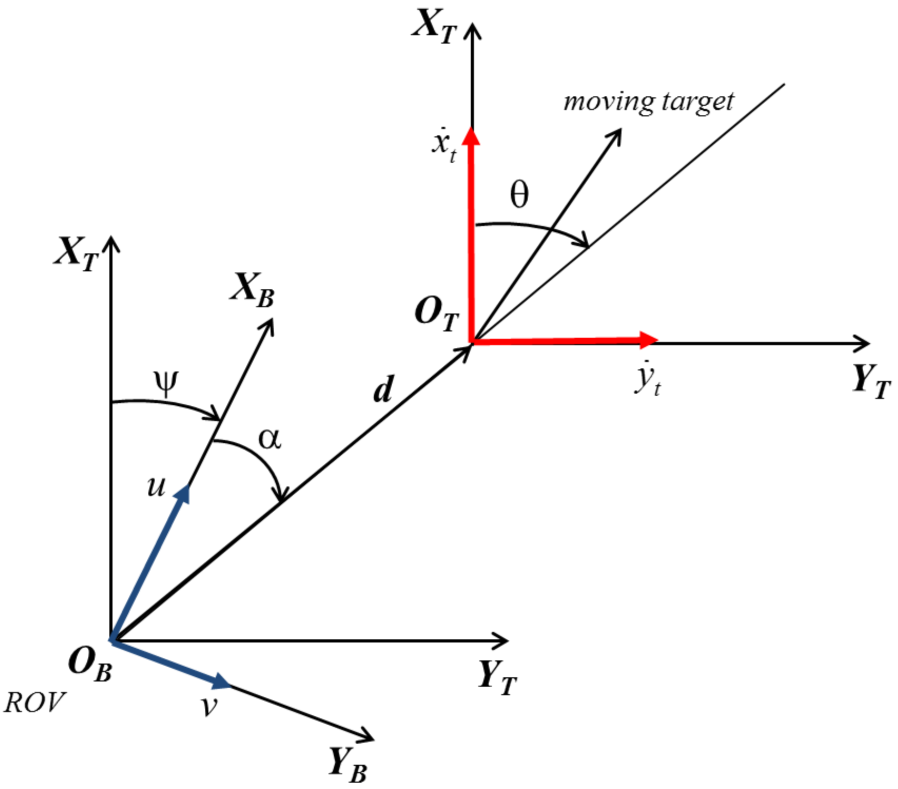


Figure 5.3: 2D representation of the vehicle kinematics in polar-like coordinate system.

5.3.1.1 Linear Controller

We assume that the vehicle operates in a specific region around the aquaculture bounded by the minimum d_{min} and the maximum d_{max} distances from the fishnet cage. These distances are defined by the requirements on the quality of underwater images from the onboard camera. For a certain desired distance d_{des} inside this region, the linear kinematic controller in the following form:

$$u_k = \frac{\gamma(d - d_{des}) + \dot{x_T}\cos\theta + \dot{y_T}\sin\theta + v\sin\alpha}{\cos\alpha} \tag{5.3}$$

is asymptotically stabilizes the vehicle at the distance d_{des} for a positive function γ and a positive constant α_0 as long as $|\alpha| < \alpha_0$.

Proof. Consider the candidate Lyapunov function

$$V_u = \frac{1}{2}(d - d_{des})^2, \tag{5.4}$$

its time derivative can be written as:

$$\dot{V_u} = (d - d_{des})\dot{d} = (d - d_{des})(\dot{x_T}\cos\theta + \dot{y_T}\sin\theta \\ -u\cos\alpha + v\sin\alpha).$$

Substituting control 5.4 into the time derivative of the Lyapunov function, we get:

$$\dot{V_u} = -\gamma(d - d_{des})^2 < 0,$$

meaning that the system 5.2 is asymptotically stable at d_{des} for a positive γ.

5.3.1.2 Angular Controller

The underwater vehicle should be pointed towards to the target during the inspection task in order to guarantee the continuous monitoring of the fishnet providing information through video. In addition, we have to take into account the constraint of the camera's field of view since the target should be within the camera's field of view all the time. Assuming the representation of the on-board camera and β is the angle between the normal to the fishnet surface at the target current position and the distance vector $\mathbf{d}$ (see Fig. 5.4), we have the following requirements:

$$\alpha || \beta,$$

$$-s_{FOV} \leq s \leq s_{FOV},$$

where $s = f\tan\alpha$ is the distance from the camera's central line to the projection of the target on the camera image plane; $s_{FOV} = f\tan\alpha_{FOV}$; α_{FOV} is the camera's field of view angle; f is the focal length of the camera.

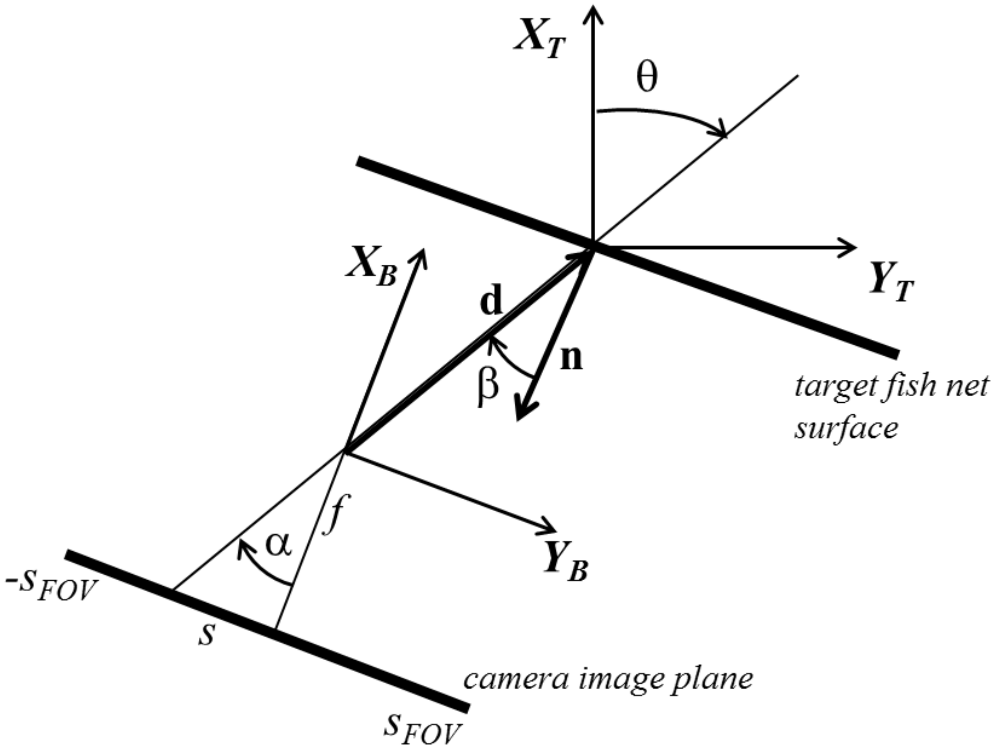


Figure 5.4: Target projection on the camera image plane.

Assuming the system 5.2, the angular control law in the form:

$$r_k = \frac{K_{rk}\left(s - s_{des}\right)}{f \sec^2 \alpha} + \dot{\theta} \tag{5.5}$$

stabilizes the target projection s at the desired distance s_{des}.

Proof. Consider the Lyapunov candidate function

$$V_r = \frac{1}{2}\left(s - s_{des}\right)^2$$

and its time derivative

$$\begin{aligned} \dot{V}_r &= \left(s - s_{des}\right)\dot{s} = \left(s - s_{des}\right) f \dot{\alpha} \sec^2 \alpha \\ &= \left(s - s_{des}\right) f \sec^2 \alpha \left(\dot{\theta} - r\right) \\ &= \left(s - s_{des}\right) f \sec^2 \alpha \left(\dot{\theta} - \frac{K_{rk}\left(s - s_{des}\right)}{f \sec^2 \alpha} - \dot{\theta}\right) \\ &= -K_{rk}\left(s - s_{des}\right)^2 < 0, \end{aligned}$$

we get that the controller 5.5 asymptotically stabilizes the system at s_{des}.

5.3.2 Controller Design at the Dynamic Level

In order to apply a backstepping procedure to design the linear and angular controllers at the dynamic levels, we linearize the dynamic equations of the vehicle 5.1 using feedback linearization:

$$\tau_{FL} = \mathbf{M}\dot{v} + \mathbf{C}v + \mathbf{D}v + \mathbf{g}, \tag{5.6}$$

where $\tau_{FL} = [Ux, 0, Uz, 0, 0, Ur]^T$.

5.3.2.1 Linear Controller

For the system 5.6 the controller defined as:

$$U_x = \dot{u_k} + \frac{K_{ud}}{2}(u_k - u) + (d - d_{des})\cos\alpha, \tag{5.7}$$

where u_k is the control law 5.4, stabilizes d at d_{des} asymptotically.

Proof. Consider the Lyapunov candidate function

$$V = V_u + \frac{1}{2}(u_k - u)^2$$

and its time derivative

$$\begin{aligned}\dot{V} &= \dot{V_u} + (u_k - u)(\dot{u_k} - \dot{u}) \\ &= (d - d_{des})\dot{d} + (u_k - u)(\dot{u_k} - \dot{u}).\end{aligned}$$

Substituting the kinematic controller 5.4 and the dynamic control law 5.7, we get:

$$\begin{aligned}\dot{V} &= -\gamma(d - d_{des})^2 - (u - u_k)\cos\alpha\,(d - d_{des}) \\ &\quad - \frac{K_{ud}}{2}(u - u_k)^2 + (d - d_{des})\cos\alpha\,(u - u_k) \\ &= -\gamma(d - d_{des})^2 - \frac{K_{ud}}{2}(u - u_k)^2 < 0.\end{aligned}$$

5.3.2.2 Angular Controller

For the system 5.6 the controller defined as:

$$U_r = \dot{r_k} - \frac{K_{rd}}{2}(r_k - r) - f\sec^2\alpha\,(s - s_{des}), \tag{5.8}$$

where r_k is the control law 5.5, stabilizes s at s_{des} asymptotically.

Proof. Consider the Lyapunov candidate function

$$V = V_r + \frac{1}{2}(r - r_k)^2$$

and its time derivative

$$\begin{aligned}\dot{V} &= \dot{V}_r + (r - r_k)(\dot{r} - \dot{r}_k) \\ &= f\dot{\alpha}\sec^2\alpha\,(s - s_{des})\,\dot{d} + (r - r_k)(\dot{r} - \dot{r}_k)\,.\end{aligned}$$

Substituting the kinematic controller 5.5 and the dynamic control law 5.8, we get:

$$\begin{aligned}\dot{V} = {} & f\left(\dot{\theta} - r\right)\sec^2\alpha\,(s - s_{des})^2 - \frac{K_{rd}}{2}(r - r_k)^2 \\ & - f\sec^2\alpha\,(s - s_{des})(r - r_k) = \\ & = f\left(\dot{\theta} - r + r_k - r_k\right)\sec^2\alpha\,(s - s_{des})^2 \\ & - \frac{K_{rd}}{2}(r - r_k)^2 - f\sec^2\alpha\,(s - s_{des})(r - r_k) = \\ & = -f\left(r - r_k + \frac{K_{rk}(s - s_{des})}{f\sec^2\alpha}\right)\sec^2\alpha\,(s - s_{des})^2 \\ & - \frac{K_{rd}}{2}(r - r_k)^2 - f\sec^2\alpha\,(s - s_{des})(r - r_k) = \\ & = -K_{rk}(s - s_{des})^2 - \frac{K_{rd}}{2}(r - r_k)^2 < 0.\end{aligned}$$

5.3.3 Depth Controller

To control the depth of the underwater vehicle, we use a PD controller since the heave motion of the vehicle is decoupled and has a dedicated vertical thruster. This controller stabilizes the vehicle at the desired depth z_{des}:

$$U_z = K_{pz}e_z + K_{dz}\dot{e}_z, \tag{5.9}$$

where K_{pz} and K_{dz} are the positive coefficients; $e_z = z - z_{des}$ is the vertical position error.

5.3.4 Drift Compensation

The ocean currents and flows impose additional complication in the control task for an underwater robot, since drift terms have to be included in the dynamic equations. The problem is even more difficult if the system is working in the proximity of underwater structures, since these structures locally modify the flow field. In order to feed-forward these effects in the control algorithm, we introduce an online estimator of the flow field around aquaculture and an additional level of control architecture.

The influence of irrotational ocean currents on the ROV's dynamics can be modeled as [36]:

$$\begin{aligned}& \mathbf{M}_{RB}\dot{\nu} + \mathbf{C}_{RB}(\nu)\,\nu + \mathbf{g}(\eta) + \ldots \\ & \quad + \mathbf{M}_A\dot{\nu}_r + \mathbf{C}_A(\nu_r)\,\nu_r + \mathbf{D}(\nu_r)\,\nu_r = \tau,\end{aligned} \tag{5.10}$$

where $\mathbf{M}_{RB}$ and $\mathbf{M}_A$ is the inertia matrix of the rigid body and added mass; $\mathbf{C}_{RB}$ and $\mathbf{C}_A$ is the coriolis-centripetal matrix for the rigid body and added mass; $\nu_r = \nu - \nu_c$ is the relative velocity vector in body fixed coordinates between the ROV velocity and the current velocity ν_c in the body frame.

The current velocity in body coordinates ν_c at the ROV's position η is calculated as a transformation of the current velocity in the global frame $\mathbf{v}_c^G$:

$$\nu_c = \left[\left(\mathbf{v}_c^b \right)^T, \omega_c^T \right]^T, \tag{5.11}$$

where $\mathbf{v}_c^b = \mathbf{R}_G^b (\eta_2) \mathbf{v}_c^G$ is the linear current velocity in body fixed coordinates and $\omega_c = 0$ is the rotational current velocity assumed to be zero due to irrotational flow assumption.

To model a flow around a fish net and calculate the current velocity $\mathbf{v}_c$ at an arbitrary ROV position η_1 in the environment, we apply a potential-based panel method for incompressible inviscid low-speed flows [60]. This method is based on solving the Laplace's equation for the total potential $\nabla^2 \Phi = 0$ by distributing "singularities" of unknown strength over the surface discretized into a number of elements (panels). The general solution to the potential equation may be found as a superposition of solutions for each panel along with boundary conditions.

For the fish net model, the "lumped-mass method" concept is used [88], according to which the net is divided into discrete net elements with lumped masses at the element nodes connected by massless springs. Each net element represents a structure of four interconnected bars that are subject to the external non-linear forces applied at the center of the element. The internal (structural) forces are incorporated into the model as well in order to address the flexibility of the net structure.

For the flow field modeling, the fish net cells are represented as three-dimensional quadrilateral source panels with constant strength. In this model, only the source elements are used since the fish net can be considered as a non-lifting symmetric surface with nonzero thickness. At each panel, a collocation point is specified at the centroid of the element and represents a point, where the Neumann boundary condition is applied at:

$$\nabla (\Phi + \Phi_\infty) \cdot \mathbf{n} = 0, \tag{5.12}$$

where Φ_∞ is the free-flow velocity potential.

Following that the velocity at each point can be obtained as a gradient of the velocity potential, the boundary condition required zero velocity normal to the net surface at each collocation point k may be written as:

$$(\mathbf{v}_k + \mathbf{U}_\infty) \cdot \mathbf{n}_k = 0, \tag{5.13}$$

where $\mathbf{v}_k = \{u, v, w\}_k$ is the velocity induced by all panels at the point k, $\mathbf{U}_\infty$ is the velocity of the flow (current), $\mathbf{n}_k$ is the normal to the net surface at the point k.

The velocity $\mathbf{v}_{kl}$ induced by a quadrilateral element l at the point $\mathbf{P}_k = \{x_k, y_k, z_k\}$ is calculated according to the expressions derived by Hess and Smith [54]:

$$\begin{aligned} u_{kl} = \frac{\sigma_l}{4\pi} \Bigg[& \frac{y_2 - y_1}{d_{12}} \ln \frac{r_1 + r_2 - d_{12}}{r_1 + r_2 + d_{12}} + \frac{y_3 - y_2}{d_{23}} \times \\ & \times \ln \frac{r_2 + r_3 - d_{23}}{r_2 + r_3 + d_{23}} + \frac{y_4 - y_3}{d_{34}} \ln \frac{r_3 + r_4 - d_{34}}{r_3 + r_4 + d_{34}} + \\ & + \frac{y_1 - y_4}{d_{41}} \ln \frac{r_4 + r_1 - d_{41}}{r_4 + r_1 + d_{41}} \Bigg], \end{aligned} \tag{5.14}$$

$$\begin{aligned} v_{kl} = \frac{\sigma_l}{4\pi} \Bigg[& \frac{x_2 - x_1}{d_{12}} \ln \frac{r_1 + r_2 - d_{12}}{r_1 + r_2 + d_{12}} + \frac{x_3 - x_2}{d_{23}} \times \\ & \times \ln \frac{r_2 + r_3 - d_{23}}{r_2 + r_3 + d_{23}} + \frac{x_4 - x_3}{d_{34}} \ln \frac{r_3 + r_4 - d_{34}}{r_3 + r_4 + d_{34}} + \\ & + \frac{x_1 - x_4}{d_{41}} \ln \frac{r_4 + r_1 - d_{41}}{r_4 + r_1 + d_{41}} \Bigg], \end{aligned} \tag{5.15}$$

$$\begin{aligned} w_{kl} = \frac{\sigma_l}{4\pi} \Bigg[& \tan^{-1} \frac{m_{12} e_1 - h_1}{z_k r_1} - \tan^{-1} \frac{m_{12} e_2 - h_2}{z_k r_2} + \\ & + \tan^{-1} \frac{m_{23} e_2 - h_2}{z_k r_2} - \tan^{-1} \frac{m_{23} e_3 - h_3}{z_k r_3} + \\ & + \tan^{-1} \frac{m_{34} e_3 - h_3}{z_k r_3} - \tan^{-1} \frac{m_{34} e_4 - h_4}{z_k r_4} + \\ & + \tan^{-1} \frac{m_{41} e_4 - h_4}{z_k r_4} - \tan^{-1} \frac{m_{41} e_1 - h_1}{z_k r_1} \Bigg], \end{aligned} \tag{5.16}$$

where

$$\begin{gathered} d_{ij} = \sqrt{\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2}, \\ m_{ij} = \frac{y_j - y_i}{x_j - x_i}, \\ r_i = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2 + z^2}, \\ e_i = (x_k - x_i)^2 + z^2, \\ h_i = (x_k - x_i)\,(y_k - y_i), \end{gathered}$$

i, j are the indices of the nodes of the quadrilateral panel ($i, j = 1, 2, 3, 4$), σ_l is the constant strength of the panel l.

The geometry of the panel are shown on 5.5. Note, that the expressions of the velocities $\{u, v, w\}_{kl}$ given by 5.14 are derived for the local coordinate system of the corresponding panel l. The origin of the local system is located at the center of each panel, the axes X and Y lay in the the plane of the panel, the axis Z is normal to the panel and points outwards.

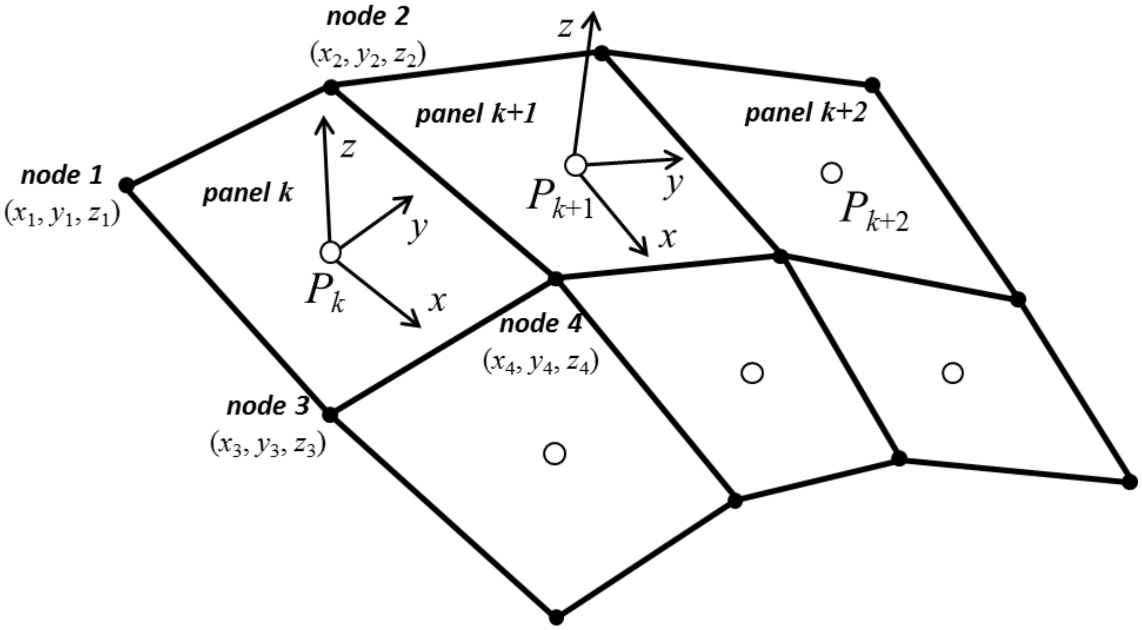


Figure 5.5: Geometry of quadrilateral source elements with constant strength.

The panel normal unit vector n_k, which is used for defining the local panel coordinate system and in 5.12, is found by taking the cross product of two unit vectors parallel to the diagonals of the element. This normal vector is used in the angular controller as well 5.5 in order to keep the vehicle aligned normally to the fish net.

In this work, several additional modifications were used in order to apply the described panel method for underwater aquacultures. Since the top of the net is laid in the plane of the water surface, the water cannot flow over it. This situation can be modeled by placing a “mirror” in the horizontal plane on the net top. The transformation of the panels will be easily the reflection of their coordinates and normal vectors around this plane.

The second issue that should be treated carefully for estimating the flow distribution around fish nets is penetration through the net cells. The Neumann boundary condition implies zero penetration velocity normal to the net surface. However, the fish net represents a mesh or porous medium. Thus, the penetration through the net cells inwards and outwards should be taken into account. We introduce a penetration coefficient K_p that describes a reduction value of the current velocity after the penetration or the percentage of the water that can pass through the net. For this case, the boundary condition can be written as follows:

$$(\mathbf{v} + \mathbf{U}_\infty) \cdot \mathbf{n} = K_p \mathbf{U}_\infty \cdot \mathbf{n}. \tag{5.17}$$

In this case, the current velocity $\mathbf{v}_c$ at an arbitrary ROV position η_1 in the environment is calculated as:

$$\mathbf{v}_c^G = \mathbf{U}_\infty + \sum_{i=0}^{N} \mathbf{v}_i\left(\eta_1\right), \tag{5.18}$$

where $\mathbf{v}_i$ is the velocity induced by a quadrilateral element i at the point η_1 calculated according to the expressions derived by Hess and Smith [54], N is the number of panels.

The drift compensation controller can be written as:

$$\tau_{drift} = \mathbf{M}_A \dot{\nu}_c + \mathbf{C}_A\left(\nu_c\right)\nu_c + \mathbf{D}\left(\nu_c\right)\nu_c. \tag{5.19}$$

5.3.5 Hybrid Control Architecture

As was discussed earlier, the underwater robot used in this project cannot perform sidewards motion due to the lack of an actuator in the sway direction. However, one of the main requirements of the underwater inspection with full coverage of an aquaculture is ability of the vehicle to move around the fish cage, i.e. to move sidewards while pointing towards the net. In order to achieve such performance, we introduce a hybrid control architecture shown on 5.6. The state transition scheme has four modes:

1. C_{Rmax}: In this mode, the angular controller with $s_{des} = s_{right}$ and the linear controller with $d_{des} = d_{max}$ are activated.

2. C_{Rmin}: In this mode, the angular controller with $s_{des} = s_{right}$ and the linear controller with $d_{des} = d_{min}$ are activated.

3. C_{Lmax}: In this mode, the angular controller with $s_{des} = s_{left}$ and the linear controller with $d_{des} = d_{max}$ are activated.

4. C_{Lmin}: In this mode, the angular controller with $s_{des} = s_{left}$ and the linear controller with $d_{des} = d_{min}$ are activated.

The transition between the modes are defined by the conditions on the current distance from the fish net d and the angle β between the current normal to the fish net surface and the vector $\mathbf{d}$ (see 5.4). The depth controller is activated independently of the linear and angular controller to keep the vehicle at the desired depth.

5.4 Simulation Results

The developed controller is validated through simulations under realistic levels of noises and disturbances. In these tests, aquaculture was assumed as a fish net of an open cylinder form which top is mounted to the fixed hoop. Sixteen weights with the mass of 400 g were

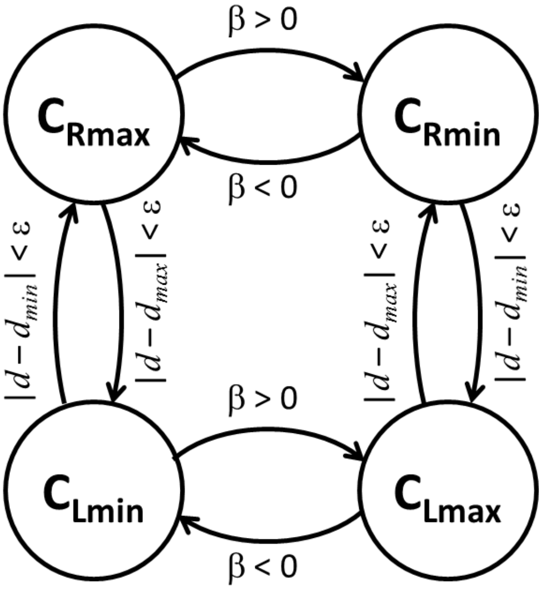


Figure 5.6: State transition diagram.

positioned uniformly along the circumference of the cylinder bottom. The fish net was divided 50 cells, while the diameter of the fish net cylinder was 2 m and the length (depth) was 2 m.

The simulation scenario was to navigate the vehicle sidewards around the fish net for 100 s. The STD of the gyro measurements was assumed as 0.05 rad/s, the STD of the accelerometers was $0.02\,\mathrm{m\,s^{-1}}$, the STD of the pressure sensor was 0.1 m, the STD of the LVS was 0.5 m for the distance measurements and 0.01 for the normal measurements. The resultant trajectory of one simulation run are shown on 5.7.

Additional simulations were conducted in order to check the drift compensation controller. The simulation scenario was to navigate the vehicle forward and downward up to the depth of 0.8 m. The results of this scenario are shown on 5.8, where the radial and vertical position errors are depicted on 5.8a and the vehicle coordinates are depicted on 5.8b. The position errors and vehicle coordinates are shown for two cases: case (a) corresponds to the case where the drift correction was applied (e.g. the ocean current velocity estimation at each position of the robot was used in the controllers), case (b) corresponds to the case without drift correction (e.g. the ocean current velocity was assumed equal to undisturbed flow velocity). We can see that applying the drift compensation controller allows the robot to navigate more accurate with position errors converging to zero.

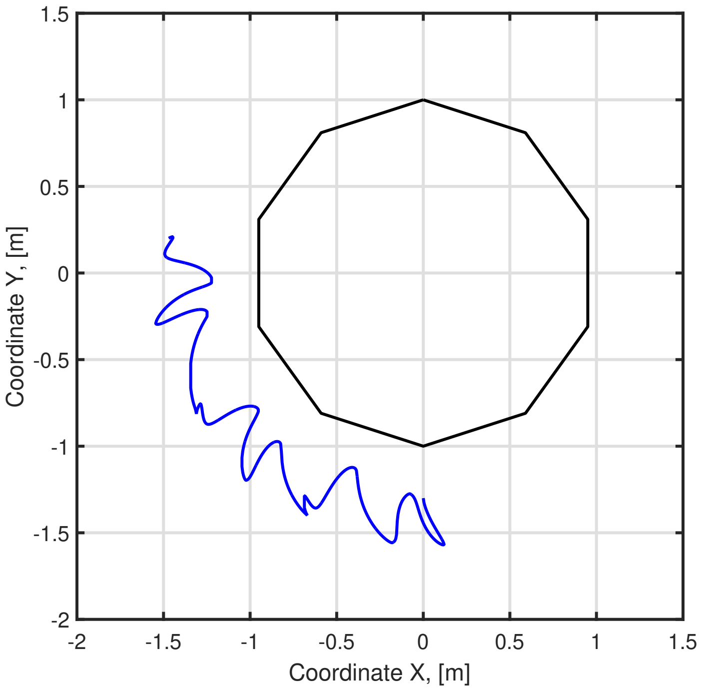
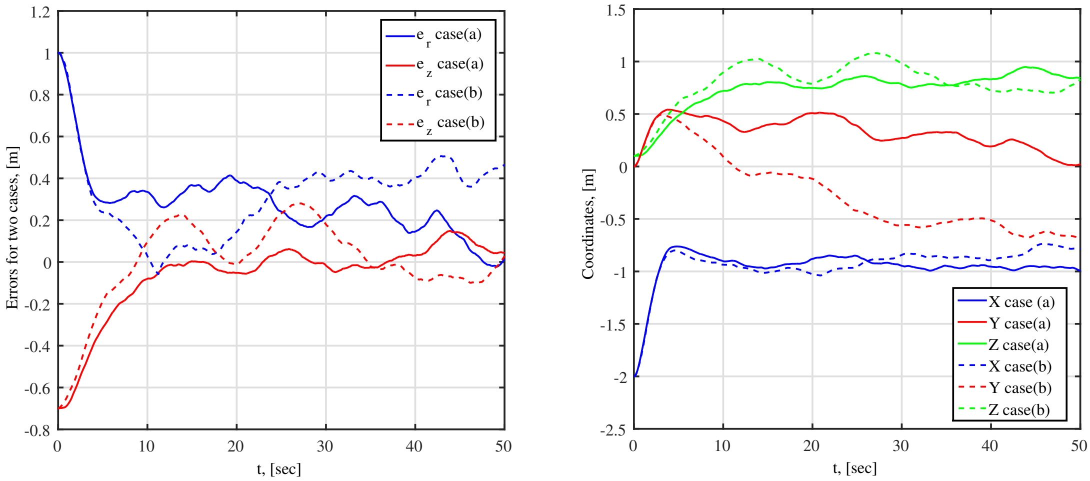


Figure 5.7: Simulation results for one run: the trajectory of the robot in X-Y plane.



(a) Position errors

(b) Vehicle position

Figure 5.8: Position errors and vehicle coordinates for two case: (a) with drift correction; (b) without drift correction.

5.5 Conclusion and Future Work

In this chapter, we presented a hybrid controller for underwater fishnet coverage task by an under-actuated underwater vehicle. It was shown that drift compensation level of the controller can improve the accuracy of the navigation task. This work includes intermediate results of a project aimed to develop an autonomous robotic system for fish farming inspection. The experimental tests in a dedicated mock-up environment and sea trials are part of our current effort.

Part IV

Multi-agent Systems

Chapter 6

Multi-Robot Coverage on Dendritic Topologies Under Communication Constraints

This chapter presents a novel algorithm for performing multi-robot coverage on networks with dendritic topology where the communication topology is location dependent and where the motion of each robot is constrained by the presence of the other robots in the network. The algorithm provides complete network coverage by the minimum number of robots, maintenance of communication constraints and robot collision avoidance. The minimum number of robots required for coverage is a by-product of the proposed algorithm. The efficiency of the algorithm is demonstrated through simulation studies. The results in this chapter were originally presented with the author contribution in [72][1].

6.1 Introduction

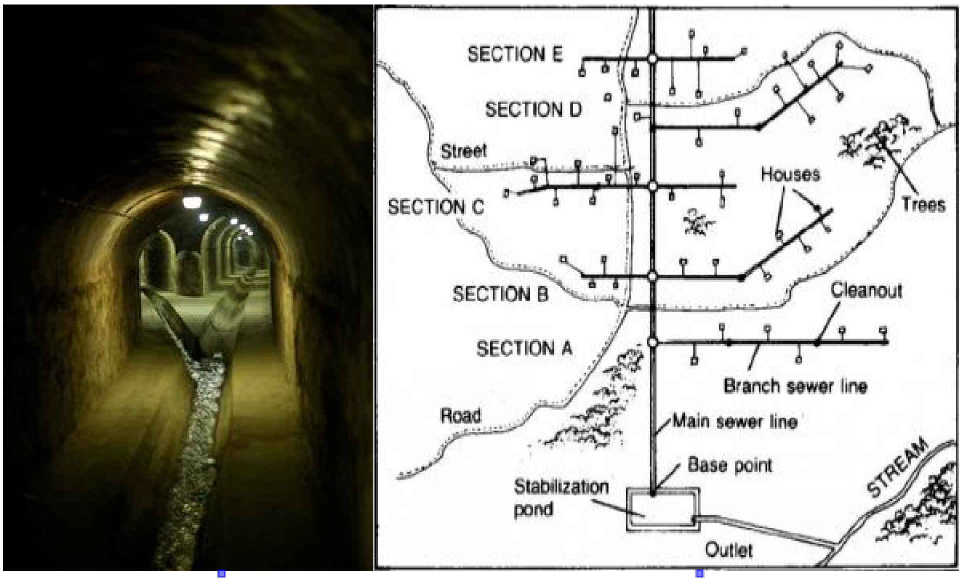
Multi-robot coverage is a topic that has received a lot of attention during the last decades. Several methodologies are surveyed in [19]. Distributed algorithms using Voronoi partitions and Lloyd's algorithm have been utilized in [27], [28] to tackle the problem. Solutions to the constrained coverage problem based on virtual potential fields were proposed in [85], [51]. In [39] the Spanning Tree Covering (STC) algorithm was proposed. In [10] the Least Recently Visited (LRV) algorithm has been presented to solve the problem of coverage, exploration and sensor deployment. In [50] an algorithm for unknown environment coverage utilizing information gathered by previously deployed nodes while simultaneously attaining line-of-sight communication was presented. Sensor-based coverage

---

[1] This work was supported under Grant Agreement 601116 EU/FP7/ECHORD++ Urban Robotics PDTI/ROBODILLOS.

algorithms for unknown environment were used in [92].

The main motivation for this work comes from the field of autonomous robotic coverage of large-scale under-ground sewer networks. An array of issues particular to the motivating problem differentiate this work from the existing literature and the proposed solution is the first to the authors' knowledge that tackles this problem. Sewer network systems (see Fig. 6.1) are typically dendritic networks converging in the downstream direction without closed loops. In network systems theory such networks are characterized as tree or more precisely directed tree networks where the directionality is inherited from the sewage flow direction. Sewer network flow channels are typically or restricted dimensions allowing only a single inspection robot at a given position. Robots operating in such networks can only interchange positions at channel junctions. Wireless communications in underground sewer networks are much more challenging than in above-ground settings. The main transmission path is through the underground network's channels, usually non-line-of-sight and with severe attenuation over corridor bends and turns and of course issues related to multi-path reflections. Hence, the communication coverage pattern for each robot differs fundamentally from the typical disk coverage pattern encountered in the multi-robot coverage literature and is strongly dependent on the local topology of the network at each robot's location.



(a) (b)

Figure 6.1: (a) Sewer flow channels at an Y-junction (b) Master sewer system map. Source: USAID (1982)

Utilizing the planar embedding of the underlying tree topology of the network, a new structure, the *topological buffer* is proposed that enables collision free multi-robot coordination. A wireless communication model that takes into account the local network topology enables the calculation of the communication range at each robot location. Finally, an appropriate abstraction of the multi-robot team as entities commuting through topological buffers embedded across the nodes of the network graph, enables the fulfillment of the coverage algorithm. The resulting algorithm guarantees complete network coverage with the minimum number of robots while maintaining communication between the robots and a base station and avoiding collisions between the robots. A side result of the algorithm is the minimum number of robots required for coverage.

The rest of the chapter is organized as follows: Section 6.2 presents preliminary notions and definitions while section 6.3 introduces the System Modeling. Section 6.4 presents the Coverage Algorithm and section 6.5 introduces the Simulation Results. Conclusions are provided in section 6.6.

6.2 Preliminaries

If K is a set then $|K|$ is the cardinality of the set. Let V denote a set of vertices and let $E \subseteq \{(v_1, v_2) \,|v_1, v_2 \in V\}$ denote a set of edges. Then a graph is the tuple:

$$G = \{V, E\}.$$

The degree of a vertex is the number of incident edges to the vertex. If $v \in V$ is a vertex then $N_G(v)$ is the set of adjacent vertices to v. A tree is a connected acyclic graph. A star graph of order n, S_n is a tree with n nodes, with one node having vertex degree $n-1$ and the rest having degree 1. Let $A = \{a_1, \dots a_i, \dots a_n\}$, where $n = |A|$, be an ordered set and denote with $A(k)$, $k \in \{1, \dots, |A|\}$ the k'th element of A. If A and B are ordered sets, then define their disjoint union as

$$C = A \biguplus B$$

where C is an ordered set where:

$$C(k) = \begin{cases} A(k), & k \in \{1, \dots, |A|\} \\ B(k), & k \in \{|A|+1, \dots, |A|+|B|\} \end{cases}$$

Define the prefix operator of a non-empty ordered set A as

$$Pre(A) \triangleq A(1)$$

and the postfix operator as

$$Post(A) \triangleq A(|A|).$$

Define the head operator of a non-empty ordered set A to be the ordered set

$$Head(A) \triangleq \{A(i)|\ i \in \{1,\ldots |A|-1\}\}$$

and the tail operator to be the ordered set

$$Tail(A) \triangleq \{A(i)|\ i \in \{2,\ldots |A|\}\}.$$

Definition 6 (Omni-directional buffer) *Assume a is an element. An omni-directional buffer is an ordered set B endowed with the following operations.*

- ${}^{\rightarrow}B[a]$*: push left operation transacted as* $B := \{a\} \uplus B$
- $B^{\leftarrow}[a]$*: push right operation transacted as* $B := B \uplus \{a\}$
- ${}^{\leftarrow}B$*: pop left operation transacted as* $B := Tail(B)$
- $B^{\rightarrow}$*: pop right operation transacted as* $B := Head(B)$

By restricting the operation of the omni-directional buffer we get the following:

Definition 7 (Directional buffer) *An omni-directional buffer restricted to operations ${}^{\rightarrow}B[a]$ and $B^{\rightarrow}$ is a left-to-right $(\rightarrow)$ buffer denoted as $\vec{B}$, whereas when restricted to operations $B^{\leftarrow}[a]$ and ${}^{\leftarrow}B$ is a right-to-left $(\leftarrow)$ buffer denoted as $\overset{\leftarrow}{B}$.*

We can now define the *topological buffer*:

Definition 8 (Topological Buffer) *A topological buffer T is an arrangement of a collection of omni-directional buffers $B_1,\ldots B_n$, each restricted to a direction $d_i \in \{\leftarrow,\rightarrow\}$, $i \in \{1,\ldots,n\}$, arranged on a star S_n topology. Left-to-right buffer operations are directed towards the central hub of the star topology whereas right-to-left operations are directed away from the hub. Assume a is an entity[2]. The topological buffer is endowed with the following operations:*

- ${}^{\rightarrow}T_i[a]$*: push left operation on $(\rightarrow)$ buffer $\vec{B}_i$ transacted as:* $\vec{B}{}_i^{\rightarrow}\ [a]$
- ${}^{\leftarrow}T_j$*: pop left operation on $(\leftarrow)$ buffer $\overset{\leftarrow}{B}_j$ transacted as:* ${}^{\leftarrow}\ \overset{\leftarrow}{B}_j$
- $T^{i\rightarrow j}$*: switch element from $\vec{B}_i$ to $\overset{\leftarrow}{B}_j$ buffer, transacted as the sequence:* $\overset{\leftarrow}{B}{}_j^{\leftarrow}\ [Post(\vec{B}_i)]$, $\vec{B}{}_i^{\rightarrow}$.

---

[2] i.e. a robot abstraction

- *$T[D]$: reset buffer directionality, where D is a set of directions and transacted as* $d_i := D(i),\ i \in \{1,\ldots,n\}$

Contents of buffer B_i of the topological buffer are addressed with $T(B_i)$.

Remark 1 *Operations on the buffers are considered to be blocking, in sense that concurrent operations are not allowed on the same buffer.*

Fig. 6.2 depicts a topological buffer on an S_3 topology.

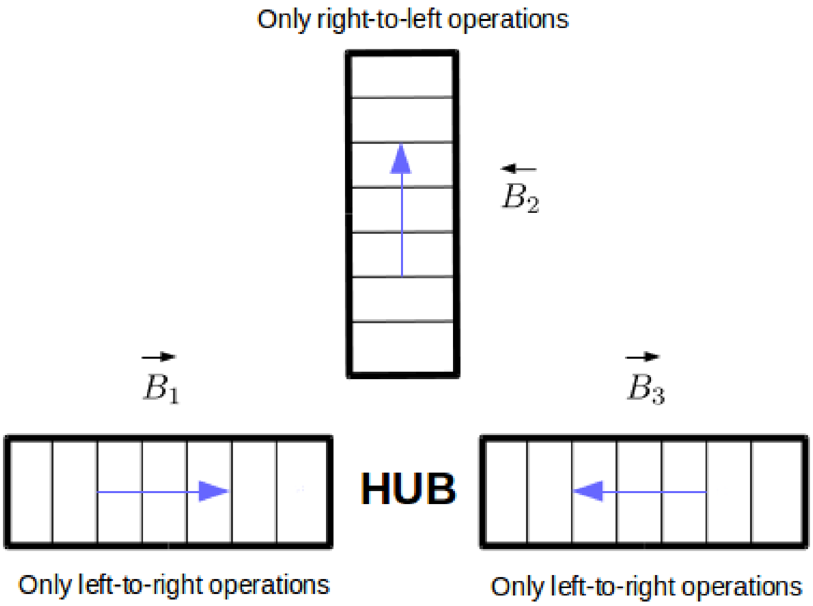


Figure 6.2: Topological buffer on an S_3 topology

6.3 System Modeling

6.3.1 Network Modeling

Since we are considering dendritic networks, a planar embedding of a rooted tree graph G is used (see e.g. Fig. 6.3). The vertices of the graph represent crossings and bends whereas graph edges represent the corridors of the underground network. At each vertex we install a topological buffer with S_k topology where k in the degree of the vertex. We denote by $v\{T\}$ the topological buffer at vertex v. Left-to-right buffer directionality is assigned to the incoming edge and right-to-left directionality is assigned to outgoing edges. The vertex set

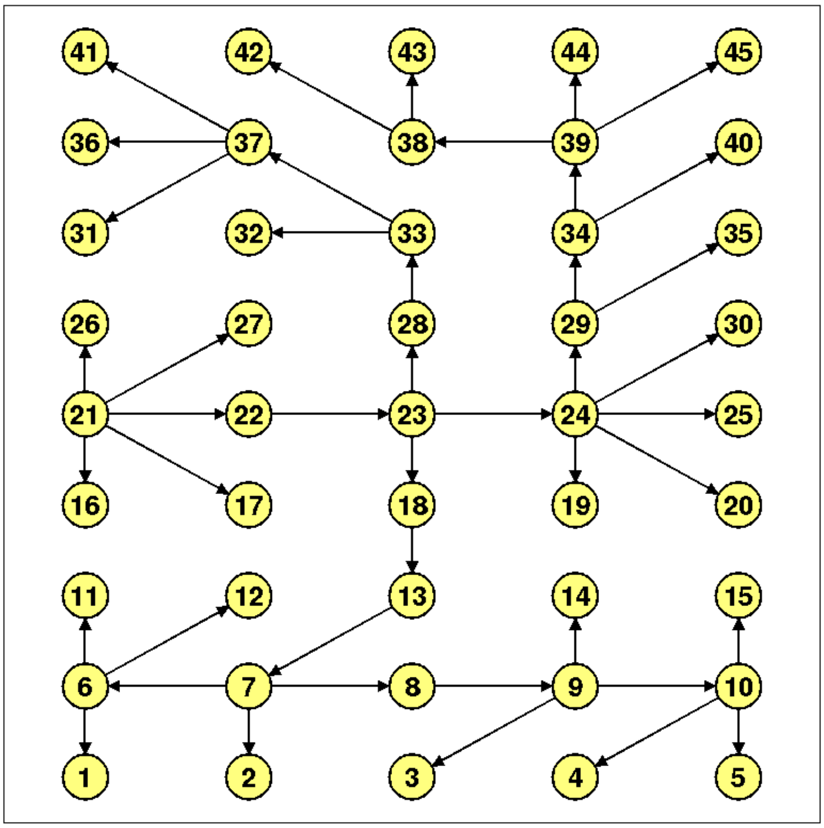


Figure 6.3: Network Model

of the graph is augmented with coordinate information for every vertex. For a vertex v, $p_x(v)$ is it's x-coordinate and $p_y(v)$ it's y-coordinate. Define the set:

$$V^* \triangleq \left\{(v, v_x, v_y) | v \in V,\ v_x = p_x(v),\ v_y = p_y(v)\right\}$$

Then the planar embedding of the graph is provided by:

$$G^* = \{V^*, E\}.$$

6.3.2 Robot Modeling

To capture robot motion in the sewer network an abstraction of the robot kinematics is assumed. Robot motion along sewer network channels represented by graph edges is abstracted as transitions of robot entities between neighboring topological buffers. A robot transition is a locking operation for the topological buffers. This enforces collision avoidance since no two robots can transverse the same edge in opposite directions at the same time. To this extend, due to the ordering enforced by the topological buffers, two robots cannot exchange positions in the same buffer. This abstract behavior is assumed to be executed by appropriate low level motion planners on each robot.

The following algorithm abstracts the robot model:

Algorithm 5 Robot Model

```
Require: Robot id $R_{id}$
Require: Robot position at $v_{init}$
Require: Robot destination at $v_{final} \in N_G(v)$
Ensure: New Robot location

if $R_{id} \in Pre(v_{init}\{T\}(\overleftarrow{B}_{v_{final}}))$ then
    $^{\rightarrow}v_{final}\{T\}\left[Pre(v_{init}\{T\}(\overleftarrow{B}_{v_{final}}))\right]$
    $^{\leftarrow}v_{init}\{T\}_{v_{final}}$
    return Success
else
    if $R_{id} \in Post(v_{init}\{T\}(\overrightarrow{B}_{v^*})),\ v^* \neq v_{final}$ then
        $T^{v^* \rightarrow v_{final}}$
        return Success
    end if
end if
return Failed
```

6.3.3 Communications Modeling

Assume that a transmitting agent a is placed at vertex v. Then the communication graph $G^*_{a,v}$ induced by the agent is a subgraph of G^*. In the proposed communications model, signal attenuation related to the geometrical arrangement of the network is being considered. Let $(v_i, v_j) \in E$ be the vertices of an incident edge $e_{i,j}$ of $G^*_{a,v}$. We consider a decay rate that depends on the angle between edges and is constant along the edge. The path loss (dB) along $e_{i,j}$ that depends on a and v, can be modeled as:

$$PL_{i,j,a,v} = -\lambda_{i,j,a,v} \left\| e_{i,j} \right\|$$

where $\left\| e_{i,j} \right\| = \left\| \begin{bmatrix} v_{x,i} \\ v_{y,i} \end{bmatrix} - \begin{bmatrix} v_{x,j} \\ v_{y,j} \end{bmatrix} \right\|$ and $\lambda_{i,j,a,v}$ the decay rate along $e_{i,j}$ for a source a placed at v. Now assume transmission along edge $e_{i,j}$ continuing along edge $e_{j,k}$. We assign a path loss on the $i \to j \to k$ connection, incurred at vertex v_j that is a function of the angle between the two edges:

$$PL_{i,j,k} = f_\lambda(arg(e_{i,j}, e_{j,k}))$$

where $arg(e_{i,j}, e_{j,k})$ is the angle between $e_{i,j}$ and $e_{j,k}$. Accurate calculation of λ and of function $f_\lambda(\cdot)$ is beyond the scope of this work.

Fig. 6.4 depicts the communication pattern for robots placed at nodes 38 and 14 calculated according to the communication model.

6.4 Coverage Algorithm

6.4.1 Algorithms

The coverage algorithm assumes that robots are identified and prioritized by their R_{id}. Since topological buffer operations by the robots are blocking operations the algorithm iteratively loops around R_{id}'s effectively permitting only a single robot operation at any time instance. During the initialization phase $G^*_{a,v_{root}}$ is calculated with cutoff signal loss at $-\infty$. This enables the determination of the *hardest* vertex to communicate to as the vertex with the minimum signal strength. A shortest path from the root to this vertex is calculated using Dijkstra's algorithm [30] and this path is used as the initial plan. During algorithm iterations, vertices identified as denoting the communication backbone for the current plan restrict their topological buffer operations to always have at least one robot in them. The communication backbone consists of robots whose communication subgraph union forms

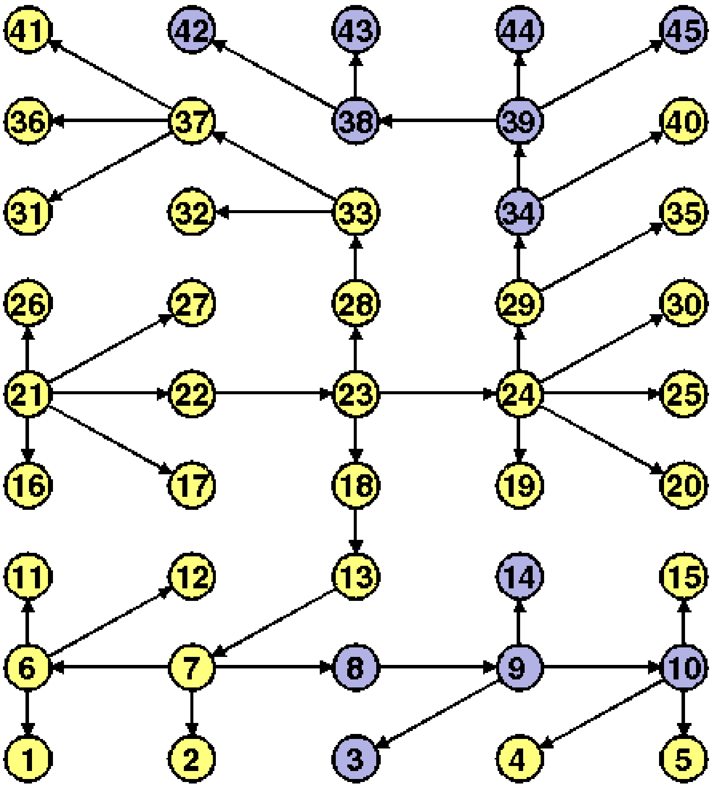


Figure 6.4: Communication patterns for $G^*_{a,38}$ and $G^*_{a,14}$

a connected component to the base station located at the root and whose per-pair intersections are at most a single vertex. The communication backbone is initialized by inserting the root vertex into it. The algorithm starts iterations after the first robot is inserted into the root buffer facing the second vertex in the current plan. The coverage algorithm is provided by 6 combined with 5 and 7. The following abbreviations are used in the presented algorithms: NPP: next plan position, PPP: previous plan position CBB: communication with backbone, cbb: communication backbone, Move Robot: Use Algorithm in 7, TB: Topological buffer, CL: current location, DSD: Determine shortest directed path from CL to non-covered leaf, NRP:Next Robot Position, RM: Robod Model: Use Algorithm in 5.

6.4.2 Analysis

Since the algorithm is an on-line algorithm, each *loop* R_{id} can be executed onboard the corresponding R_{id} robot. Thus, the complexity of the proposed algorithm is dictated by the DSD step that entails the determination of an uncovered reachable leaf from the current location and then the shortest directed path to it using Dijkstras algorithm hence having a worst case complexity of n^2 per iteration.

The algorithm provides by construction the collision avoidance property where in the context of this work a collision is interpreted as traversal of the same edge by two robots at the same time in opposite directions or position swapping on the same edge between two robots.

Assuming operation of a rooted tree network the algorithm is guaranteed to be deadlock free since each R_{id} loop is performed by locking all other robot's R_{id} loops. Moreover livelock situations at leaf nodes where plan reversal is taking place are guaranteed to never occur since only a singe robot is allowed to be at a leaf topological buffer (line 4 of Algorithm 6).

Communication of every robot with the base station is guaranteed by the communication backbone structure, where a robot can move to a new position only if backbone communication is available and backbone nodes are canceled only when becoming leaves.

Coverage is guaranteed since vertex nodes are abandoned only after becoming a leaf nodes (plan reversal only possible at a leaf node).

The resulting solution uses only the minimum number of robots since at each time all robots are involved in the same plan. This implies that no robots are being utilized for auxiliary plans that would reduce the robot availability for the main plan. Since robots are introduced to the system only when the limits of communication have been reached this implies that only the minimum required number of robots will be used.

Algorithm 6 Coverage Algorithm

Require: Initial plan *Plan*
Ensure: Network Coverage

```
while $G^*$ not covered do
    loop    $R_{id}$
        if Not at $Plan$ Leaf Node then
            if NPP is $Plan$ Leaf and populated  then
                return noProgress
            end if
            if CBB avaiable at NPP then
                Progress? ← Move Robot to NPP
                return Progress?
            else
                Add CL to cbb
                return Progress
            end if
        else
            if At backbone and Last in TB then
                Remove CL from cbb
            end if
            Reset CL TB direction to point out to PPP
            Reset PPP TB direction to point in from CL
            Progress? ← Move Robot to PPP
            if TB is empty then
                $Plan$ ← Head(Plan) $\uplus$ DSD
            end if
            return Progress?
        end if
    end loop
    if noProgress for all $R_{id}$ then
        Insert new robot in root buffer
    end if
end while
```

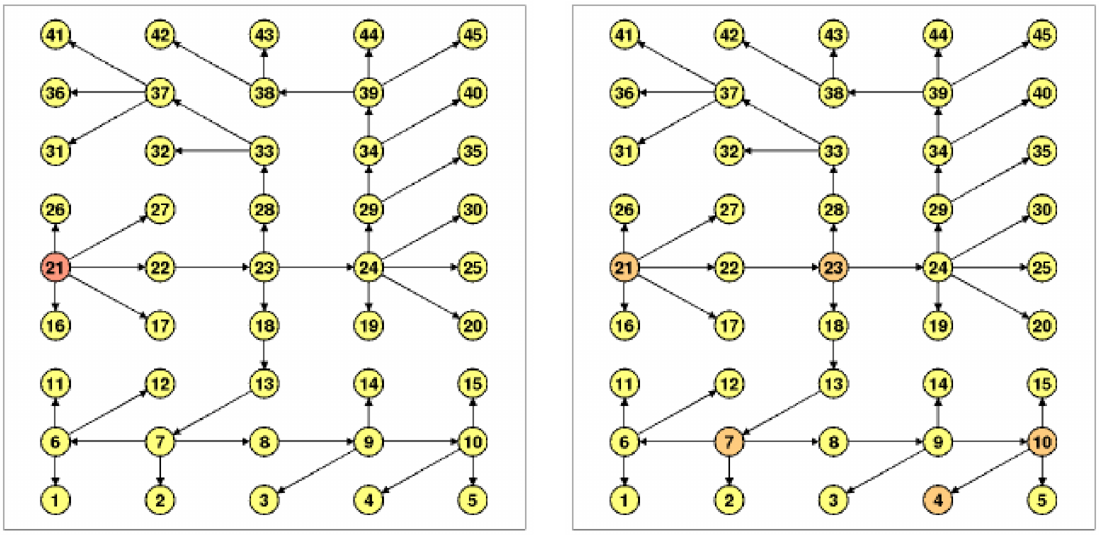
Algorithm 7 Move Robot Algorithm

```
Require: $R_{id}$, NRP
Ensure: Motion Succesful

if $R_{id}$ is in TB pointing towards NRP and at Prefix then
    if (CL NOT in cbb) OR (CL in cbb AND size(TB) $>$ 1) then
        Progress? $\leftarrow$ RM($R_{id}$, CL, NRP )
        return Progress?
    else
        return noProgress
    end if
else
    if $R_{id}$ is in TB at Postfix then
        Progress? $\leftarrow$ RM($R_{id}$, CL, NRP )
        return Progress?
    else
        return noProgress
    end if
end if
```

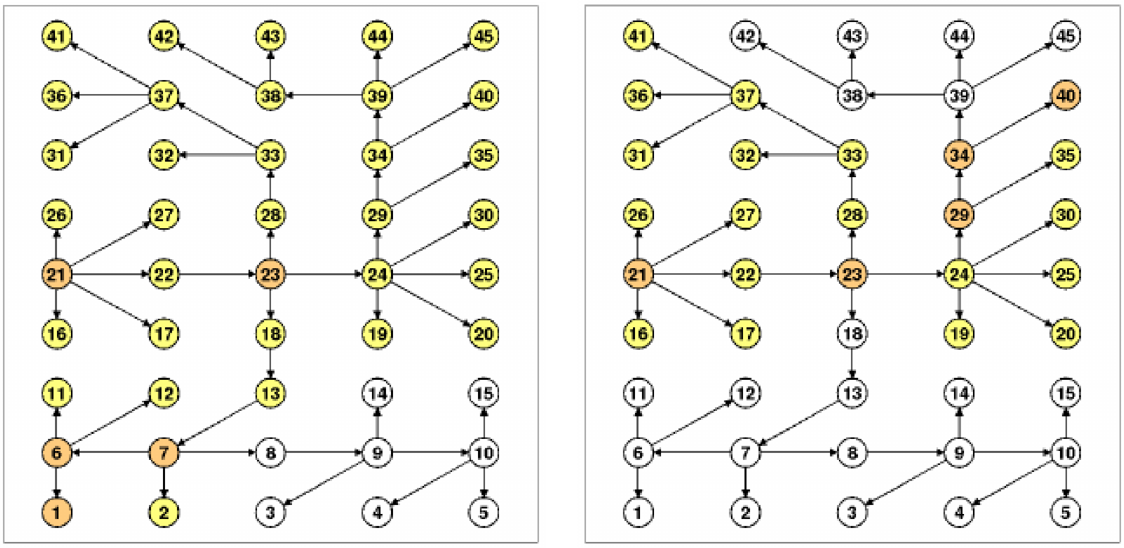
6.5 Simulations

In order to test and verify our algorithms, a set of simulations were performed using Matlab and the Octave - networks - toolbox [42]. A 5×9 square grid with a rooted tree was used as a network model (see Fig. 6.3).



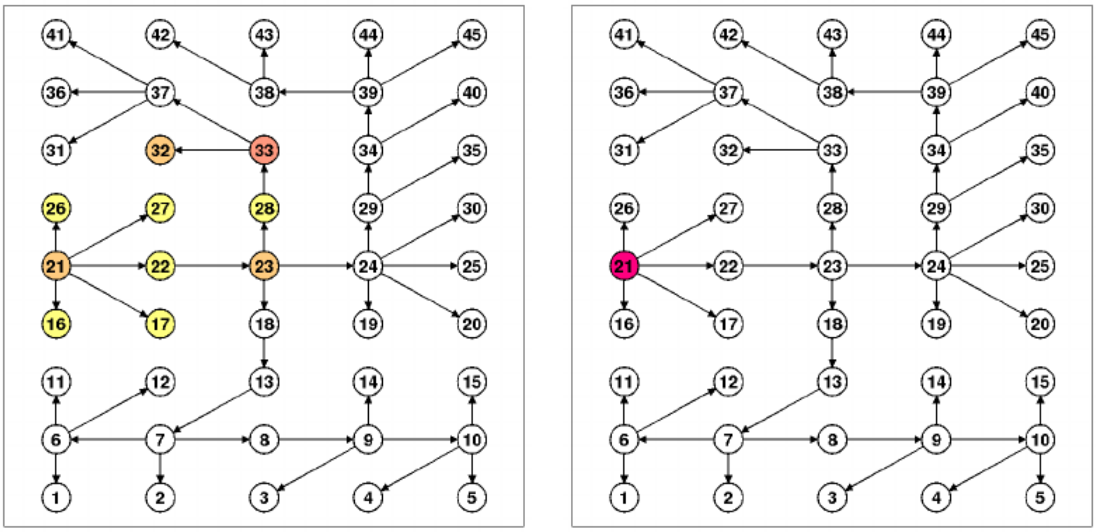
(a) Initial condition

(b) (Deployment to the hardest to communicate to vertex



(c) Coverage in progress

(d) Coverage in progress



(e) Multiple robots at vertex 33

(f) Coverage task completed successfully

Figure 6.5: Simulation results of the proposed algorithm

The algorithm was initialized and executed as described in section 6.4. The root was fixed at node 21. Fig. 6.5 illustrates the simulation results for our algorithms. The yellow vertices show the places that were not visited and the white vertices show the places that the robots have covered. The pink vertex represents that there are more than one robot in this buffer. Finally, orange vertices indicate the presence of a single robot in that buffer. As mentioned earlier, the starting point for operations is the root vertex node 21 (see Fig. 6.5(a)). Robots initially deploy up to the vertex with the lowest communication signal (Fig.

6.5(b)). Graphs (c) and (d) depict snapshots of the robots covering the network while commuting through topological buffers and maintaining communication. In graph (d) vertex 33 is pink to denote more than one robots in the buffer. Finally, graph (e) demonstrates that the robots have managed to perform the coverage task successfully.

6.6 Conclusions

In this chapter, we have proposed the development of a novel algorithm for multi-robot coverage on dendritic topologies under communication constraints. This is the first to the authors' knowledge solution to the specific problem. The resulting algorithm guarantees complete network coverage with the minimum number of robots while maintaining communication between the robots and a base station and avoiding collisions between the robots. The minimum number of required robots for coverage is a by-product of the algorithm, and a very useful information for planning ahead inspection operations. Non-trivial simulation results support the provided claims for the presented algorithm.

One of the major advantage of our algorithms is that they can tackle a coverage problem in urban restricted environments such as sewers and pipe networks with restricted dimensionality allowing only a single inspection robot at a given position.

Further research is directed towards optimizing the algorithm for time-optimal coverage and for energy optimal coverage. Moreover adaptation of the methodology for implementation on actual systems is something that we are looking into. Furthermore, a realistic simulation scenario was developed in ROS/Gazebo for testing new algorithms in simulated sewer network environment. Fig. 6.6 and Fig. 6.7 shows the robot while navigating in a sewer network environment.

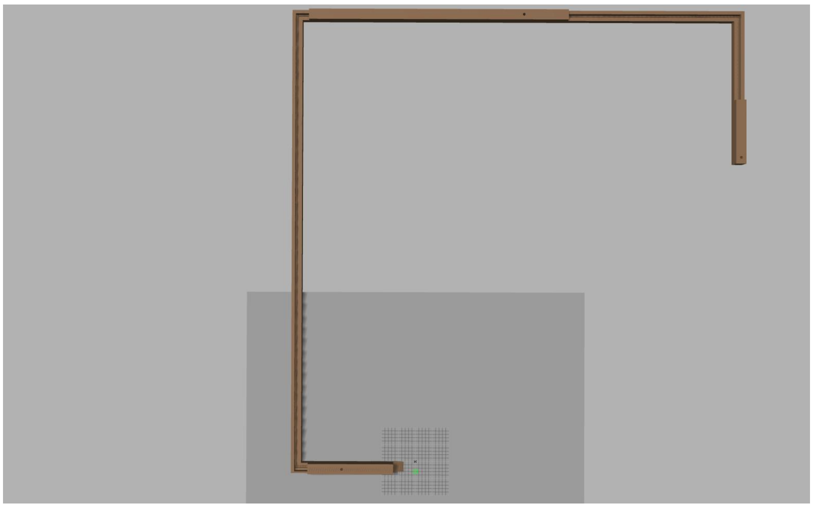


Figure 6.6: Sewer part in real dimensions.

Ros / Gazebo simulations are very important especially in environments where it is difficult to perform multiple tests to verify algorithms such as underground and underwater environments. The realistic simulations also help in testing algorithms that are intended for real robots avoiding unexpected consequences during experiments. The transition from the ROS / Gazebo simulator to the real one usually does not require changes since it can take into account all the possible real conditions.

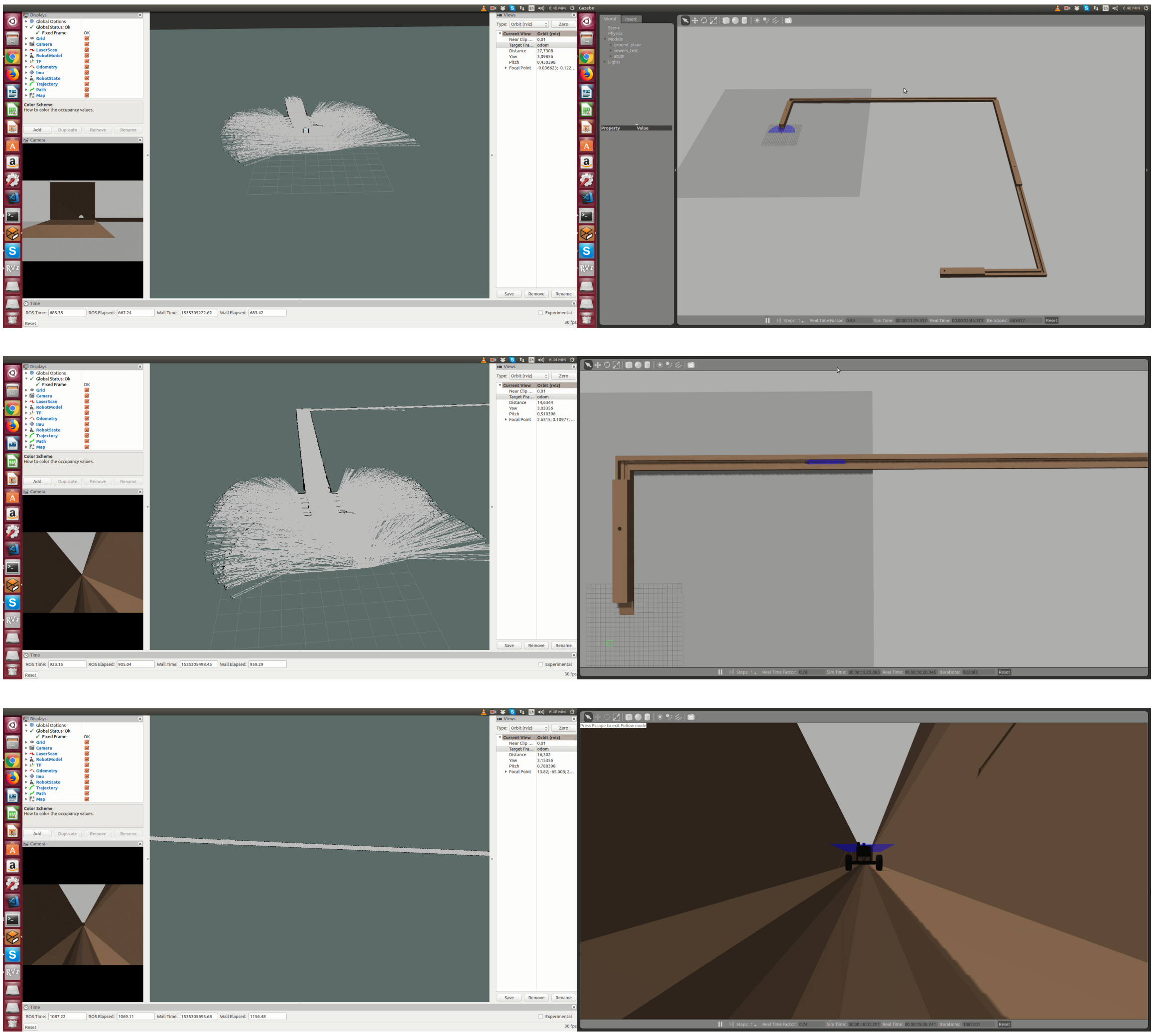


Figure 6.7: The left side of the figure shows the robot visualization in rviz environment while building the map and the right side of the figure shows the simulation scenario where the robot is operated in sewer network.

Part V

Closing Remarks

Chapter 7

Conclusions and Future Work

In this chapter, the conclusions of this dissertation are presented along with the future work for each chapter.

In chapter 2, we have presented a novel methodology for synthesizing motion tasks with real-time objectives from high-level temporal logic specifications. The main advantages of the presented methodology are correct by construction solution and the utilization of the Navigation Transformation by a fragment of LTL. This approach decouples the time abstracted navigation problem from the task sequencing problem, simplifying the solution. Furthermore, the proposed method is more computational efficient since the model checking technique is avoided. Further research issues include the addition of allowed and forbidden regions in the set of predicates, agents with input constraints, and real-world hardware experiments. Moreover, cooperative tasks with multiple agents are under investigation.

In chapter 3 we proposed a laser vision system for relative 3-D posture estimation of an underwater robot considering the hemispherical optics from the camera-dome housing. The analytical model that was developed is taking into account the non-linear hemispherical optics for the three medium refraction phenomenon air, acrylic dome, and water. The analytical model is accompanied with the software and hardware LVS for underwater ROV localization in close proximity to aquaculture fish-nets. The analytical model also provides the flexibility to calculate the refractive index where the autonomous system is operated, expanding its applicable range. The LVS is capable of generating information as point-cloud sets from each laser and, by utilizing the proposed algorithms, high-level information like distance and relative orientation of the target with respect to the ROV can be recovered. Another advantage of the proposed methodology is the automatic calibration technique that was developed in order to reduce the time which is needed for the regular maintenance required by the system (typical for underwater vehicles and their components). The spatial filter algorithm gives the advantage to the system to discriminate mesh-like targets from other artifacts in LVS measurements. The proposed system was tested and validated by experimental results both in the laboratory and field experiments at sea in aquaculture installations. The real-world experiments showed that the system is suitable for visual inspection operations in aquacultures. Furthermore, future works include the development of new algorithms based on the LVS for hull and port inspections, and coast mapping.

In chapter 4, we presented a state estimation algorithm for an autonomous underwater vehicle for the aquaculture inspection task. The algorithm fused all the required information from several sensors providing measurements with different and various frequencies. The performance of the algorithm was validated through simulation results under real-world conditions and noises. As future work, we consider conducting experiments and field tests to verify the proposed models and algorithms. Using other available sensors and information (for example, optical flow) in the state estimator along with applying observability constraints analysis to the filter is considered to be a part of the future work as well.

Chapter 5 a hybrid controller was proposed for the fishnet coverage task for visual inspection operations. The hybrid control architecture utilizes information from different sources, including on-line adaptive prediction of the fishnet geometry and estimation of the flow velocity in the proximity to aquacultures. The drift compensation controller has improved the accuracy of the navigation task. This work includes intermediate results of a project aimed to develop an autonomous robotic system for aquaculture inspection. The experimental tests in a dedicated mock-up environment and sea trials are part of the future implementation of the proposed algorithms.

In chapter 6 we have proposed the development of a novel algorithm for multi-robot coverage on dendritic topologies under communication constraints. This is the first to the authors' knowledge solution to the specific problem. The resulting algorithm guarantees complete network coverage with the minimum number of robots while maintaining communication between the robots and a base station and avoiding collisions between the robots. The minimum number of required robots for coverage is a by-product of the algorithm and very useful information for planning ahead inspection operations. Non-trivial simulation results support the provided claims for the presented algorithm. One of the major advantages of the proposed algorithms is that they can tackle a coverage problem in urban restricted environments such as sewers and pipe networks with restricted dimensionality allowing only a single inspection robot at a given position. Further research is directed towards optimizing the algorithm for time-optimal coverage and for energy optimal coverage. Moreover, an adaptation of the methodology for the implementation of actual systems is something that we are looking into.

Appendix A

Notations

A.1 Linear Temporal Logic

The syntactic rules according to which formula in LTL can be constructed. Temporal logic is a class of logic that extends propositional or predicate logic with temporal properties [8]. The basic ingredients of LTL-formula are atomic propositions (state labels $a \in P$), the Boolean connectors like conjunction $\wedge$, and negation $\neg$, and two basic temporal modalities $\bigcirc$ ("next") and $\mathcal{U}$ ("until").

Definition 9 *Syntax of LTL [8]*
LTL formula over a set P of atomic proposition are formed according to the following grammar:

$$\varphi \ ::= \ true \mid a \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc \ \varphi \mid \varphi_1 \mathcal{U} \varphi_2$$

where $a \in P$.

From the basic LTL operators we can derive additional standard Boolean operators. More specific the until operator allows to derive the temporal modalities $\Diamond$ ("eventually", sometimes in the future) and $\Box$ ("always", from now on forever) as follows:

$$\Diamond\phi = \mathbf{true}\mathcal{U}\phi \qquad \Box\phi = \neg\Diamond\neg\phi$$

In addition, we can derive other temporal operators such as "disjunction" $\vee$ and "implication" $\Rightarrow$.

Definition 10 *Semantics of LTL [8]*
Let φ be an LTL formula over P. The LT property by φ is

$$Words(\varphi) = \left\{\sigma \in (2^P)^\omega \mid \sigma \ \models \varphi\right\}$$

where the satisfaction relation $\models \subseteq (2^P)^\omega \times$ LTL is the smallest relation with the properties below.

$$\begin{array}{lll}
\sigma \models true & & \\
\sigma \models a & \text{iff} & a \in A_0 \ (i.e., A_0 \models a) \\
\sigma \models \varphi_1 \wedge \varphi_2 & \text{iff} & \sigma \models \varphi_1 \text{ and } \sigma \models \varphi_2 \\
\sigma \models \neg\varphi & \text{iff} & \sigma \nvDash \varphi \\
\sigma \models \bigcirc\varphi & \text{iff} & \sigma[1\ldots] = A_1A_2A_3\ldots \models \varphi \\
\sigma \models \varphi_1 \mathcal{U} \varphi_2 & \text{iff} & \exists j \geq 0. \sigma[j\ldots] \models \varphi_2 \text{ and } \sigma[i\ldots] \models \varphi_1, \text{ for all } 0 \leq i \leq j
\end{array}$$

Here, for $\sigma = A_0A_1A_2\ldots \in (2^P)^\omega, \sigma[j\ldots] = A_jA_{j+1}A_{j+2\ldots}$ is the suffix of σ starting in the $(j+1)$st symbol A_j.

Bibliography

[1] J.V. Aarsnes, H. Rudi, and G. Loland. Current Forces on Cage, Net Deflection. *Engineering for Offshore Fish Farming*, pages 137–152, 1990.

[2] Jose Joaquin Acevedo, Begoña C Arrue, Ivan Maza, and Aníbal Ollero. Cooperative perimeter surveillance with a team of mobile robots under communication constraints. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5067–5072. IEEE, 2013.

[3] Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A Henzinger, P-H Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. The algorithmic analysis of hybrid systems. *Theoretical computer science*, 138(1):3–34, 1995.

[4] Rajeev Alur, Tomás Feder, and Thomas A Henzinger. The benefits of relaxing punctuality. *Journal of the ACM (JACM)*, 43(1):116–146, 1996.

[5] Gianluca Antonelli, Thor I Fossen, and Dana R Yoerger. Underwater robotics. *Springer handbook of robotics*, pages 987–1008, 2008.

[6] AQUABOT. www.aquaculturebot.com, visited 2018. 2018.

[7] Ronald C Arkin and Jonathan Diaz. Line-of-sight constrained exploration for reactive multiagent robotic teams. In *7th International Workshop on Advanced Motion Control. Proceedings (Cat. No. 02TH8623)*, pages 455–461. IEEE, 2002.

[8] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.

[9] Yakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. New York: Wiley, 2001.

[10] Maxim A Batalin and Gaurav S Sukhatme. The analysis of an efficient algorithm for robot coverage and exploration based on sensor network deployment. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3478–3485. IEEE, 2005.

[11] Michael Bloesch, Sammy Omari, Peter Fankhauser, Hannes Sommer, Christian Gehring, Jemin Hwangbo, Mark A.Hoepflinger, Marco Hutter, and Roland Yves Siegwart. Fusion of Optical Flow and Inertial Measurements for Robust Egomotion Estimation. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[12] Robert Bogue. Underwater robots: a review of technologies and applications. *Industrial Robot: An International Journal*, 42(3):186–191, 2015.

[13] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[14] Thomas Brihaye, Morgane Estiévenart, and Gilles Geeraerts. On mitl and alternating timed automata. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 47–61. Springer, 2013.

[15] Charles Cain and Alexander Leonessa. Laser based rangefinder for underwater applications. In *American Control Conference (ACC), 2012*, pages 6190–6195. IEEE, 2012.

[16] Christos G Cassandras and Stephane Lafortune. *Introduction to discrete event systems*. Springer Science & Business Media, 2009.

[17] Andrew D. Chambers, Sebastian Scherer, Luke Yoder, Sezal Jain, Stephen T. Nuske, and Sanjiv Singh. Robust Multi-Sensor Fusion for Micro Aerial Vehicle Navigation in GPS-Degraded/Denied Environments. In *American Control Conference (ACC)*, June 2014.

[18] Young-Il Choo and Mario J. Casarella. Hydrodynamic Resistance of Towed Cables. *Journal of Hydronautics*, 5(4):126–131, 1971.

[19] Howie Choset. Coverage for robotics–a survey of recent results. *Annals of mathematics and artificial intelligence*, 31(1-4):113–126, 2001.

[20] Howie M Choset, Seth Hutchinson, Kevin M Lynch, George Kantor, Wolfram Burgard, Lydia E Kavraki, and Sebastian Thrun. *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[21] Edmund M Clarke and E Allen Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Workshop on Logic of Programs*, pages 52–71. Springer, 1981.

[22] Christos Constantinou, Savvas Loizou, and George Georgiades. A laser vision system for relative 3-d posture estimation of an underwater vehicle with hemispherical optics. under revision.

[23] Christos Constantinou, Svetlana Potyagaylo, Savvas Loizou, and George Georgiades. Towards multi-sensor ukf-based localization of an underwater robotic vehicle for underwater inspection operations. under submission.

[24] Christos C Constantinou and Savvas G Loizou. Automatic controller synthesis of motion-tasks with real-time objectives. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 403–408. IEEE, 2018.

[25] Christos C Constantinou, Savvas G Loizou, and George P Georgiades. An underwater laser vision system for relative 3-d posture estimation to mesh-like targets. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 2036–2041. IEEE, 2016.

[26] Christos C Constantinou, Savvas G Loizou, George P Georgiades, Svetlana Potyagaylo, and Dimitrios Skarlatos. Adaptive calibration of an underwater robot vision system based on hemispherical optics. In *Autonomous Underwater Vehicles (AUV), 2014 IEEE/OES*, pages 1–5. IEEE, 2014.

[27] Jorge Cortés and Francesco Bullo. Coordination and geometric optimization via distributed dynamical systems. *SIAM Journal on Control and Optimization*, 44(5):1543–1574, 2005.

[28] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1327–1332. IEEE, 2002.

[29] Witold Czajewski and Andrzej Sluzek. Development of a laser-based vision system for an underwater vehicle. In *Industrial Electronics, 1999. ISIE'99. Proceedings of the IEEE International Symposium on*, volume 1, pages 173–177. IEEE, 1999.

[30] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik 1*, pages 269–271, 1959.

[31] Jindrich Dunik, Miroslav Simandl, and Ondrej Straka. Unscented Kalman Filter: Aspects and Adaptive Setting of Scaling Parameter. *IEEE Transactions on Automatic Control*, 57(9):2411–2416, 2012.

[32] Alexandre Duret-Lutz, Alexandre Lewkowicz, Amaury Fauchille, Thibaud Michaud, Etienne Renault, and Laurent Xu. Spot 2.0—a framework for ltl and\omega-automata manipulation. In *International Symposium on Automated Technology for Verification and Analysis*, pages 122–129. Springer, 2016.

[33] Georgios E Fainekos, Hadas Kress-Gazit, and George J Pappas. Hybrid controllers for path planning: A temporal logic approach. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 4885–4890. IEEE, 2005.

[34] Georgios E Fainekos, Hadas Kress-Gazit, and George J Pappas. Temporal logic motion planning for mobile robots. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2020–2025. IEEE, 2005.

[35] Thor I. Fossen. *Guidance and Control of Ocean Vehicles*. New York: Wiley, 1994.

[36] Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[37] Arne Fredheim. *Current Forces on Net Structures*. PhD thesis, Norwegian University of Science and Technology, 2005.

[38] David W. Fredriksson, M. Robinson Swift, James D. Irish, Igor Tsukrov, and Barbaros Celikkol. Fish Cage and Mooring System Dynamics Using Physical and Numerical Models with Field Measurements. *Ocean Engineering*, 30:251–270, 2003.

[39] Yoav Gabriely and Elon Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):77–98, 2001.

[40] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous systems*, 61(12):1258–1276, 2013.

[41] Paul Gastin and Denis Oddoux. Fast LTL to Büchi automata translation. In Gérard Berry, Hubert Comon, and Alain Finkel, editors, *Proceedings of the 13th International Conference on Computer Aided Verification (CAV'01)*, volume 2102 of *Lecture Notes in Computer Science*, pages 53–65, Paris, France, July 2001. Springer.

[42] Gergana. Octave networks toolbox first release, July 2014.

[43] Rob Gerth, Doron Peled, Moshe Y Vardi, and Pierre Wolper. Simple on-the-fly automatic verification of linear temporal logic. In *International Conference on Protocol Specification, Testing and Verification*, pages 3–18. Springer, 1995.

[44] Jordi Guillen and Arina Motova. Summary of the 2013 Economic Performance Report on the EU Aquaculture Sector (STECF 13-30). Technical report, Scientific, Technical and Economic Committee for Fisheries (STECF), 2013.

[45] Meng Guo, Karl H Johansson, and Dimos V Dimarogonas. Motion and action planning under ltl specifications using navigation functions and action description language. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 240–245. IEEE, 2013.

[46] Nicholas Hansen, Mikkel C Nielsen, David Johan Christensen, and Mogens Blanke. Short-range sensor for underwater robot navigation using line-lasers and vision. *IFAC-PapersOnLine*, 48(16):113–120, 2015.

[47] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[48] Janne Heikkila and Olli Silvén. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106–1112. IEEE, 1997.

[49] Thomas A Henzinger. The theory of hybrid automata. In *Verification of digital and hybrid systems*, pages 265–292. Springer, 2000.

[50] Andrew Howard, Maja J Matarić, and Gaurav S Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. *Autonomous Robots*, 13(2):113–126, 2002.

[51] Andrew Howard, Maja J Matarić, and Gaurav S Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Distributed Autonomous Robotic Systems 5*, pages 299–308. Springer, 2002.

[52] Chai-Cheng Huang, Hung-Jie Tang, and Jin-Yuan Liu. Dynamic Analysis of Net Cage Structures for Marine Aquaculture: Numerical Simulaion and Model Testing. *Aquacultural Engineering*, 35:258–270, 2006.

[53] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. A Quadratic-Compexity Observability-Constrained Uscented Kalman Filter for SLAM. *IEEE Transactions on Robotics*, 29(5):1226–1243, 2013.

[54] J.L. Hess and A.M.O. Smith. Calculation of Potential Flow About Arbitrary Bodies. *Progress in Aeronautical Sciences*, 8:1–183, 1967.

[55] Anne Jordt-Sedlazeck and Reinhard Koch. Refractive calibration of underwater cameras. *Computer Vision–ECCV 2012*, pages 846–859, 2012.

[56] Simon Julier, Jeffrey Uhlmann, and Hugh F. Durrant-Whyte. A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, 2000.

[57] George C. Karras, Savvas G. Loizou, and Kostas J. Kyriakopoulos. Towards Semi-autonomous Operation of Under-actuated Underwater Vehicles: Sensor Fusion, On-line Identification and Visual Servo Control. *Autonomous Robots*, 31(1):67–86, 2011.

[58] George C Karras, Dimitra J Panagou, and Kostas J Kyriakopoulos. Target-referenced localization of an underwater vehicle using a laser-based vision system. In *OCEANS 2006*, pages 1–6. IEEE, 2006.

[59] George C. Karras, Dimitra J. Panagou, and Kostas J. Kyriakopoulos. Target-referenced Localization of an Underwater Vehicle using a Laser-based Vision System. In *OCEANS*, 2006.

[60] Joseph Katz and Allen Plotkin. *Low-Speed Aerodynamics*. Cambridge Aerospace Series. Cambridge University Press, 2nd edition, 2011.

[61] Pascal Klebert, Pal Lader, Lars Gansel, and Frode Oppedal. Hydrodynamic Interactions on Net Panel and Aquaculture Fish Cages: A Review. *IEEE Journal of Ocean Engineering*, 58:260–274, 2013.

[62] Marius Kloetzer and Calin Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 53(1):287–297, 2008.

[63] Dimitrios G. Kottas and Stergios I. Roumeliotis. Efficient and Consistent Vision-Aided Inertial Navigation using Line Observation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[64] Ron Koymans. Specifying real-time properties with metric temporal logic. *Real-time systems*, 2(4):255–299, 1990.

[65] Hadas Kress-Gazit, Georgios E Fainekos, and George J Pappas. Temporal-logic-based reactive mission and motion planning. *IEEE transactions on robotics*, 25(6):1370–1381, 2009.

[66] Clayton Kunz and Hanumant Singh. Hemispherical refraction and camera calibration in underwater vision. In *OCEANS 2008*, pages 1–7. IEEE, 2008.

[67] Pal F. Lader, Birger Enerhaug, Arne Fredheim, and Jorgen Krokstad. Modelling of 3D Net Structures Exposed to Waves and Current. In *International Conference on Hydroelasticity in Marine Technology*, 2003.

[68] Jean-Claude Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.

[69] Jean-Paul Laumond et al. *Robot motion planning and control*, volume 229. Springer, 1998.

[70] Jun Liu and Pavithra Prabhakar. Switching control of dynamical systems from metric temporal logic specifications. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5333–5338. IEEE, 2014.

[71] Savvas G Loizou. The navigation transformation. *IEEE Transactions on Robotics*, 33(6):1516–1523, 2017.

[72] Savvas G Loizou and Christos C Constantinou. Multi-robot coverage on dendritic topologies under communication constraints. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 43–48. IEEE, 2016.

[73] Savvas G Loizou and Kostas J Kyriakopoulos. Automatic synthesis of multi-agent motion tasks based on ltl specifications. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 1, pages 153–158. IEEE, 2004.

[74] Flávio Lopes, Hugo Silva, José Miguel Almeida, Alfredo Martins, and Eduardo Silva. Structured light system for underwater inspection operations. In *OCEANS 2015-Genova*, pages 1–6. IEEE, 2015.

[75] Manolis IA Lourakis. A brief description of the levenberg-marquardt algorithm implemented by levmar. *Foundation of Research and Technology*, 4(1):1–6, 2005.

[76] Simon Lynen, Sammy Omari, Matthias Wuest, Markus Achtelik, and Roland Siegwart. Tightly Coupled Visual-Inertial Navigation System Using Optical Flow. In *2nd IFAC Workshop on Research, Education and Development of Unmanned Aerial Systems*, 2013.

[77] Oded Maler, Dejan Nickovic, and Amir Pnueli. From mitl to timed automata. In *FORMATS*, volume 4202, pages 274–289. Springer, 2006.

[78] Musa Morena Marcusso Manhães, Sebastian A Scherer, Martin Voss, Luiz Ricardo Douat, and Thomas Rauschenbach. Uuv simulator: A gazebo-based package for underwater intervention and multi-robot simulation. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–8. IEEE, 2016.

[79] Zohar Manna and Amir Pnueli. *The temporal logic of reactive and concurrent systems: Specification*. Springer Science & Business Media, 2012.

[80] Rudolph Van Der Merwe and Eric A. Wan. The Square-Root Unscented Kalman Filter for State and Parameter-Estimation. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 3461–3464, 2001.

[81] K Muljowidodo, Mochammand A Rasyid, N SaptoAdi, and Agus Budiyono. Vision based distance measurement system using single laser pointer design for underwater vehicle. *Indian journal of marine science*, 38(3):324–331, 2009.

[82] Alexandros Nikou, Dimitris Boskos, Jana Tumova, and Dimos V Dimarogonas. Cooperative planning for coupled multi-agent systems under timed temporal specifications. In *American Control Conference (ACC), 2017*, pages 1847–1852. IEEE, 2017.

[83] Anshika Pal, Ritu Tiwari, and Anupam Shukla. Communication constraints multi-agent territory exploration task. *Applied intelligence*, 38(3):357–383, 2013.

[84] Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 46–57. IEEE, 1977.

[85] Sameera Poduri and Gaurav S Sukhatme. Constrained coverage for mobile sensor networks. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 165–171. IEEE, 2004.

[86] Svetlana Potyagaylo, Christos C Constantinou, George Georgiades, and Savvas G Loizou. Asynchronous ukf-based localization of an underwater robotic vehicle for aquaculture inspection operations. In *OCEANS 2015-MTS/IEEE Washington*, pages 1–6. IEEE, 2015.

[87] Svetlana Potyagaylo, Christos C. Constantinou, George Georgiades, and Savvas G. Loizou. Asynchronous UKF-based Localization of an Underwater Robotic Vehicle for Aquaculture Inspection Operations. In *OCEAS Conference*, 2015.

[88] Svetlana Potyagaylo and Savvas G. Loizou. Online Adaptive Geometry Predictor of Aquaculture Fish-Nets. In *Mediterranean Conference on Control and Automation (MED)*, June 2014.

[89] Arthur N Prior. *Past, present and future*, volume 154. Clarendon Press Oxford, 1967.

[90] Daniel Priour. *A Finite Element Method for Netting: Application to Fish Cages and Fishing Gear*. Springer Briefs in Environmental Science. Springer, 2013.

[91] Morgan Quigley, Josh Faust, Tully Foote, and Jeremy Leibs. Ros: an open-source robot operating system.

[92] Ioannis Rekleitis, Ai Peng New, Edward Samuel Rankin, and Howie Choset. Efficient boustrophedon multi-robot coverage: an algorithmic approach. *Annals of Mathematics and Artificial Intelligence*, 52(2-4):109–142, 2008.

[93] Miguel Ribo and Markus Brandner. State of the art on vision-based structured light systems for 3d measurements. In *Robotic Sensors: Robotic and Sensor Environments, 2005. International Workshop on*, pages 2–6. IEEE, 2005.

[94] Chris Roman, Gabrielle Inglis, and James Rutter. Application of structured light imaging for high resolution mapping of underwater archaeological sites. In *OCEANS 2010 IEEE-Sydney*, pages 1–9. IEEE, 2010.

[95] Paulo Tabuada and George J Pappas. Linear time logic control of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 51(12):1862–1877, 2006.

[96] Tali Treibitz, Yoav Y Schechner, and Hanumant Singh. Flat refractive geometry. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[97] Roger Y Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344, 1987.

[98] Moshe Y Vardi. An automata-theoretic approach to linear temporal logic. In *Logics for concurrency*, pages 238–266. Springer, 1996.

[99] Eric A. Wan and Rudolph van der Merwe. The Unscented Kalman Filter for Nonlinear Estimation. In *IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158, 2000.

[100] CC Wang, SW Shyue, HC Hsu, JS Sue, and TC Huang. Ccd camera calibration for underwater laser scanning system. In *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, volume 4, pages 2511–2517. IEEE, 2001.

[101] Wei Wang and Christopher M. Clark. Modeling and Simulation of the VideoRay Pro III Underwater Vehicle. In *OCEANS*, 2007.

[102] Louis L Whitcomb. Underwater robotics: Out of the research laboratory and into the field. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 709–716. IEEE, 2000.

[103] Eric M Wolff, Ufuk Topcu, and Richard M Murray. Efficient reactive controller synthesis for a fragment of linear temporal logic. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5033–5040. IEEE, 2013.

[104] Bugao Xu. Identifying fabric structures with fast fourier transform techniques. *Textile Research Journal*, 66(8):496–506, 1996.

[105] Junku Yuh. Design and control of autonomous underwater robots: A survey. *Autonomous Robots*, 8(1):7–24, 2000.

[106] Junku Yuh and Michael West. Underwater robotics. *Advanced Robotics*, 15(5):609–639, 2001.

[107] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666–673. IEEE, 1999.

[108] Yun-Peng Zhao, Yu-Cheng Li, Guo-Hai Dong, Fu-Kun Gui, and Bin Teng. Numerical Simulation of the Effects of Structure Size Ratio and Mesh Type on Three-Dimensional Deformation of the Fishing-Net Gravity Cage in Current. *Aquacultural Engineering*, 36:285–301, 2007.

[109] Yun-Peng Zhao, Tiao-Jian Xu, Chun-Wei Bi, Guo-Hai Dong, and Sheng-Cong Liu. The Numerical Simulation of Hydrodynamics of Fishing Net Cage. *Hydrodynamics - Theory and Model*, 2012.

[110] Yuchen Zhou, Dipankar Maity, and John S Baras. Timed automata approach for motion planning using metric interval temporal logic. In *Control Conference (ECC), 2016 European*, pages 690–695. IEEE, 2016.