National Technical University of Athens

School of Mechanical Engineering

Mechanical Design Option

Control Systems Laboratory

Cooperation between two underwater vehicles for ship hull inspection using implicit communications and optical feedback

GEORGIOS KONSTANTINOS KARAVAS

Submitted in partial fulfillment of the requirements for the degree of

DIPLOMA IN MECHANICAL ENGINEERING

Diploma Thesis Advisor: Professor Kostas J. Kyriakopoulos

Athens, July 18, 2013

[This page intentionally left blank.]

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Μηχανολόγων Μηχανικών

Ροή Κατασκευαστή

Εργαστήριο Αυτομάτου Ελέχγου και Ρύθμισης Μηχανών

Συνεργασία δύο υποβρύχιων ρομποτικών οχημάτων για την επισκόπηση υφάλων πλοίου με χρήση έμμεσης επικοινωνίας και οπτικής ανατροφοδότησης

ΓΕΩΡΓΙΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ ΚΑΡΑΒΑΣ

Κατατέθηκε για τη μερική εκπλήρωση των υποχρεώσεων για την απόκτηση του τίτλου

ΔΙΠΛΩΜΑΤΟΥΧΟΥ ΜΗΧΑΝΟΛΟΓΟΥ ΜΗΧΑΝΙΚΟΥ

Υπεύθυνος Καθηγητής: Κωνσταντίνος Ι. Κυριακόπουλος

Αθήνα, July 18, 2013

``Everything will be all right in the end. If it’s not all right, it is not yet the end."
- Patel, Hotel Manager, The Best Exotic Marigold Hotel

``It never gets easier. You just get better."
- Anonymous

Contents

List of Figures

Acknowledgements

First of all, I would like to thank my advisor, Professor Kostas J. Kyriakopoulos, for our collaboration over the last three years. His advice and perceverance helped me exceed my capabilities and achieve goals that I never imagined I could. His insight and deep understanding of the field of Robotics gave me the opportunity to work on a very interesting subject that included many diverse problems and ideas. This subject introduced me to a wide variety of concepts, from image processing and computer vision to nonlinear control, and provided me with a wide perspective with respect to the field of Robotics and Automation and its applications. It would be difficult to acquire this perspective otherwise.

I wish also to thank the postdoctoral researcher George C. Karras whose work was used as a guidance for my thesis. His assistance and his advice helped me understand the value of being a researcher and he showed me the way to be a successful one. He taught me how to be more efficient and he helped me enhance my problem solving capabilities.

I wish to express my gratitude to my colleague and PhD candidate Shahab Heshmati for his friendship and his valuable help during the period where the experiments were taking place. I also wish to thank my close friend and former colleague John Fillipidis for his support, his advice and the time he devoted on discussing with me various aspects of my thesis.

I would like to thank my former and current colleagues at the Control System Lab, namely Panos Marantos, Dimitra Panagou, Spyros Maniatopoulos, Alina Eqtami, Dimitris Paraskevas, Nikos Kampras, Minas Liarokapis and Pantelis Katsiaris for being good friends and excellent colleagues. Their friendly spirit provided a very pleasant working environment that helped me to be productive and efficient.

Last but not least, I would like to thank deeply my parents Andreas and Suzanna as well as my brother Angelos for their patience, their valuable support when things got difficult and their understanding during this whole time. Without their assistance, I would not be able to achieve what I have achieved so far.

English Abstract

This thesis concerns the cooperation of two underwater vehicles with different kinematic capabilities for the inspection of a flat surface, e.g. a ship hull. The vehicles do not communicate explicitly with each other. Instead they use an implicit communications strategy which is based on a configuration of underwater laser pointers.

In particular, the vehicles exchange localization information. This information is related to their relative pose (position and orientation) with respect to each other and to their pose with respect to the inspected surface. A specific relative localization procedure is designed. This procedure defines the setup of laser pointers that must be equipped on each vehicle. The laser pointers of each vehicle project their beams onto the surface and produce distinct laser dots that are visible from the camera sensor of the other vehicle. A dedicated computer vision algorithm is designed and implemented for the detection of these laser dots on the image frame of the camera. The vehicles use this localization procedure in order to navigate themselves relative to each other and relative to the surface.

The vehicles cooperate using a Leader - Follower approach. One vehicle is nonholonomic with respect to its planar motion. This vehicle is the Leader of the formation. It moves independently from the other vehicle and follows a prescribed meander-like trajectory which is defined by waypoints. It must always face the inspected surface despite its nonholonomic constraints. To this end, a dedicated motion controller is designed that addresses this issue. On the other hand, the laser dots projected by the Leader must never leave the Follower image plane because otherwise the localization procedure will fail to produce any results. This is due to the fact that the localization information depends directly on the position of these dots inside the Follower image frame. Thus, a motion controller for the Follower is designed that combines notions from both position-based and image-based visual servoing. This controller guarantees that the Leader laser dots will never leave the Follower image plane, while at the same time the vehicle will be able to move at a specific distance offset with respect to the Leader.

The efficacy of the localization procedure and the motion controllers is tested through simulation. It is also tested through an experimental procedure, except for the case of the Follower motion controller where the procedure is designed and the setup is complete, but the actual experiment is not performed yet.

Keywords: underwater vehicles, explicit / implicit communications, visual servoing, cooperation / coordination, localization, motion constraints, Leader - Follower approach, decentralized control

Greek Abstract

Η διπλωματική αυτή αναφέρεται στη συνεργασία δύο υποβρύχιων ρομποτικών οχημάτων με διαφορετικές κινηματικές δυνατότητες για την επόπτευση μιας πλατιάς επιφάνειας, όπως για παράδειγμα των υφάλων ενός πλοίου. Τα οχήματα δεν επικοινωνούν άμεσα μεταξύ τους. Αντίθετα χρησιμοποιούν μια στρατηγική έμμεσης επικοινωνίας, η οποία βασίζεται σε μια διάταξη υποβρυχίων lasers.

Συγκεκριμένα, τα οχήματα ανταλλάσσουν πληροφορίες χωροθεσίας. Αυτές οι πληροφορίες συνδέονται με τη σχετική πόζα (pose, θέση και προσανατολισμό) μεταξύ τους και την πόζα τους σε σχέση με την υπό επόπτευση επιφάνεια. Στα πλαίσια της δπλωματικής αυτής σχεδιάστηκε μια ειδική διαδικασία σχετικής χωροθεσίας. Αυτή η διαδικασία ορίζει το setup των lasers που θα πρέπει να τοποθετηθούν σε κάθε όχημα. Τα lasers κάθε οχήματος προβάλλουν τις ακτίνες τους πάνω στην επιφάνεια και δημιουργούν ευδιάκριτες κουκκίδες που είναι ορατές από τον οπτικό σένσορα (camera) του άλλου υποβρυχίου. Ένας ειδικός αλγοριθμος τεχνητής όρασης σχεδιάστηκε και υλοποιήθηκε για τον εντοπισμό των κουκκίδων αυτών στο επίπεδο της εικόνας της κάμερας. Τα οχήματα χρησιμοποιούν αυτή τη διαδικασία χωροθεσίας προκειμένου να πλοηγηθούν σε σχέση με τα ίδια και σε σχέση με την επιφάνεια.

Τα οχήματα συνεργάζονται χρησιμοποιώντας μια προσέγγιση Αρχηγού - Ακόλουθου (Leader - Follower approach). Ένα από τα οχήματα είναι μη-ολονομικό σε σχέση με την επίπεδη κινησή του. Αυτό το όχημα είναι ο Αρχηγός του σχηματισμού. Κινείται ανεξάρτητα από το άλλο όχημα και ακολουθεί μια προδιαγεγραμμένη διαδρομή που μοιάζει με μέανδρο και ορίζεται από σημεία διέλευσης. Το όχημα πρέπει πάντα να παρατηρεί την υπό επόπτευση επιφάνεια παρά τους μη-ολονομικούς περιορισμούς του. Για αυτό το σκοπό, σχεδιάστηκε ένας εξειδικευμένος ελεγκτής κίνησης ο οποίος αντιμεντωπίζει το εν λόγω πρόβλημα. Από την άλλη, οι κουκκίδες των lasers που προβάλλονται από τον Αρχηγό δεν πρέπει ποτέ να φύγουν από το οπτικό πεδίο του Ακόλουθου διότι αλλιώς η διαδικασία χωροθεσίας δεν θα μπορέσει να δώσει αποτελέσματα. Αυτό οφείλεται στο γεγονός ότι οι πληροφορίες χωροθεσίας εξαρτώνται απευθείας από τη θέση αυτών των κουκκίδων στο επίπεδο της εικόνας του Ακόλουθου. Επομένως, σχεδιάστηκε ένας ελεγκτής κίνησης για τον Ακόλουθο, ο οποίος συνδυάζει έννοιες και από τον οπτικό έλεγχο που βασίζεται στη θέση (position-based visual servoing) αλλά και από αυτόν που βασίζεται στην εικόνα (image-based visual servoing). Ο ελεγκτής αυτός εξασφαλίζει ότι οι κουκκίδες των lasers του Αρχηγού δεν θα εγκαταλείψουν ποτέ το επίπεδο της εικόνας του Ακόλουθου, ενώ ταυτόχρονα το όχημα θα μπορεί να κινείται σε συγκεκριμένη απόσταση σε σχέση με τον Αρχηγό.

Η αποτελεσματικότητα της διαδικασίας χωροθεσίας και των ελεγκτών κίνησης των οχημάτων ελέγχθηκε μέσω προσομοίωσης. Η αποτελεσματικότητα τους ελέγχθηκε και μέσα από μια πειραματική διαδικασία, εκτός από την περίπτωση του ελεγκτή του Ακόλουθου όπου σχεδιάστηκε η διαδικασία και ολοκληρώθηκε το setup, αλλά το πείραμα δεν έχει πραγματοποιηθεί ακόμη.

Λέξεις Κλειδιά: υποβρύχια ρομποτικά οχήματα, άμεση/έμμεση επικοινωνία, οπτική ανατροφοδότηση, συνεργασία / συντονισμός, χωροθεσία, κινηματικοί περιορισμοί, προσέγγιση Αρχηγού - Ακόλουθου, αποκεντρωμένος έλεγχος

Chapter 1

Preface

1.1 Introduction

The domain of underwater robotics includes the results from many different disciplines of robotics, such as machine vision, path planning and motion control. Underwater autonomous platforms are used quite often to replace humans in difficult, dangerous or repetitive tasks and missions, such as wreckage recovery, ship hull or pipeline inspection, marine life observation and repair of underwater structures. The two most common underwater autonomous platforms are the Remotely Operated Vehicles (ROVs) and the Autonomous Underwater Vehicles (AUVs).

The main chracteristic of a ROV is the fact that it is attached to an offshore Control Unit (CU) through a long umbilical cable (also known as tether). The user is able to steer the vehicle using a joystick attached or connected to the CU. Usually, he can also monitor the surroundings of the vehicle using its on-board camera. The cable plays a very important role as it transfers data between the vehicle and its CU, while it supplies power to the vehicle. On the other hand, the cable can also pose certain constraints to the steering of the ROV and the distance it can travel. In most cases, commercial ROVs are small, since their power source does not need to be on-board. They can reach great depths and they may be equipped with various sensors and devices, such as cameras, bathymeters, magnetic compasses, lights and grippers.

On the other hand, AUVs, as the acronym implies, are autonomous, with no cables attached to them and no direct interaction between the vehicle and its operator. Any interaction with an AUV employs either electromagnetic signals when they are on the surface or acoustic signals when they are underwater. They are usually large because they need to carry batteries as their power source and they can also be equipped with many types of sensors. Their autonomy is a key aspect in completing successfully complex and difficult tasks. It is also worth mentioning that in most cases researchers design and test algorithms and methodologies for ROVs that are intended to be used to AUVs.

An idea that gained a lot of ground and was studied extensively through numerous works and papers over the last decades was to form groups of such vehicles (often called *platoons* or *fleets* of underwater vehicles) in order to execute more complex tasks or simply enhance the execution of simple ones.The use of multiple underwater vehicles provides robustness and efficiency during mission execution. First, there is a redundancy of vehicles which may prove to be very useful when one vehicle fails to operate because the other (or others) will continue the execution of the task. Second, this approach improves the efficiency of the vehicles and minimizes the execution times in cases where the vehicles have to cover great areas (e.g. surveillance, inspection). There are more vehicles to execute the task and thus they can divide it into smaller portions and execute it more quickly and more efficiently. Finally, there may be an issue of resources where one vehicle is equipped with certain sensors and another vehicle is equipped with a different sensing setup. The two vehicles can cooperate in order to combine their sensory information or compensate for their respective lack of sensory equipment.

This leads to the notions of coordination and cooperation. The term of *coordination* refers to the case where a central module coordinates the agents of the fleet. It plans the execution of the task and it sends control inputs to the agents in order to execute it. The communication between them is not present or is barely present. On the contrary, the term of *cooperation* refers to the case where the agents communicate and interact with each other in order to choose the most efficient way to execute the mission. Sometimes the two terms are used interchangeably in the literature. This depends on the level of communication between the agents and the existence or nonexistence of a central module.

As it is already mentioned, the cooperation between a number of vehicles or agents necessitates some form of communication. This communication can be either explicit or implicit. Explicit communication is the communication where direct messages that contain specific information are exchanged between the agents. The agents receive these messages, interpret them and act upon them. Explicit communication in underwater domain may involve acoustic signals when the vehicles are underwater and wireless communication when they are on the surface. Implicit communication, on the other hand, refers to the case where an agent changes its environment in a way that it can be perceived from the other agents and no expicit information exchange takes place. The other agents understand and interpret this change and perform actions accordingly.

1.2 Literature Review

This work concerns the cooperation between two underwater vehicles for the inspection of a flat surface (e.g. a ship hull). The vehicles use implicit communications in order to navigate themselves with respect to each other and to the surface.

There are many ways in which two vehicles can cooperate with each other. The main cooperative actions in the underwater domain can be divided into three (3) main categories: data acquisition and registration (e.g. ocean sampling), cooperative task execution (e.g. obstacle avoidance [40], cooperative navigation and localization [6, 4], SLAM [11], area coverage) and cooperative mission planning and execution, which usually includes task allocation and optimized task execution. For example, in [40] the authors used a potential field approach based on the principles of electric fields for obstacle avoidance of a formation. In [6] a number of Autonomous Surface Crafts is used as Communication/Navigation Aids in order to coordinate a group of AUVs using dead-reckoning, acoustic ranging data and GPS measurements.

In order to ensure the fulfillment of the cooperation goals, a number of methods has been used. These methods refer to (a) formation control strategies, (b) behavior based approaches and (c) design of complex frameworks and architectures that provide diverse capabilities and include several modules (e.g. [36, 21]). Referring to formation control strategies, these can be divided into (a) *Leader - Follower* approaches (e.g. [10, 12]), where a vehicle is declared as the Leader of the fleet and the other vehicles (Followers) follow its motion, (b) *Virtual Structure* approaches (e.g. [24, 34]), where the entire fleet is considered as a compact structure, (c) *Potential Fields* approaches (e.g. [23]), where the formation of the fleet is controlled by virtual forces induced by neighbouring vehicles and (d) *behavioural* approaches (e.g. [3]), where the formation of the fleet is controlled based on a behaviour pattern which changes according to the situation.

As far as communication is concerned, explicit communication is mainly used. As already mentioned, this kind of communication usually involves some sort of acoustic communication ([6, 4]), e.g. use of acoustic modems ([18, 37]). Data exchange using acoustic signals can present some drawbacks though. These drawbacks may relate to low communication bandwidth and possible loss of communication during the mission execution. In addition, acoustic signals can provide erroneous information due to reflections on hard surfaces. Many researchers develop cooperation strategies and architectures without taking these constraints into account. Others, though they do take into consideration the above constraints ([16, 8]), they assume that at some point communication will be established (even briefly) (in [35] the authors examine the use of time-varying networks). Thus, even not in a continuous manner, explicit communication between vehicles is present.

These problems were the motivation for the development of an implicit communications strategy between the vehicles. The use of implicit communications in underwater domain has not been extensively examined and may be considered rare in the literature. In some cases, it may involve the use of methods that rely on potential fields, as in [5, 28], where the inter-vehicle forces induced by the potential fields provide the necessary implicit communication to drive the formation towards the global minimum (or maximum) of a sampled environmental gradient field ([5]) or to navigate the vehicles around certain obstacles in an unknown environment ([28]). This is in contrast to the mobile robotics domain where the literature is more rich ([25, 30, 19]). In [33], the authors present the cooperation of two mobile robots that carry an object. Their implicit communication scheme is based on the force sensors that they are equipped with. In [32] authors use color markers equipped on mobile robots to provide implicit communication using vision with respect to robot position and robot identification.

With respect to the task itself, it must be noted that the inspection of underwater structures is a familiar application in underwater domain and a great deal of work has been done towards its solution. Most approaches use sonars and/or acoustic sensors ([13]), although vision has also been used for this purpose ([26]). Cooperative inspection includes pipeline inspection ([41]), ship-hull inspection and inspection of other underwater structures ([42]). In [41], a leader-follower strategy is followed. The cooperation between the vehicles is established via acoustic communication between the Leader and the Followers which transmit as minimum information as possible. In [42], the vehicles communicate with each other through a smart cable that connects them.

It is also worth mentioning that, in this work, the case of vehicles with different kinematic capabilities is considered. One of the vehicles is subject to nonholonomic constraints because it is underactuated along its lateral axis of motion, i.e. it has no thrusters to provide lateral motion. The other vehicle is fully actuated. The navigation of nonholonomic vehicles is a problem that has been studied by numerous researchers. One such work is presented in [2] where the propose a control scheme which is based on Lyapunov stability. Another example, which is also related to cooperation, is the work of [14] where the authors propose a formation control algorithms for nonholonomic vehicles.

1.3 Main Contributions

In this work, we present a decentralized cooperation scheme between two underwater vehicles for the inspection of a flat surface. During the cooperation, the vehicles follow a Leader - Follower approach. The vehicles use implicit communications based on a setup of laser pointers equipped on the vehicles and a computer vision algorithm. We also present a control scheme for the motion of a nonholonomic vehicle in parallel to a flat surface while facing it. This control scheme is used to navigate the Leader vehicle in front of the surface to be inspected. The Leader vehicle must reach a set of desired waypoints in order to execute a meander-like trajectory. A motion control scheme for the motion of the Follower is also presented. The Follower must comply to certain cooperation constraints while at the same time moving at a specific distance offset with respect to the Leader.

Thus, two main contributions are presented. The first relates to the implementation of the implicit communication strategy based on the laser pointer configuration. A separate laser pointer setup is equipped on each vehicle and a localization strategy is derived. Each vehicle can then localize itself with respect to the inspected surface and to the other vehicle. As already mentioned, the motivation for the derivation of this procedure was the disadvatages related to the various explicit communications strategies. The proposed approach provides higher data rates compared to its acoustic counterparts, high accuracy and no erroneous data. As long as the laser dots projected by one vehicle lie inside the image frame of the other vehicle, there is no chance of data loss and there are no bandwidth limitations. The only limitations relate to the type of the exchanged information (only localization information).

The second contribution relates to the design of a motion control scheme for the cooperation of the vehicles based on the implicit communication strategy. This includes two motion controllers, one for each vehicle. As mentioned, the vehicles follow a Leader - Follower approach in order to maintain their relative formation. This means that the Leader moves independently from the Follower. The latter follows the motion of the Leader by observing the laser dots projected by its laser pointers. In order to always keep the Leader laser dots inside its image frame, the Follower has to overcome the visibility constraints imposed by the implicit communications system. The Follower motion controller adresses these constraints by taking into account the motion of the Leader and its velocity. In other words, the controller tracks the motion of the Leader by observing its laser dots. To the best knowledge of the author, this is a unique technique that is not present in the literature. It is task specific, i.e. it can be used only for inspection operations, but it provides cooperation between two vehicles without direct observation of each other as in other vision-based cooperation cases.

Finally, the theoretical results of this work are validated through a simulation and an experimental procedure. The experimental setup includes the vehicles, a certain number of underwater laser pointers and a dedicated construction for each vehicle that hosted the laser pointers. It also includes the vision algorithm that detects the laser dots inside the image frame and the corresponding motion control algorithms. The experimental procedure was divided in steps where each component was tested before used in the final experiment.

1.4 Thesis Structure

The rest of this thesis is organized as follows. Chapter 2 defines the main problem and its subproblems and presents the modelling of the vehicles and their environment, while Chapter 3 presents the approach for the solution of the previously defined problems. Chapter 4 presents the state constraints related to the cooperation of the vehicles and it also analyses their derivation. Chapter 5 presents the localization procedure, while Chapter 6 describes the computer vision algorithm employed by the Follower vehicle. Chapter 7 includes the motion control schemes for the two vehicles, while Chapter 8 presents simulation results of the theory described in the previous chapters. Finally, Chapter 9 presents the equipment used during the experiments, while Chapter 10 describes the experimental procedure and presents the corresponding results.

Chapter 2

Problem Overview

This section defines the basic problem and provides the appropriate mathematical analysis to describe it accurately and in a concise manner. The problem can be stated as follows:

Move two underwater vehicles with different kinematic capabilities in a cooperative manner in order to inspect a flat surface without the use of explicit communications.

The underwater vehicles, also referred to as Unmanned Underwater Vehicles (UUVs), must cooperate in order to inspect a flat surface. They must follow a meander-like trajectory defined by waypoints (WPs) maintaining a specific distance offset with respect to each other in order to provide coverage of the surface. Their cooperation is based on a Leader-Follower approach. During their mission, they must not use any explicit communications.

In the next sections, we will provide the environment and the vehicle modelling. We will also present the *Implicit Communications Problem* and the *Motion Control Problem*.

2.1 Environment Modelling

The space in which the vehicles move can be described as a 3D cartesian space with no stationary obstacles.

Assumption 1: There are no other objects or agents in the 3D space other than the two UUVs and the surface. Thus, for each vehicle the only obstacle that may obstruct its line-of-sight to the surface is the other vehicle.

The vehicles must inspect a flat surface placed in an upright position. The surface divides the 3D space into two subspaces. Let O_W denote the origin of a *World Coordinate System (WCS)* which is fixed on the surface. Its yz-plane is parallel to the surface and its y-axis is parallel to the horizontal edges of the surface, as shown in Fig. 2.1. Let O_G denote the origin of a *Global Coordinate System (GCS)* which lies on the surface and is parallel to the WCS, as it is also shown in the same figure. The origin O_G coincides with the intersection of the longitudinal axis of the Leader vehicle and the surface (Fig. 2.1). The word ``Global'' refers to the formation of the two vehicles. The two subspaces can then be defined as follows:

$$\begin{aligned} S_1 &= \{x, y, z \in \mathbb{R} : x > 0\} \\ S_2 &= \{x, y, z \in \mathbb{R} : x < 0\} \end{aligned} \tag{2.1}$$

where x, y, z are the coordinates of a vehicle with respect to the WCS.

Assumption 2: The vehicles move inside S_2 and they never enter S_1. This means that the vehicles inspect the surface only from its front.

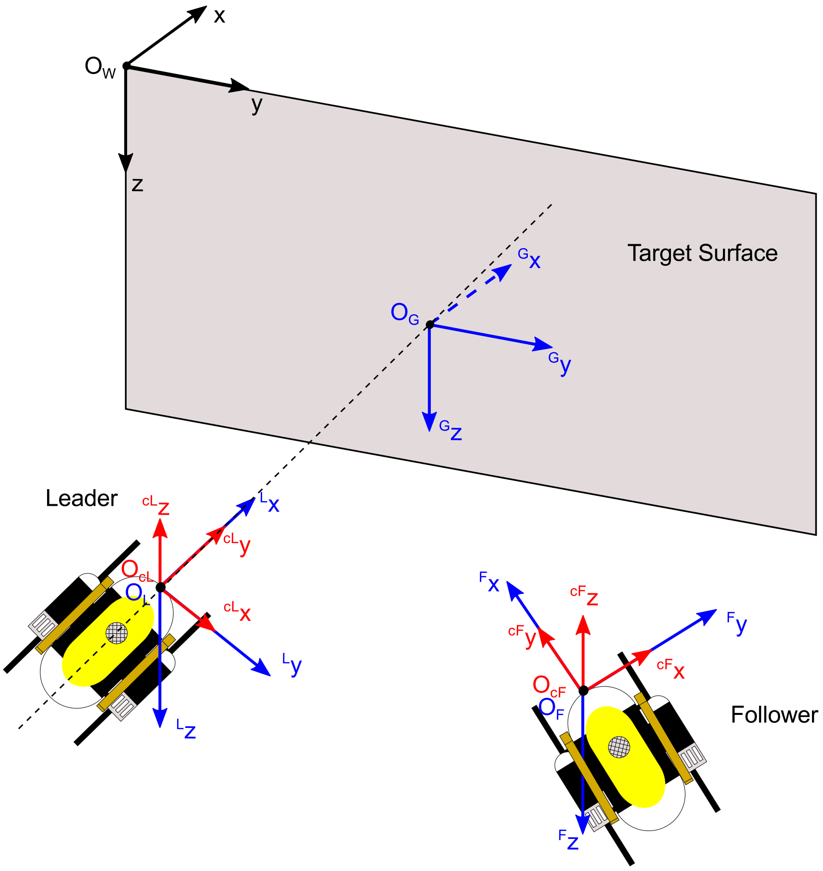


Figure 2.1: The coordinate systems for the vehicles and the surface. On each vehicle blue colour indicates the CS of the vehicle and red colour the CS of its camera.

Let O_L and O_F define the origins of two coordinate systems mounted on the Leader (*Leader Coordinate System, LCS*) and the Follower vehicle (*Follower Coordinate System, FCS*), respectively. For each CS, the x-axis coincides with the longitudinal axis of the vehicle and the other axes form an orthonormal CS with the x-axis.

Let O_{CL} and O_{CF} define the corresponding origins of the camera coordinate systems on each vehicle, i.e. *Leader camera Coordinate System* (LcCS) and *Follower camera Coordinate System* (FcCS), respectively. For those CSs, it is the y-axis that coincides with the longitudinal axis on each vehicle and the optical axis of the camera and the others form an orthonormal CS with the y-axis.

It is assumed that the origins $O_{\{L,F\}}$ and $O_{C\{L,F\}}$ on each vehicle coincide. In order to trasform the coordinates of a point from the CS of the vehicle to the CS of its camera, we use the following relation:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.2}$$

2.2 Vehicle Modelling

The pose (position and orientation) vector of a 6 Degrees-of-Freedom (DoF) marine vehicle is defined as follows:

$$\mathbf{n} = [\ x \quad y \quad z \quad \phi \quad \theta \quad \psi\ ]^T \tag{2.3}$$

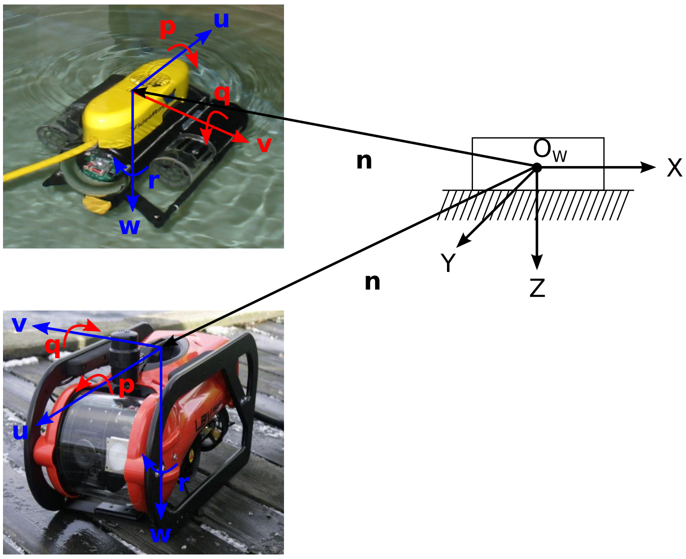


Figure 2.2: VideoRay PRO and Seabotix LBV. Red color indicates no actuation availability, while blue color indicates actuation availability along body frame axes.

where x, y, z refer to position and ϕ, θ, ψ are the Euler angles with respect to an earth-fixed coordinate system. The corresponding velocity vector is written as:

$$\mathbf{v} = [\begin{array}{cccccc} u & v & w & p & q & r \end{array}]^T \tag{2.4}$$

where u, v, w are the 3D linear velocities and p, q, r are the 3D angular velocities with respect to a body-fixed coordinate system. The components of the velocity vector are named according to SNAME [39] as surge, sway, heave, roll, pitch and yaw respectively (see also Appendix A).

In this work, two Remotely Operated Vehicles (ROVs) are used: a 3 DoF Videoray Pro and a 4 DoF Seabotix LBV (Fig. 2.2). Both vehicles are under-actuated but statically stable (configuration by design) about x and y axis (roll and pitch respectively). So, the corresponding angles ϕ, θ and angular velocities p and q are negligible and we can consider them to be equal to zero. Thus, the pose and velocity vectors for both vehicles can be written as follows:

$$\begin{aligned} \mathbf{n} &= [\begin{array}{cccc} x & y & z & \psi \end{array}]^T \\ \mathbf{v} &= [\begin{array}{cccc} u & v & w & r \end{array}]^T \end{aligned} \tag{2.5}$$

The VideoRay Pro ROV is equipped with three thrusters, which are effective only in surge, heave and yaw motion, meaning that the vehicle is under-actuated along the sway (lateral) axis and, thus, it is subject to nonholonomic constraints. Its kinematic model is assumed to be that of a unicycle and the corresponding kinematic equations are:

$$\begin{aligned} \dot{x} &= u\cos\psi \\ \dot{y} &= u\sin\psi \\ \dot{z} &= w \\ \dot{\psi} &= r \end{aligned} \tag{2.6}$$

The Seabotix LBV ROV is equipped with four thrusters, which are effective in surge, sway, heave and yaw motion. Based on the kinematic equations for a UUV provided by [15] (also presented in Appendix A) and the fact that $\phi = \theta = 0$ the corresponding equations are:

$$\begin{aligned}\dot{x} &= u\cos\psi - v\sin\psi \\ \dot{y} &= u\sin\psi + v\cos\psi \\ \dot{z} &= w \\ \dot{\psi} &= r\end{aligned} \tag{2.7}$$

During the mission, the VideoRay PRO ROV plays the role of the Leader and the Seabotix LBV ROV the role of the Follower.

2.3 Implicit Communications Problem

In underwater operations that involve multiple agents, the coordination and/or cooperation between them demands certain communication patterns. In most cases, these patterns include acoustic means, such as acoustic modems and sonars, in order to transmit explicit information between the agents. The use of acoustic communications is dictated by the fact that in great depths electromagnetic signals do not travel well and this in turn leads to data loss or even cease of communication. Unfortunately, the use of acoustic signals entails certain drawbacks such as bandwidth and data rate limitations or erroneous transmissions due to reflections on hard surfaces.

On the other hand, implicit communications seem to provide a more efficient way of data exchange. A methodology that is hardly investigated in underwater operations is implicit communications with the use of vision. In [22], Karras et al. presented a methodology that used a pair of underwater laser pointers to provide localization information of a vehicle with respect to a target on a flat surface. One of the goals of this work is to derive a similar methodology that will provide to each vehicle the relative position vector of the other vehicle. Each ROV will calculate its relative position with respect to the other based only on projections from a certain setup of laser pointers onto a flat surface without explicit data exchange.

In short, the implicit communications problem boils down to a problem of relative localization based on visual cues from a setup of laser pointers. In order for the relative localization to be successful, two important premises must be met.

- The laser setup must provide unique localization information, i.e. there must be a one-to-one correspondence between different configurations of the vehicles and the information derived from the laser setup.
- Each vehicle must be able to localize itself both with respect to the surface using its own lasers and with respect to the other vehicle based on the lasers that the latter is equipped with.

In addition, the implicit communications problem, as it is described in this work, includes the derivation of a computer vision algorithm that will detect all laser dots produced by the laser setup inside the image frames of the vehicles. Although the procedure of isolating multiple laser dots in an image is considered almost straightforward for images taken out of the water, this is not the case for underwater images. The problems that may be encountered relate not only to the quality of the camera itself, but mostly to the way the light is perceived by the camera. This is due to absorption and scattering that lead to blurry images, low contrast and poorly defined shapes. The camera is not in a steady position, which limits the choice between available solutions. For instance, background subtraction techniques cannot be used.

The necessary information for the localization of each vehicle and their cooperation is provided only by vision. This localization information is then used to provide the necessary inputs for the controllers of the vehicles. Thus, the vision algorithm must have the following requirements (denoted as *Vision Requirements*):

Accuracy The algorithm must provide the centre of each and every dot in an image with a certain accuracy. It must not produce false positives, i.e. the detected laser dots must correspond to real laser dots. It must also recognise accurately which laser dots belong to the Leader and which to the Follower.

Robustness The algorithm must detect all lasers dots in the image under different lightning conditions that may exist either locally on the surface or globally in the scene. The range of conditions that the algorithm can cope with may not be too wide but it must permit to the algorithm to provide accurate results under normal operation.

Speed The algorithm must produce results in real-time in order to provide data inputs to the controllers at a high rate. This means that the execution time must be small.

2.4 Motion Control Problem

The *Motion Control Problem* refers to the motion of the vehicles and their cooperation. It can be more accurately described by two statements, one for each vehicle:

- For the Leader: Follow a prescribed meander-like trajectory defined by waypoints which lies in parallel to the surface, while maintaining a perpendicular orientation with respect to it.
- For the Follower: Remain at specific distance offset w.r.t. the Leader and to the surface, while keeping all laser dots inside the image frame.

One of the vehicles is subject to nonholonomic constraints. In general, the motion of the vehicles is also subject to other constraints, namely *orientation constraints*, *vision constraints* and *cooperation constraints*. These constraints are imposed to the system either by the task itself (orientation and cooperation constraints) or the laser pointers setup (vision constraints) and they are presented below. The analysis behind the derivation of these constraints is presented in Chapter 4.

Orientation constraint

The vehicles must inspect a flat surface. Thus, they must face the surface at all times during the mission. This is expressed in mathematical terms as follows:

$$-90° < \psi_{min} \leq \psi_L, \psi_F \leq \psi_{max} < 90° \tag{2.8}$$

where ψ_L and ψ_F represent the yaw angle of the Leader and the Follower vehicle respectively, while ψ_{min}, ψ_{max} represent the minimum and maximum values that these angles can take.

Vision constraints

The vehicles must also be able to calculate their relative position with respect to each other. In order to do so, they need to keep the laser dots of the other vehicle inside their own image plane. When the Follower tries to retain the laser dots of the Leader inside its image frame, the corresponding constraints can be written as:

$$y_F - x_F \tan\left(\psi_F - \frac{\beta_H}{2}\right) + \frac{dl}{2\cos\psi_L} \leq 0 \tag{2.9}$$

$$-y_F + x_F \tan\left(\psi_F + \frac{\beta_H}{2}\right) + \frac{dl}{2\cos\psi_L} \leq 0 \tag{2.10}$$

$$x_F \tan\left(\frac{\beta_V}{2}\right) + z_F \leq 0 \tag{2.11}$$

$$x_F \tan\left(\frac{\beta_V}{2}\right) - z_F + max\{H_{L,L}, H_{R,L}\} \leq 0 \tag{2.12}$$

$$H_{L,L} = \left[\frac{-x_L}{\cos\psi_L} - \frac{dl}{2}\tan\psi_L + r_{off,L}\right]\tan\omega_L \tag{2.13}$$

$$H_{R,L} = \left[\frac{-x_L}{\cos\psi_L} + \frac{dl}{2}\tan\psi_L + r_{off,R}\right]\tan\omega_R \tag{2.14}$$

The above equations represent the constraints on the x-y and x-z plane. In essense, they represent the much simpler constaints:

$$\begin{gathered} 0 \leq u_i \leq \text{WIDTH} \\ 0 \leq v_i \leq \text{HEIGHT} \end{gathered} \tag{2.15}$$

where u_i and v_i are the coordinates of the laser dots inside the image frame and WIDTH, HEIGHT are the dimensions of the image frame itself.

Cooperation constraints

During their cooperation the vehicles must comply to certain cooperation constraints. In particular, the vehicles must avoid collisions with each other and they must not block each other's view. These prerequisites are expressed through a *collision constraint* and a *line-of-sight constraint*. The collision constraint is expressed as:

$$R_L + R_F - \sqrt{\frac{l_L^2 + l_F^2}{4} + D^2 + l_L D \sin(\psi_L - \lambda) - l_F D \sin(\psi_F - \lambda) - \frac{l_L l_F}{2}\cos(\psi_L - \psi_F)} \leq 0 \tag{2.16}$$

where D expresses the distance between the coordinate systems of the vehicles, l_L and l_F describe the length of the vehicles and R_L, R_F are the radii of two circles circumscribed around the vehicles. The angles ψ_L, ψ_F represent the orientation (yaw angle) of the Leader and Follower respectively with respect to the global CS, while angle λ can be calculated as:

$$\lambda = \tan^{-1}\left(\frac{x_L - x_F}{y_F - y_L}\right)$$

In the above equation, the pairs (x_L, y_L) and (x_F, y_F) represent the coordinates of the Leader and the Follower vehicle on the x-y plane with respect to the global CS.
As for the line-of-sight constraint, this is expressed as:

$$\psi_F - \frac{\beta_H}{2} - \kappa - \lambda + \frac{\pi}{2} \leq 0 \tag{2.17}$$

where κ and λ are calculated as:

$$\kappa = \sin^{-1}\left(\frac{R_L}{\sqrt{\frac{l_L^2}{4} + (x_L - x_F)^2 + (y_L - y_F)^2 - l_L(x_L \cos\psi_L - x_F \cos\psi_L + y_L \sin\psi_L - y_F \sin\psi_L}}\right)$$

$$\lambda = \tan^{-1}\left(\frac{x_L - x_F - \frac{l_L}{2}\cos\psi_L}{y_F - y_L + \frac{l_L}{2}\sin\psi_L}\right)$$

Apart from the previously described constraints, there are also two important factors that impose constraints to the design of the motion control schemes and they are described below.

- The trajectories of the vehicles should produce smooth sequences of images in order to provide a good inspection quality.
- The vehicles are not equipped with any localization and navigation equipment other than a camera and an on-board bathymeter, while the inspection procedure does not involve any unique target of interest with respect to which a vehicle can localize itself constantly. Thus, the Leader must position itself with respect to the surface based only on measurements of its distance and orientation from it. These measurements can be provided by a localization system as that in [22].

Chapter 3

Technical Approach

In this chapter, the approach for solving the *Implicit Communications Problem* and the *Motion Control Problem* will be presented.

3.1 Implicit Communications

As already described in Chapter 2, the implicit communications problem consists in finding an appropriate laser setup so that the vehicles can localize themselves accurately with respect to the surface as well as to each other.

In [22], Karras et al. proposed a localization procedure in which the vehicle calculated its distance and orientation with respect to a flat surface based on the projections from a pair of laser pointers. Both laser pointers were parallel to the longitudinal axis of the camera (optical axis) mounted on the vehicle. Each laser pointer projected its beam onto the surface thus producing a distinct dot. Then, based on the location of the dots inside its image frame, the vehicle calculated the range of each laser pointer and, finally, using these ranges, its distance and orientation with respect to the surface.

In this work, a similar approach is pursued. The vehicles must exchange with each other localization information using implicit communications. The communication pattern is based on two separate configurations of laser pointers where each configuration corresponds to each of the vehicles. This pattern imposes to the system certain constraints though. For example, when the Follower vehicle moves, the laser dots produced by the pointers of the Leader may leave Follower's image frame. Thus, the constraints relate to the visibility of one vehicle's laser dots inside the image frame of the other vehicle. The motion of the latter must guarantee that the laser dots will not escape its image frame.

Although there is some related work in [22] and [31], these solutions refer to imovable targets with known and non-changing geometry. In general, the laser pointers of one vehicle do not have a specific distance offset with respect to the camera of the other vehicle, while at the same time their projections will not produce a constant pattern on the surface. Thus, we are dealing with a moving target of changing geometry. In fact, as it will be shown in the following analysis, the relative localization procedure is based exactly on the changing geometry of the laser dots on the surface.

All of the above led to the design and construction of a slightly different laser pointer configuration than proposed in [22]. This configuration is presented in the next section.

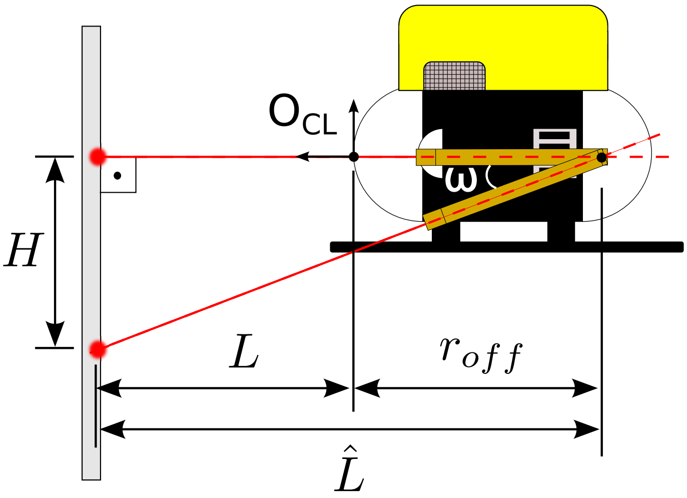


Figure 3.1: Laser configuration for a pair of laser pointers placed on one side of the Leader vehicle.

3.1.1 Laser Pointer Setup

The proposed configuration comprises two sets of laser pointers, one equipped on each vehicle. The setup equipped on the Leader comprises two pairs of red laser pointers, one at the port and the other at the starboard of the vehicle. In each pair, one laser is parallel to the longitudinal axis of the on-board camera and the other forms with the latter an acute angle ω as shown in Fig. 3.1. According to the same figure, the ranges of the Leader vehicle can be calculated as follows:

$$\begin{aligned} L_{L,L} &= \frac{H_L}{\tan \omega_L} - r_{off,L} \\ L_{R,L} &= \frac{H_R}{\tan \omega_R} - r_{off,R} \end{aligned} \tag{3.1}$$

where H_L, H_R denote the vertical distance between each pair of laser dots (Left and Right) projected from the Leader vehicle, ω_L, ω_R are the angles for each pair of laser pointers and $r_{off,L}, r_{off,R}$ denote specific constant offsets that are necessary for the calculation of the ranges as explained in Chapter 5. The angles ω_L and ω_R are usually chosen so that $\omega_L = \omega_R = \omega$.

On the other hand, the setup equipped on the Follower comprises two green laser pointers, both parallel to the axis of the camera and at specific distance with respect to each other. The ranges of the Follower can be calculated by the following relations:

$$\begin{aligned} L_{L,F} &= -\frac{d}{2} \frac{a_x}{(u_{ll} - u_0)} \\ L_{R,F} &= \frac{d}{2} \frac{a_x}{(u_{lr} - u_0)} \end{aligned} \tag{3.2}$$

where a_x and u_0 represent intrinsic parameters of the camera, d is the distance between the laser pointers and u_{ll}, u_{lr} denote the u coordinates of the laser dots inside the image frame of the vehicle. The distance from the surface (mean range) and the orientation (yaw angle) can be calculated for both vehicles by the next equations:

$$\begin{aligned} \psi &= \arctan\left(\frac{L_R - L_L}{d}\right) \\ L_m &= \frac{L_L + L_R}{2} \end{aligned} \tag{3.3}$$

The exact localization procedure and the calculation of the relative pose of one vehicle with respect to the other is described in more detail in Chapter 5.

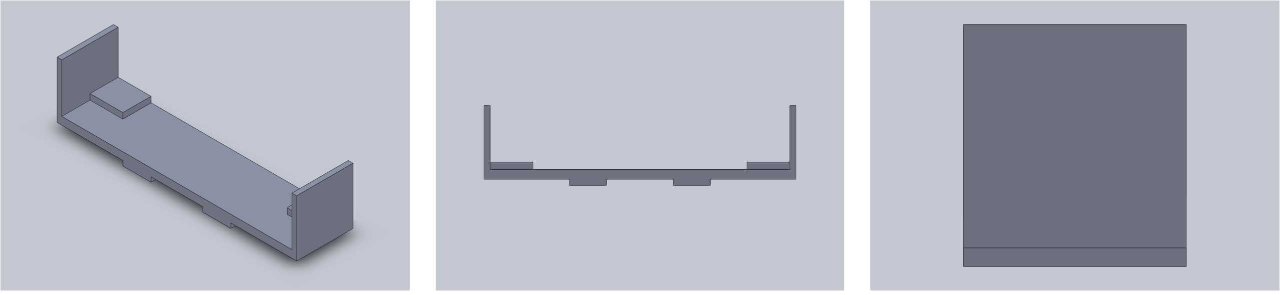


Figure 3.2: Leader laser configuration construction.

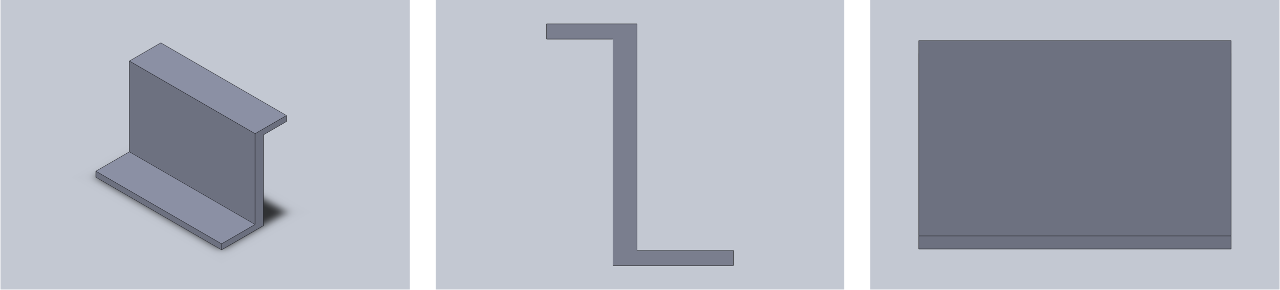


Figure 3.3: Follower laser configuration construction.

In order to equip the laser pointers on the vehicles, two separate constructions (one for each vehicle) were made. The construction for the Leader vehicle is shown in Fig. 3.2 and it comprises the bottom part of the laser configuration which hosts the inclided laser pointers. The construction for the Follower is shown in Fig. 3.3 . This construction is used twice; once to equip a laser pointer at the port of the vehicle and once to equip a laser pointer at the starboard. The exact schematics of both constructions are given in Appendix D, while they are also presented in Chapter 9.

3.1.2 Computer Vision Algorithm

According to Chapter 2, the implicit communications problem includes also the derivation of an appropriate computer vision algorithm for the detection of the laser dots produced on the flat surface by the laser setup. The proposed algorithm satisfies the Vision Requirements defined in the same chapter. It detects laser dots based on their intensity and categorises them as Leader laser dots or Follower laser dots based on their respective colour. The algorithm uses a filtering procedure to remove any noise from the image, a thresholding procedure to detect pixels that belong to laser dots and a labelling procedure to group these pixels together, while it categorises each group of pixels based on the mean values of their corresponding RGB triplets (see also Appendix B). Its functionality is described in more detail in Chapter 6.

3.2 Motion Control Scheme

According to Chapter 2, the cooperation of the vehicles is based on a Leader-Follower methodology. The Leader moves independently following a prescribed meander-like trajectory defined by waypoints, while the Follower maintains a specific distance offset with respect to the Leader and to the surface. The Leader does not take into account the movement of the Follower. Each control scheme is described in the next sections.

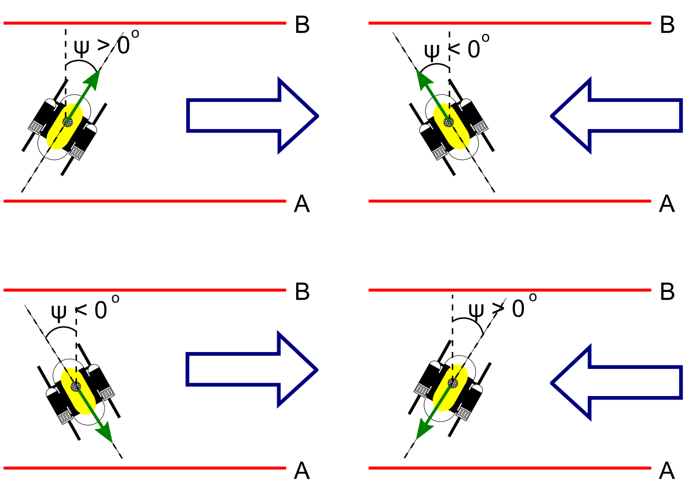


Figure 3.4: Saw-like motion primitives. The vehicle moves to the right, when it moves forwards with a positive angle or backwards with a negative angle (left side). The vehicle moves to the left, when it moves forwards with a negative angle or backwards with an positive angle (right side).

3.2.1 Leader Motion Control Scheme

According to Chapter 2 and Section 2.4, the Leader must move parallel to the surface while maintaining an orientation that is almost perpendicular. This requirement may lead to trajectories that involve motion of the vehicle along its lateral (sway) axis. The Leader is subject to nonholonomic constraints, since it is underactuated along its sway axis and, thus, this kind of motion is impossible.

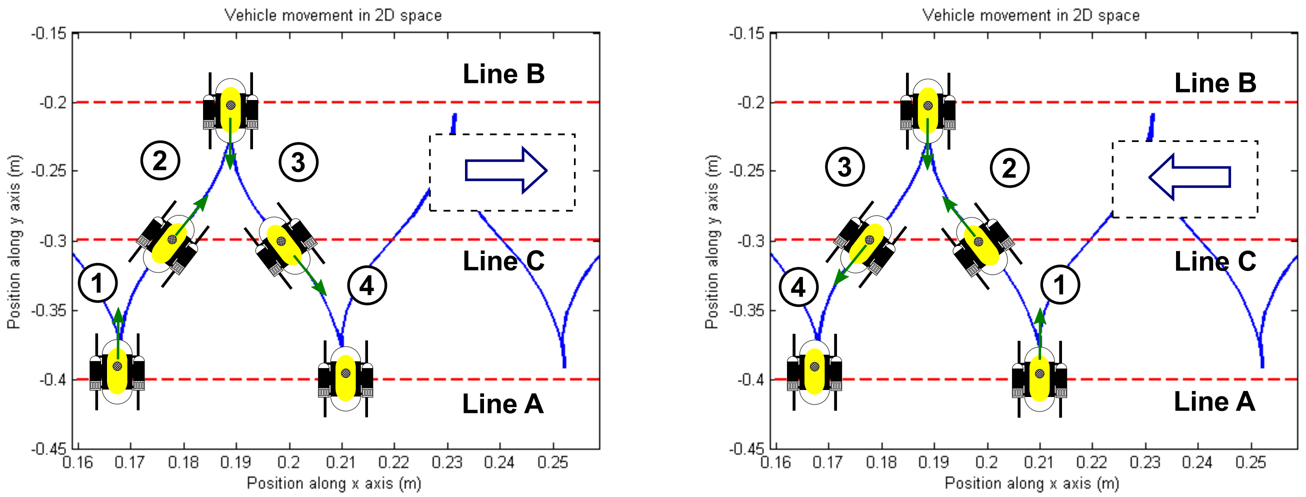
On the other hand, a saw-like motion would help the vehicle to move to any direction (left or right), while respecting at the same time the orientation constraint. In order to move to the right, the vehicle must have either a positive yaw angle while it moves forwards or a negative one while it moves backwards. Correspondingly, in order to move to the left, the vehicle must move either forwards with a negative yaw angle or backwards with a positive one. This kinds of motion are depicted in Fig. 3.4.

The proposed control scheme involves motion of the vehicle between two lines on the xy-plane, line A and line B (Fig. 3.5). A third line C is needed in order to fully control the vehicle's movement. The use of this line forces the vehicle to acquire the necessary positive or negative yaw angle in order to move towards the desired direction (left or right) and at the same time reach line A or line B with an angle of approximately 0°.

The controller comprises 4 modes that form a cycle. Each mode has the same controller but different desired states. The desired states refer only to the vertical distance of the vehicle from the surface and its orientation (yaw angle) and are denoted as x_d and ψ_d respectively.

The modes will be described briefly for the case of the motion to the right. During Mode 1, the vehicle must move towards line C and acquire a positive yaw angle ($\psi_d > 0$°). In Mode 2, the vehicle must move towards line B and acquire an orientation perpendicular to the surface ($\psi_d = 0$°). In Mode 3, the vehicle must move again towards line C, but the desired yaw angle is negative ($\psi_d < 0$°). Finally, in Mode 4, the vehicle must navigate towards line A with perpendicular orientation to the surface ($\psi = 0$°). It is clear that during modes 1 and 2 the vehicle travels forwards having a positive yaw angle and during modes 3 and 4 it travels backwards having a negative one. Thus, it is guaranteed that during the whole cycle the vehicle moves to the right (Fig. 3.5a) when no disturbances are present. The same observations apply also when the vehicle moves to the left, but in that case the vehicle executes the same motion in reverse as shown in Fig. 3.5b.

The corresponding surge (u_L) and yaw (r_L) control inputs for each individual mode is described



(a) Vehicle motion to the right.

(b) Vehicle motion to the left.

Figure 3.5: Control scheme for each direction of motion.

by the following equations:

$$u_L = -k_u \frac{x_L - x_d}{\sin \psi_L} \tag{3.4}$$

$$r_L = -k_r(\psi_L - \psi_d) \tag{3.5}$$

where k_u, k_v are positive constants and x_L, ψ_L denote the position along the x-axis and the yaw orienation of the vehicle with respect to the WCS. The above analysis refers to the motion of the vehicle on the xy-plane. The heave control input (w_L) for the motion along the z-axis is given simply by:

$$w_L = -k_w(z_L - z_d) \tag{3.6}$$

where k_w is a positive constant, z_L denotes the position of the vehicle along the z axis of the WCS and z_d denotes the corresponding desired position. This controller is used in this work only for simulation purposes. During the real experiments, though, the motion control along the z-axis is given instead by a simple PID controller, which is of the form:

$$w_L = -k_{pz}e_z - k_{dz}\dot{e}_z - k_{iz}\sum e_z dt \tag{3.7}$$

where e_z is the position error along the z-axis ($e_z = z_L - z_d$),$\dot{e}_z$ is the derivative of the error with respect to time, dt is the time interval between two successive measurements and k_{pz}, k_{dz}, k_{iz} are positive constants.

The above motion control scheme guarantees that the vehicle will be able to perform all the motions that are necessary in order to follow the meander-like trajectory, i.e. motion to the left, to the right, upwards and downwards. The control scheme (except from the PID controller) is based on a Lyapunov stability analysis and the kinematic equations of the vehicle ([38]) and it is analysed in more detail in Chapter 7.

3.2.2 Follower Motion Control Scheme

The Follower vehicle must maintain a specific distance offset with respect to the Leader, while the latter moves. In addition, there are certain cooperation constraints that must not be violated. These constraints have already been presented in Chapter 2 and are analysed in detail in Chapter

4. The most important constraint is that the Follower must maintain all laser dots (including its own) inside its image frame.

The problem statement for the motion control of the Follower vehicle can be divided into the following distinct goals:

- Keep a specific distance offset with respect to the surface along the x-axis of the GCS.
- Keep a specific distance offset with respect to the Leader along the y-axis of the GCS.
- The centre of the four Leader laser dots must remain near the centre of the Follower image frame.

The last goal guarantees that the Leader laser dots will not leave the image frame of the Follower. This is crucial for the control procedure of the Follower, since its only sensor input is based on machine vision and on the premise that all four laser dots of the Leader lie inside its image frame.

By examining the above goals, it is understood that the solution of the problem relates both to the Cartesian and to the image space. The part of the controller that satisfies the first two goals is analyzed and developed in the Cartesian space, while the part of the controller that satisfies the last goal is developed in the image space. Moreover, the first goal is related only to the positioning information provided by the Follower itself, while the other two goals relate also to the positioning information of the Leader. This leads to a Follower controller that relates both to position-based and image-based visual servoing and includes terms related to the motion of the Leader.

In general, it can be stated that the Follower controller drives the state vector **p** $= [{}^{G}x_F, {}^{G}y_F, u_{lc}, v_{lc}]^T$ to the desired state vector **p**$_d = [{}^{G}x_d, {}^{G}y_d, u_d, v_d]^T$, where the vectors **p**$_{xy} = [{}^{G}x_F, {}^{G}y_F]^T$ and **p**$_{xy,d} = [{}^{G}x_d, {}^{G}y_d]^T$ relate to the Cartesian space and denote the position coordinates of the vehicle along the x and y axes of the GCS and their desired values, respectively. The vectors **p**$_{uv} = [u_c, v_c]^T$ and **p**$_{uv,d} = [u_d, v_d]^T$ relate to the image space and denote the image space coordinates of the centre of the four Leader laser dots inside the image frame of the Follower and their corresponding desired values.

The vector **p**$_{xy}$ is driven to the vector **p**$_{xy,d}$ under the control inputs:

$$\begin{aligned} u_F &= -k_x({}^{G}x_F - {}^{G}x_d)\cos{}^{G}\psi_F - k_y({}^{G}y_F - {}^{G}y_d)\sin{}^{G}\psi_F + \dot{y}_L\sin{}^{G}\psi_F \\ v_F &= k_x({}^{G}x_F - {}^{G}x_d)\sin{}^{G}\psi_F - k_y({}^{G}y_F - {}^{G}y_d)\cos{}^{G}\psi_F + \dot{y}_L\cos{}^{G}\psi_F \end{aligned} \tag{3.8}$$

which are related to the Cartesian space. The coefficients k_x and k_y are positive constants. The term ${}^{G}\psi_F$ denotes the orientation of the Follower with respect to the GCS, while the term $\dot{y}_L$ denotes the velocity of the Leader vehicle along the y-axis of the GCS.

In order to drive vector **p**$_{uv}$ to the desired vector **p**$_{uv,d}$, the velocities of the image coordinates of the Leader laser centre must first be expressed with respect to the control input vector of the Follower (**u**$_F = [u_F, v_F, w_F, r_F]^T$ and the motion of the Leader. This leads to an expression of the form:

$$\begin{bmatrix} \dot{u}_c \\ \dot{v}_c \end{bmatrix} = M\mathbf{u}_F + N\dot{\mathbf{x}}_L \tag{3.9}$$

where the elements of the matrices M and N are calculated and presented in Chapter 7, while $\dot{\mathbf{x}}_L = [\dot{x}_L, \dot{y}_L, \dot{z}_L, \dot{\psi}_L]^T$ represents the velocities of the Leader with respect to the WCS. The corresponding control inputs that retain the Leader laser centre inside the Follower image frame are calculated as:

$$
\begin{aligned}
r_F &= \frac{-k_r(u_c - u_d) - m_1 u_F - m_2 v_F - n_1 \dot{x}_L - n_2 \dot{y}_L - n_4 \dot{\psi}_L}{m_4} \\
w_F &= \frac{-k_w(v_c - v_d) - m_5 u_F - m_8 r_F - n_5 \dot{x}_L - n_6 \dot{y}_L - n_7 \dot{z}_L - n_8 \dot{\psi}_L}{m_7}
\end{aligned}
\tag{3.10}
$$

where k_r, k_w are positive constants and the terms m_1, m_2, m_4, m_5, m_7, m_8, n_1, n_2, n_4, n_5, n_6, n_7, n_8 correspond to elements of the matrices M and N as explained in Chapter 7.

Chapter 4

State Constraints

This section presents the mathematical and geometrical analysis with regard to the state constraints that are present during the cooperation of the two vehicles. These state constraints can be divided into *vision constraints*, *collision constraints* and *line-of-sight blocking constraints*.

4.1 Vision Constraints

Vision constraints relate to the fact that a vehicle must not lose from its image plane the laser dots produced by the other vehicle in order to successfully localize itself with respect to the latter. In Fig. 4.1, the Follower vehicle localizes itself with respect to the Leader.

First, we examine the constraints imposed on the xy-plane. Based on Fig. 4.1a, the Follower will not lose Leader's laser dots from its image plane if and only if the following relation holds:

$$0 \leq \overline{AP} \leq \overline{AB} - \overline{PQ} \tag{4.1}$$

In the figure, the triplets (x_L, y_L, ψ_L) and (x_F, y_F, ψ_F) represent the state (x,y coordinates and orientation) of each vehicle on the xy-plane with respect to the GCS and the signs represent the fact that in geometry all values must be positive. Based on the same figure, $\overline{AP}$ can be expressed as:

$$\overline{AP} = \overline{AK} - \overline{MK} - \frac{\overline{PQ}}{2}$$

The segment $\overline{AK}$ can be calculated as:

$$\overline{AK} = -x_F \tan\left(\frac{\beta_H}{2} - \psi_F\right)$$

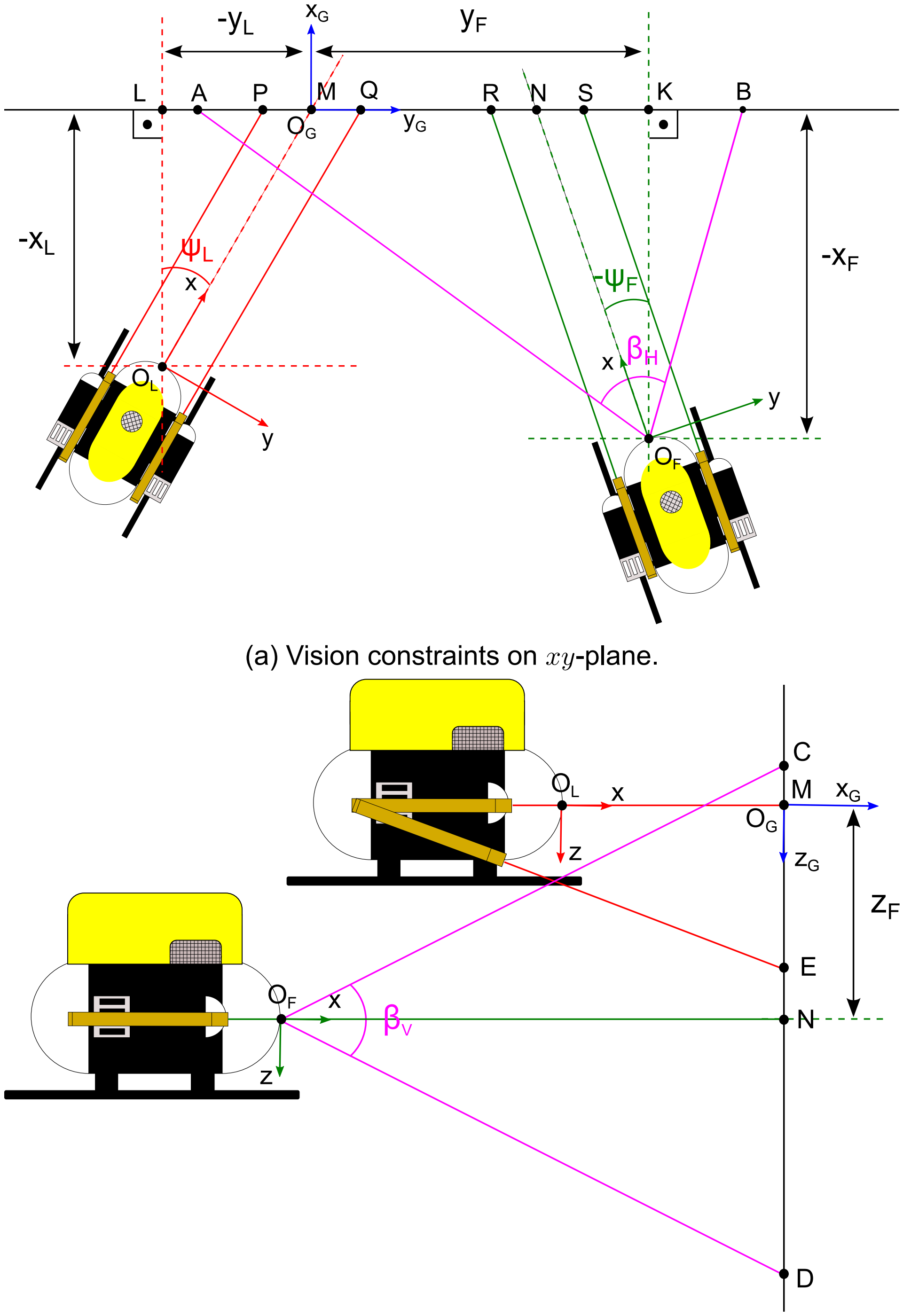
where β_H is the angle-of-view for the Follower on the horizontal (xy-) plane. The segment $\overline{MK}$ is equal to y_F and segment $\overline{PQ}$ can be expressed as:

$$\overline{PQ} = \frac{dl}{\cos\psi_L}$$

where dl is the distance between the laser pointers of the Leader. Finally, segment $\overline{AB}$, which represents the area that is visible to the Follower, can be calculated as follows:

$$\overline{AB} = \overline{AK} + \overline{KB} = -x_F \left[\tan\left(\frac{\beta_H}{2} - \psi_F\right) + \tan\left(\frac{\beta_H}{2} + \psi_F\right)\right]$$

Thus, Eq. 4.1 becomes:



(a) Vision constraints on xy-plane.

(b) Vision constraints on xz-plane.

Figure 4.1: Vision constraints due to relative localization procedure.

$$0 \leq -x_F \tan\left(\frac{\beta_H}{2} - \psi_F\right) - y_F - \frac{dl}{2\cos\psi_L} \leq$$
$$- x_F \left[\tan\left(\frac{\beta_H}{2} - \psi_F\right) + \tan\left(\frac{\beta_H}{2} + \psi_F\right)\right] - \frac{dl}{\cos\psi_L}$$

The last relation leads to two separate inequalities which represent the *Vision constraints on the xy-plane* and are the following:

$$y_F - x_F \tan\left(\psi_F - \frac{\beta_H}{2}\right) + \frac{dl}{2\cos\psi_L} \leq 0 \tag{4.2}$$
$$-y_F + x_F \tan\left(\psi_F + \frac{\beta_H}{2}\right) + \frac{dl}{2\cos\psi_L} \leq 0 \tag{4.3}$$

In Fig. 4.1b, we examine the constraints on the xz-plane. The Follower will not lose the Leader's laser dots from its image plane if:

$$0 \leq \overline{CM} \leq \overline{CD} - \overline{ME} \tag{4.4}$$

The segment $\overline{CM}$ can be written as:

$$\overline{CM} = \overline{CN} - \overline{MN} \Rightarrow$$
$$\overline{CM} = -x_F \tan\left(\frac{\beta_V}{2}\right) - z_F$$

where β_V is the angle-of-view for the Follower on the vertical (xz-) plane. Segment $\overline{CD}$ represents the visible area to the Follower along the vertical direction and is calculated as:

$$\overline{CD} = -2x_F \tan\left(\frac{\beta_V}{2}\right)$$

The Leader projects its laser beams onto the flat surface. For each pair of laser pointers (port, starboard), there is a vertical distance between the two laser dots of the pair. Segment $\overline{ME}$ represents the maximum of these distances. As already mentioned in Chapter 3, these distances can be expressed as H_L for the left pair and H_R for the right and they are calculated as:

$$H_L = [L_{L,L} + r_{off,L}] \tan\omega_L = \left[L_{m,L} - \frac{dl}{2}\tan\psi_L + r_{off,L}\right]\tan\omega_L = \left[\frac{-x_L}{\cos\psi_L} - \frac{dl}{2}\tan\psi_L + r_{off,L}\right]\tan\omega_L$$
$$H_R = [L_{R,L} + r_{off,R}] \tan\omega_R = \left[L_{m,L} + \frac{dl}{2}\tan\psi_L + r_{off,R}\right]\tan\omega_R = \left[\frac{-x_L}{\cos\psi_L} + \frac{dl}{2}\tan\psi_L + r_{off,R}\right]\tan\omega_R$$

where $L_{m,L}$ denotes the mean range of the Leader and dl denotes the distance between the two pairs of laser pointers equipped on the Leader. This leads to:

$$\overline{ME} = max\{H_L, H_R\}$$

Then, Eq. 4.4 becomes:

$$0 \leq -x_F \tan\left(\frac{\beta_V}{2}\right) - z_F \leq -2x_F \tan\left(\frac{\beta_V}{2}\right) - max\{H_L, H_R\}$$

This leads to two separate inequalities that represent the *Vision constraints on the xz-plane* and are expressed as follows:

$$x_F \tan\left(\frac{\beta_V}{2}\right) + z_F \leq 0 \tag{4.5}$$

$$x_F \tan\left(\frac{\beta_V}{2}\right) - z_F + max\{H_L, H_R\} \leq 0 \tag{4.6}$$

$$H_L = \left[\frac{-x_L}{\cos\psi_L} - \frac{dl}{2}\tan\psi_L + r_{off,L}\right]\tan\omega_L \tag{4.7}$$

$$H_R = \left[\frac{-x_L}{\cos\psi_L} + \frac{dl}{2}\tan\psi_L + r_{off,R}\right]\tan\omega_R \tag{4.8}$$

4.2 Collision Constraints

Collision constraints refer to those constraints that must not be violated because otherwise the two vehicles would collide with each other. Let the pairs (l_L, w_L) and (l_F, w_F) denote the dimensions (length and width) of the Leader and the Follower respectively. If each of these two pairs creates a rectangle, let R denote the radius of the circle that is circumscribed around the rectangle (R_L for the Leader and R_F for the Follower) as depicted in Fig. 4.2a.

In Fig. 4.2, let D denote the distance between the camera centres of the vehicles and D_{cm} the distance between their respective centres of mass. The collision constraint can then be expressed as:

$$D_{cm} \geq R_L + R_F \tag{4.9}$$

A more clear view of the geometry of the problem is shown in Fig. 4.2b where lines ε_1 and ε_2 are parallel with each other and thus the angle between segment $\overline{AB}$ and line ε_2 is equal to $\psi_L - \psi_F$. Once again, the triplets (x_L, y_L, ψ_L) and (x_F, y_F, ψ_F) represent the states of the vehicles. The distance D can be easily calculated as:

$$D = \sqrt{(x_L - x_F)^2 + (y_L - y_F)^2}$$

while the angle λ can be computed as:

$$\tan\lambda = \frac{-x_F + x_L}{y_F - y_L} \Rightarrow \lambda = \tan^{-1}\left(\frac{x_L - x_F}{y_F - y_L}\right)$$

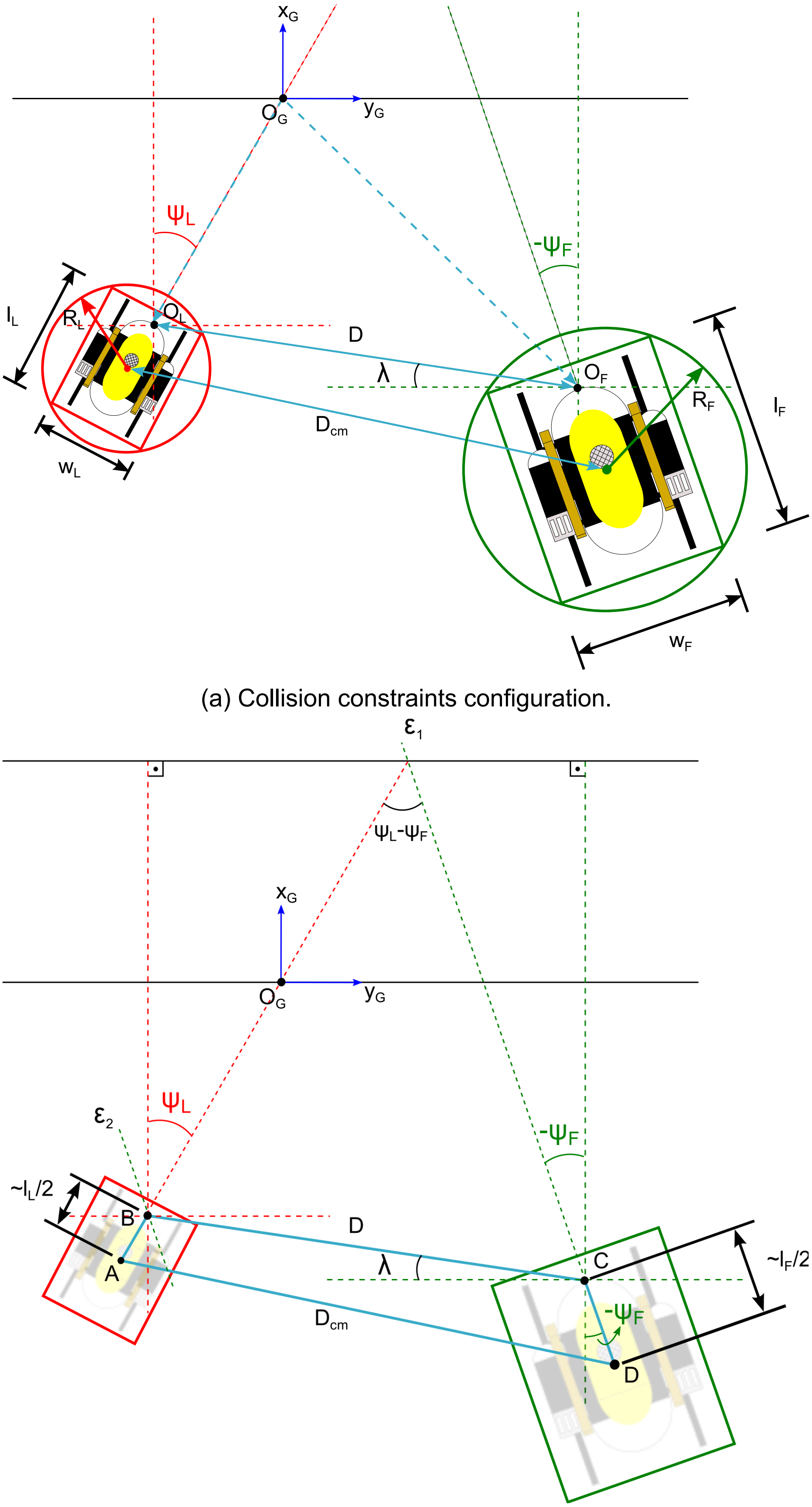
In the quadrilateral $(ABCD)$, which is resketched for more clarity in Fig. 4.3, it is observed that the angle $\widehat{BCL}$ in the triangle $\triangle BCL$ is equal to:

$$\widehat{BCL} = \pi - (-\psi_F + \frac{\pi}{2} + \lambda) = \frac{\pi}{2} + \psi_F - \lambda$$

Then, segments $\overline{BL}$ and $\overline{CL}$ are each equal to:

$$\overline{BL} = D\sin\left(\frac{\pi}{2} + \psi_F - \lambda\right) = D\cos(\psi_F - \lambda)$$
$$\overline{CL} = D\cos\left(\frac{\pi}{2} + \psi_F - \lambda\right) = -D\sin(\psi_F - \lambda)$$

In triangle $\triangle ABE$ segments $\overline{AE}$ and $\overline{BE}$ are equal to:



(a) Collision constraints configuration.

(b) Collision constraints simplified configuration.

Figure 4.2: Collision constraints.

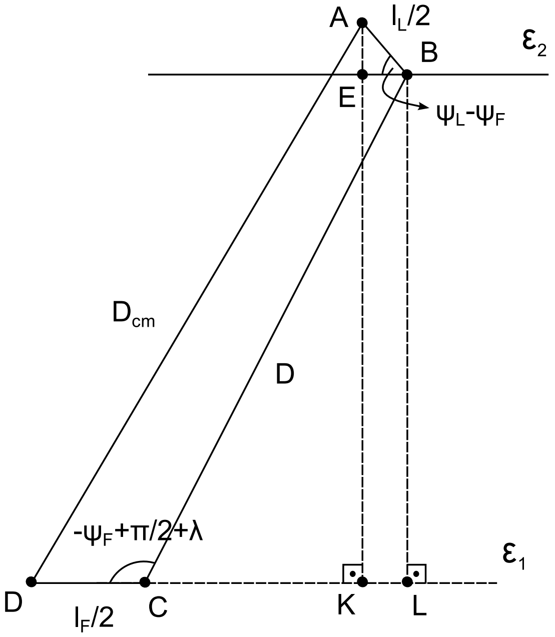


Figure 4.3: Collision constraints geometry.

$$\overline{AE} = \frac{l_L}{2}\sin(\psi_L-\psi_F)$$
$$\overline{BE} = \frac{l_L}{2}\cos(\psi_L-\psi_F)$$

Segments $\overline{AK}$ and $\overline{DK}$ are equal to:

$$\overline{AK} = \overline{AE}+\overline{BL} = \frac{l_L}{2}\sin(\psi_L-\psi_F) + D\cos(\psi_F-\lambda)$$
$$\overline{DK} = \overline{DC}+\overline{CL}-\overline{BE} = \frac{l_F}{2} - D\sin(\psi_F-\lambda) - \frac{l_L}{2}\cos(\psi_L-\psi_F)$$

which leads to the calculation of D_{cm} as follows:

$$D_{cm} = \sqrt{\overline{AK}^2+\overline{DK}^2} \Rightarrow$$
$$D_{cm} = \sqrt{\frac{l_L^2+l_F^2}{4} + D^2 + l_L D\sin(\psi_L-\lambda) - l_F D\sin(\psi_F-\lambda) - \frac{l_L l_F}{2}\cos(\psi_L-\psi_F)}$$

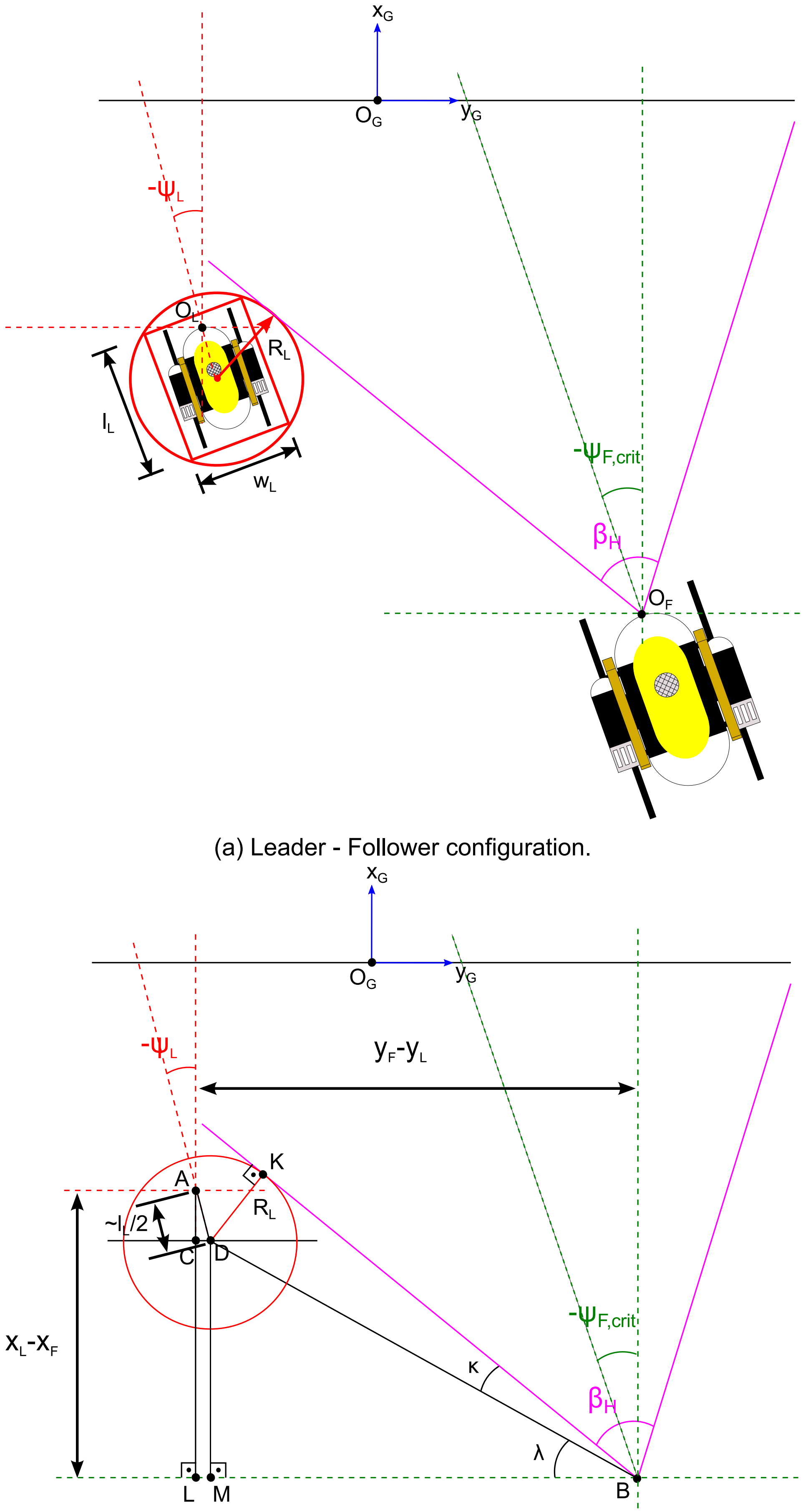
Finally, Eq. 4.9 becomes:

$$R_L + R_F - \sqrt{\frac{l_L^2+l_F^2}{4} + D^2 + l_L D\sin(\psi_L-\lambda) - l_F D\sin(\psi_F-\lambda) - \frac{l_L l_F}{2}\cos(\psi_L-\psi_F)} \leq 0 \quad (4.10)$$

4.3 Line-of-Sight Blocking Constraints

During their cooperation, the vehicles must not intrude the vision space of one another. This is shown in Fig. 4.4a where the Leader is only marginally outside the angle-of-view of the Follower. If we assume that at this time Follower has a critical orientation, namely a critical yaw angle $\psi_{F,crit}$, then it must not exceed this angle and the corresponding constraint is written as follows:

$$\psi_F \leq \psi_{F,crit} \quad (4.11)$$



(a) Leader - Follower configuration.

(b) Problem geometry.

Figure 4.4: Line-of-Sight constraints.

In order to calculate $\psi_{F,crit}$, we must examine the geometry of the problem which is depicted in Fig. 4.4b. As already shown in this figure:

$$\overline{AL} = x_L - x_F$$
$$\overline{BL} = y_F - y_L$$

In triangle $\triangle ACD$, segments $\overline{AC}$ and $\overline{CD}$ can be calculated as:

$$\overline{AC} = \frac{l_L}{2}\cos(-\psi_L) = \frac{l_L}{2}\cos\psi_L$$
$$\overline{CD} = \frac{l_L}{2}\sin(-\psi_L) = -\frac{l_L}{2}\sin\psi_L$$

Thus, segments $\overline{DM}$ and $\overline{BM}$ are written as:

$$\overline{DM} = \overline{AL} - \overline{AC} = x_L - x_F - \frac{l_L}{2}\cos\psi_L$$
$$\overline{BM} = \overline{BL} - \overline{CD} = y_F - y_L + \frac{l_L}{2}\sin\psi_L$$

The angle λ is calculated then as follows:

$$\tan\lambda = \frac{\overline{DM}}{\overline{BM}} = \frac{x_L - x_F - \frac{l_L}{2}\cos\psi_L}{y_F - y_L + \frac{l_L}{2}\sin\psi_L} \Rightarrow$$
$$\lambda = \tan^{-1}\left(\frac{x_L - x_F - \frac{l_L}{2}\cos\psi_L}{y_F - y_L + \frac{l_L}{2}\sin\psi_L}\right)$$

while segment $\overline{BD}$ is written as:

$$\overline{BD} = \sqrt{\overline{DM}^2 + \overline{BM}^2} \Rightarrow$$
$$\overline{BD} = \sqrt{\frac{l_L^2}{4} + (x_L - x_F)^2 + (y_L - y_F)^2 - l_L(x_L\cos\psi_L - x_F\cos\psi_L + y_L\sin\psi_L - y_F\sin\psi_L)}$$

and angle κ is calculated as:

$$\sin\kappa = \frac{\overline{KD}}{\overline{BD}} = \frac{R_L}{\sqrt{\frac{l_L^2}{4} + (x_L - x_F)^2 + (y_L - y_F)^2 - l_L(x_L\cos\psi_L - x_F\cos\psi_L + y_L\sin\psi_L - y_F\sin\psi_L)}} \Rightarrow$$
$$\kappa = \sin^{-1}\left(\frac{R_L}{\sqrt{\frac{l_L^2}{4} + (x_L - x_F)^2 + (y_L - y_F)^2 - l_L(x_L\cos\psi_L - x_F\cos\psi_L + y_L\sin\psi_L - y_F\sin\psi_L}}\right)$$

Finally, it stands that:

$$\frac{\pi}{2} = -\psi_{F,crit} + \frac{\beta_H}{2} + \kappa + \lambda \Rightarrow$$
$$\psi_{F,crit} = \frac{\beta_H}{2} + \kappa + \lambda - \frac{\pi}{2} \tag{4.12}$$

and thus Eq. 4.11 becomes:

$$\psi_F \le \frac{\beta_H}{2} + \kappa + \lambda - \frac{\pi}{2} \Rightarrow$$
$$\psi_F - \frac{\beta_H}{2} - \kappa - \lambda + \frac{\pi}{2} \le 0 \tag{4.13}$$

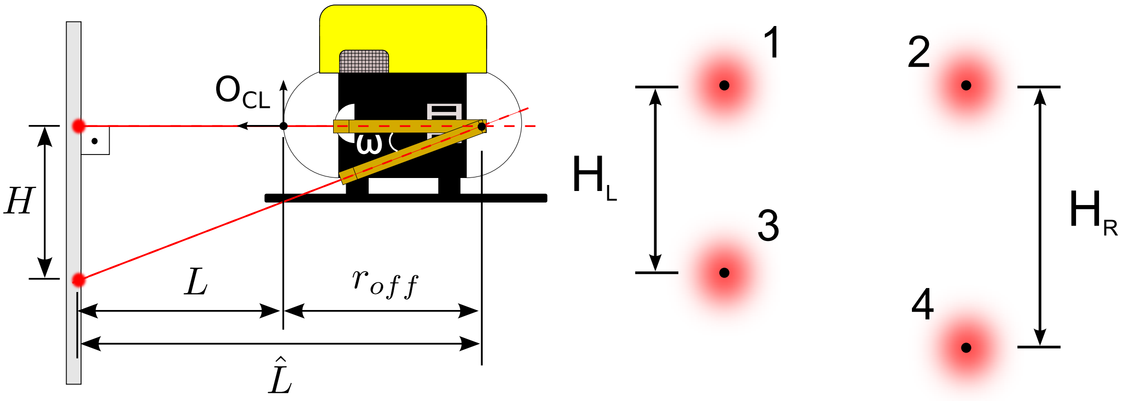
Chapter 5

Localization Procedure

This section describes the methodology for the relative localization between two underwater vehicles operating in a cooperative inspection task. Using computer vision algorithms and general projection perspective models, each vehicle calculates its range and bearing with respect to the inspected surface, as well as its pose vector with respect to the other vehicle. The calculations are strictly decentralized (no pose or other information are exchanged) and based only on the projection of the laser pointers on the inspected surface. In this work, we will examine the case where only the Follower vehicle calculates its relative position with respect to the Leader.

Each ROV is equipped with a set of laser pointers that project distinct dots onto the flat surface. Each set has different color and different configuration. The Leader set comprises four red laser pointers, divided into two pairs. One pair is mounted at the port and the other at the starboard of the vehicle. In each of these pairs, one laser pointer is parallel to the onboard camera axis (which coincides to the vehicle's longitudinal axis), while the other forms with the latter an acute angle ω as shown in Fig. 5.1a. The Follower is equipped with two green laser pointers, one mounted at the port and the other at the starboard, both parallel to the vehicle's camera axis.

Each vehicle calculates its distance and orientation with respect to the surface using its own laser pointers. For the Leader, only the pointers that are parallel to the camera axis are used for this purpose. This is done by considering the camera model, its intrinsic parameters and the fact that the laser configuration with respect to the camera origin of each vehicle is known. Let the position of a laser dot with respect to the camera coordinate system be expressed as $[x_s, y_s, z_s]^T$. Considering a linear perspective projection model (pin-hole camera model) we have the following result:



(a) Laser configuration for a pair of laser pointers placed on one side of the Leader vehicle.

(b) The laser dots configuration on the surface.

Figure 5.1: Leader laser set configuration.

$$\lambda\begin{bmatrix}u_c\\v_c\\1\end{bmatrix}=\begin{bmatrix}a_x & s & u_0 & 0\\0 & -a_y & v_0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x_s\\z_s\\y_s\\1\end{bmatrix} \tag{5.1}$$

where

$$a_x = fk_x$$
$$a_y = fk_y$$

while $[u_c, v_c]^T$ are the coordinates (in pixels) of a laser dot center inside the image frame, $[u_0, v_0]^T$ are the coordinates of the principal point inside the image frame, s is the skew factor, f the camera's focal length and k_x, k_y the scaling factors along the u (horizontal) and v (vertical) image axes respectively.

Assuming that $s = 0$, Eq. 5.1 is also written as:

$$\begin{aligned}u_c &= a_x\frac{x_s}{y_s} + u_0 \Rightarrow x_s = \frac{(u_c - u_0)}{a_x}y_s\\ v_c &= -a_y\frac{z_s}{y_s} + v_0 \Rightarrow z_s = -\frac{(v_c - v_0)}{a_y}y_s\end{aligned} \tag{5.2}$$

where the minus sign in the expression for z_s is explained by the fact that the z-axis of the FcCS has opposite direction to the v-axis of the image frame (see also Appendix B Section B.2). Regarding the horizontal lasers, y_s represents each laser range L and x_s denotes the known distance of each laser with respect to the camera center along the lateral axis of the vehicle. The lasers are placed symmetrically with respect to the camera axis and at a fixed distance d to each other. Thus, $x_s = \pm\frac{d}{2}$. Under these assumptions, the laser ranges for each vehicle are calculated as follows:

$$\begin{aligned}L_L &= -\frac{d}{2}\frac{a_x}{(u_{ll} - u_0)}\\ L_R &= \frac{d}{2}\frac{a_x}{(u_{lr} - u_0)}\end{aligned} \tag{5.3}$$

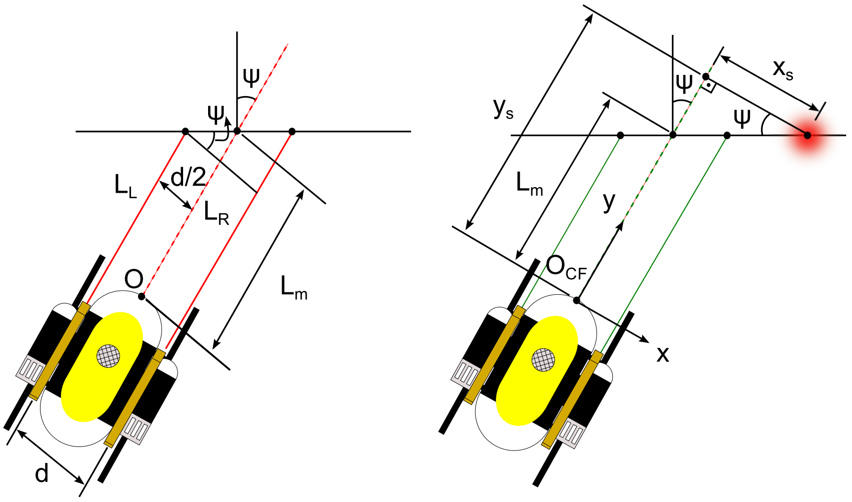
where L_L, L_R refer to the left and right laser ranges respectively and u_{ll}, u_{lr} denote the u image coordinates of the left and right laser dots.

For each vehicle, its distance L_m (range) and its orientation ψ (bearing) with respect to the surface are calculated as (Fig. 5.2a):

$$\begin{aligned}\psi &= \arctan\left(\frac{L_R - L_L}{d}\right)\\ L_m &= L_L + \frac{d}{2}\tan\psi \Rightarrow L_m = \frac{L_L + L_R}{2}\end{aligned} \tag{5.4}$$

The Follower calculates the distance $L_{m,L}$ and the orientation ψ_L of the Leader with respect to the surface. Let $[x_{lk}, y_{lk}, z_{lk}]^T, k = 1\ldots4$ (Fig. 5.1b) denote the coordinates of each laser dot on the surface projected by the Leader with respect to the camera coordinate system of the Follower (FcCS). They are calculated as follows:

$$\begin{aligned}x_{lk} &= \frac{(u_{lk,L} - u_{0,F})L_{m,F}}{a_{x,F} - (u_{lk,L} - u_{0,F})\tan\psi_F}\\ y_{lk} &= L_{m,F} + x_{lk}\tan\psi_F\\ z_{lk} &= -\frac{(v_{lk,L} - v_{0,F})y_{lk}}{a_{y,F}}\end{aligned} \tag{5.5}$$



(a) Calculation of distance L_m and orientation ψ.

(b) Calculation of the laser spot coordinates with respect to camera coordinate system.

Figure 5.2: Localization with respect to a laser spot.

where $[u_{lk,F}, v_{lk,F}]^T$ denote the image coordinates of the Leader's k laser dot inside the Follower's image frame, while the subscripts L and F refer to sizes that correspond to the Leader and the Follower, respectively. The above equations are based on Eq. 5.2 where the vector $[x_s, y_s, z_s]^T$ is substituted by the vector $[x_{lk}, y_{lk}, z_{lk}]^T$. We calculate y_{lk} using Fig. 5.2b. Then we substitute y_{lk} in the expression of x_{lk} and we solve for x_{lk}. The expression for z_{lk} remains as it is.

The laser ranges $L_{L,L}$, $L_{R,L}$ are calculated by the Follower based on the vertical distances H_L and H_R (Fig. 5.1b) between the dots of each pair of laser pointers (Left and Right respectively) using the right triangle of Fig. 5.1a. The ranges of the Leader are calculated with respect to the origin of its camera but, for each pair of pointers, the point where the two axes of the pointers intersect lies behind the camera origin. Thus, from each calculated range, an appropriate offset must be substracted. Let $r_{off,L}$ and $r_{off,R}$ denote these offsets for the left and the right pair respectively. Then, based on Fig. 5.1a, the ranges are calculated as follows:

$$\begin{aligned} L_{L,L} &= \hat{L}_{L,L} - r_{off,L} = \frac{H_L}{\tan\omega_L} - r_{off,L} = \frac{z_{l1} - z_{l3}}{\tan\omega_L} - r_{off,L} \\ L_{R,L} &= \hat{L}_{R,L} - r_{off,R} = \frac{H_R}{\tan\omega_R} - r_{off,R} = \frac{z_{l2} - z_{l4}}{\tan\omega_R} - r_{off,R} \end{aligned} \tag{5.6}$$

The distance $L_{m,L}$ and the orientation ψ_L are then calculated by the Follower as in Eq. 5.4. The Follower calculates the position vector $\mathbf{^G n_L} = [^G x_L, ^G y_L, ^G z_L]^T$ of the Leader with respect to the origin O_G as follows:

$$\begin{aligned} ^G x_L &= -L_{m,L} \cos\psi_L \\ ^G y_L &= -L_{m,L} \sin\psi_L \\ ^G z_L &= c_z \end{aligned} \tag{5.7}$$

The coordinate $^G z_L$ may be considered as equal to zero due to the definition of origin O_G and the fact that $\theta = 0$. In reality though, the coordinate $^G z_L$ is equal to a constant value $c_z =^L b_{zTL} =^L b_{zTR}$ that represents the vertical distance offset at which the parallel laser pointers are placed with respect to the camera origin. This is depicted in more detail in Chapter 7 Fig. 7.6.

The Follower calculates the vector $\mathbf{^{cF} n_M} = [x_M, y_M, z_M]^T$ which corresponds to the origin O_G of the GCS as expressed with respect to the Follower camera coordinate system. It resides in the

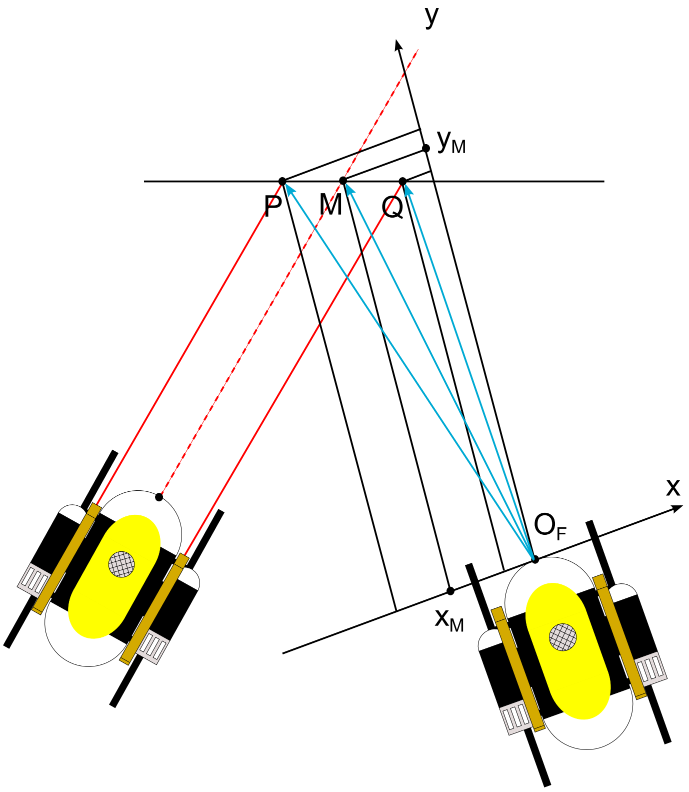


Figure 5.3: Localization of the Follower with respect to the origin O_G.

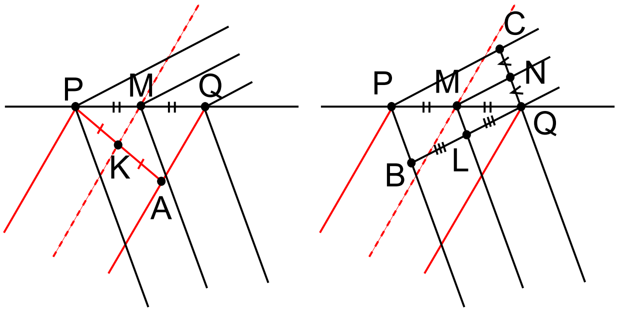


Figure 5.4: Triangle similarities and segment equalities.

middle of the two upper laser dots of the Leader produced by its horizontal laser pointers. The coordinates $[x_{l1}, y_{l1}, z_{l1}]^T$ and $[x_{l2}, y_{l2}, z_{l2}]^T$ are already known (see Eq. 5.5) and denoted in Fig. 5.3 as $\overrightarrow{O_{CF}P}$ and $\overrightarrow{O_{CF}Q}$ respectively.

In Fig. 5.4 (left) segments $\overline{AK}$ and $\overline{KP}$ denote the distance of each laser pointer axis from the longitudinal axis of the vehicle ($\overline{AK} = \overline{KP} = d/2$). The triangles $\triangle PAQ$ and $\triangle PMK$ are similar and due to the equality of segments $\overline{AK}$ and $\overline{KP}$, segments $\overline{PM}$ and $\overline{MQ}$ are also equal. In Fig. 5.4 (right), triangle $\triangle QBP$ is similar to $\triangle QML$ and triangle $\triangle QCP$ similar to $\triangle QNM$. Thus, since $\overline{PM} = \overline{MQ}$, segment $\overline{BL}$ is equal to $\overline{LQ}$ and segment $\overline{CN}$ is equal to $\overline{NQ}$. Segments $\overline{BL}$ and $\overline{LQ}$ match to corresponding intervals on the x axis of the Follower camera coordinate system as depicted in Fig. 5.3. Segments $\overline{CN}$ and $\overline{NQ}$ correspond to intervals on the y axis of the same coordinate system. Thus:

$$x_M = \frac{x_{l1} + x_{l2}}{2}$$
$$y_M = \frac{y_{l1} + y_{l2}}{2}$$
$$z_M = z_{l1} = z_{l2}$$

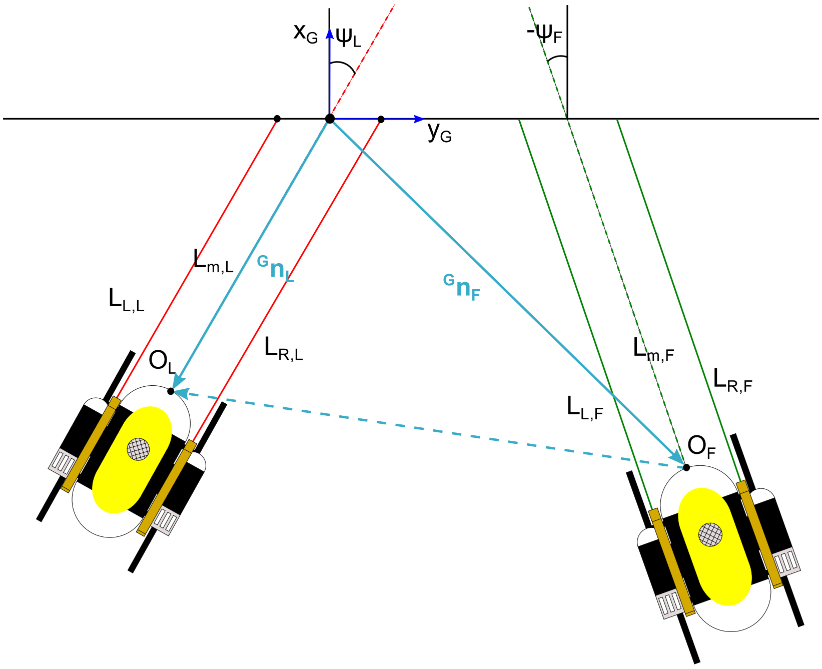


Figure 5.5: The localization concept between two cooperating vehicles.

Finally, the Follower calculates its position with respect to the GCS. In order to do so, it firstly expresses the vector $[x_M, y_M, z_M]^T$ with respect to the FCS using Eq. 2.2, then it expresses its position with respect to a coordinate system parallel to FCS but with origin on O_G and, finally, it applies a rotation matrix. This leads to the calculation of $\mathbf{^G n_F} = [^G x_F, ^G y_F, ^G z_F]^T$ as follows:

$$\begin{aligned} ^G x_F &= x_M \cos\psi - y_M \sin\psi \\ ^G y_F &= -x_M \sin\psi - y_M \cos\psi \\ ^G z_F &= z_M \end{aligned} \tag{5.8}$$

The Follower uses the vectors $\mathbf{^G n_L}$ and $\mathbf{^G n_F}$ in order to maintain specific distance offsets from the Leader and the surface with respect to the GCS. The localization of the Leader with respect to the Follower is depicted in Fig. 5.5.

Chapter 6

Computer Vision Algorithm

As explained in the previous section, we examine the case where only the Follower calculates its relative position with respect to the Leader. Nevertheless, the proposed algorithm can be also used for the inverse case under some minor modifications. During the inspection task, the Follower must localize itself with respect to the Leader. In order to do so, the Follower must detect its own laser dots as well as the Leader's laser dots inside its image frame. This leads to the definition of the *Vision Problem* as follows:

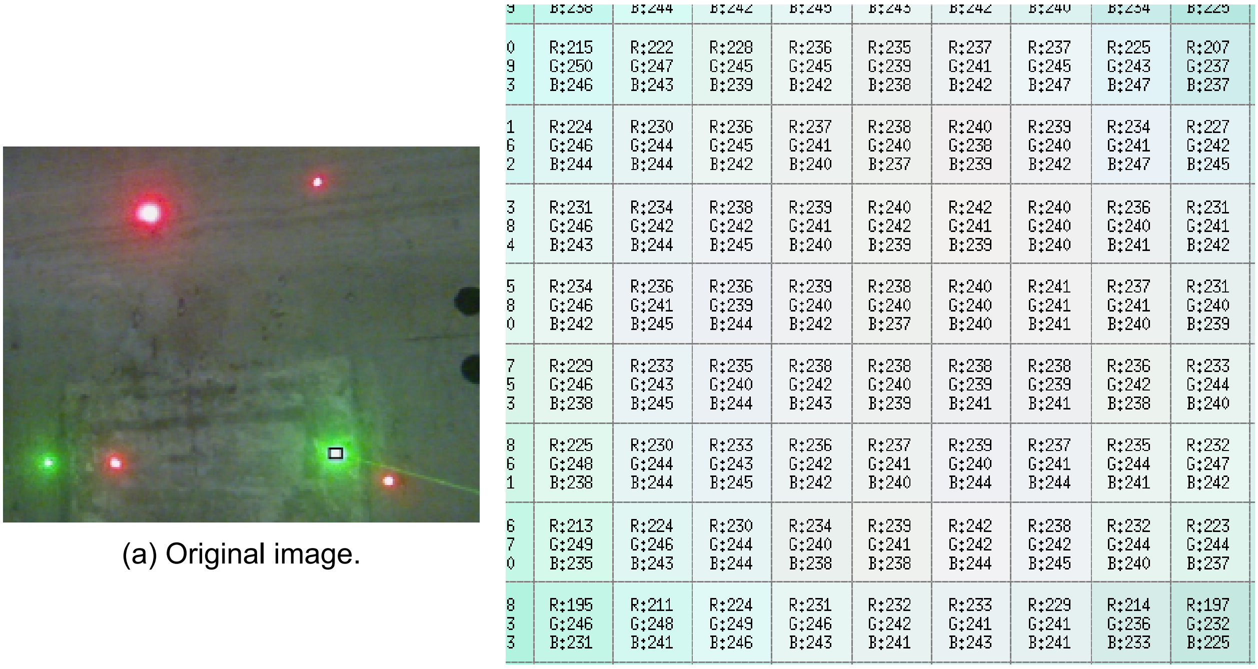
Detect six (6) laser dots inside the image frame, four (4) of which are red and the other two (2) are green, accurately, consistently and at a high frequency.

The word ``*accurately*'' refers to the fact that the algorithm must detect the centre of each laser dot in the image frame with a certain accuracy, while at the same time it must distinguish correctly between red laser dots and green ones. The *consistency* relates to the fact that the algorithm must produce accurate results under a broad range of lighting conditions, while avoiding to produce false positives. Finally, the results can be produced in *high frequency* only if the execution time of the algorithm is small. This means that the algorithm must be as simple as possible (see also *Vision Requirements*).

6.1 Vision Algorithm Design Procedure

In order to address the *Vision Problem*, certain aspects of the vision system were taken into consideration. The cameras were equipped on the vehicles and, thus, they were not stationary. This excluded any approach that used background substraction techniques, since in those methods the algorithm detects an object by subtracting from the current image a stationary background. It was also not possible to use any techniques that would rely on specific patterns pertinent to the laser dots. Such techniques could be related to template matching or searching in the image for circles using the Hough transform. These methods do not produce any useful results because laser dots do not appear the same in every image (different lighting conditions, different angle) and so, no explicit template can be matched to them. Moreover, based on their intensity and the angle from where they are observed, laser dots may have a more random shape than a circular one. This is the reason why Circular Hough Tranform fails most of the time to detect such objects.

On the other hand, laser dots are dots of light with high intensity, while the laser dots in this work have also a specific and unique colour; red for the Leader vehicle and green for the Follower. Another interesting aspect of the vision system is the fact that, due to their high intensity, green laser dots are also depicted fairly well in the red channel of the RGB colour space. In fact, it is observed that the centres of the green dots contain pixels that have high values of the red component in the RGB colour space. This is due to the fact that the high intensity of the green



(a) Original image.

9	B:238	B:244	B:242	B:245	B:245	B:242	B:240	B:234	B:225
0 9 3	R:215 G:250 B:246	R:222 G:247 B:243	R:228 G:245 B:239	R:236 G:245 B:242	R:235 G:239 B:238	R:237 G:241 B:242	R:237 G:245 B:247	R:225 G:243 B:247	R:207 G:237 B:237
1 6 2	R:224 G:246 B:244	R:230 G:244 B:244	R:236 G:245 B:242	R:237 G:241 B:240	R:238 G:240 B:237	R:240 G:238 B:239	R:239 G:240 B:242	R:234 G:241 B:247	R:227 G:242 B:245
3 8 4	R:231 G:246 B:243	R:234 G:242 B:244	R:238 G:242 B:245	R:239 G:241 B:240	R:240 G:242 B:239	R:242 G:241 B:239	R:240 G:240 B:240	R:236 G:240 B:241	R:231 G:241 B:242
5 8 0	R:234 G:246 B:242	R:236 G:241 B:245	R:236 G:239 B:244	R:239 G:240 B:242	R:238 G:240 B:237	R:240 G:240 B:240	R:241 G:241 B:241	R:237 G:241 B:240	R:231 G:240 B:239
7 5 3	R:229 G:246 B:238	R:233 G:243 B:245	R:235 G:240 B:244	R:238 G:242 B:243	R:238 G:240 B:239	R:238 G:239 B:241	R:238 G:239 B:241	R:236 G:242 B:238	R:233 G:244 B:240
8 6 1	R:225 G:248 B:238	R:230 G:244 B:244	R:233 G:243 B:245	R:236 G:242 B:242	R:237 G:241 B:240	R:239 G:240 B:244	R:237 G:241 B:244	R:235 G:244 B:241	R:232 G:247 B:242
6 7 0	R:213 G:249 B:235	R:224 G:246 B:243	R:230 G:244 B:244	R:234 G:240 B:238	R:239 G:241 B:238	R:242 G:242 B:244	R:238 G:242 B:245	R:232 G:244 B:240	R:223 G:244 B:237
8 3 3	R:195 G:246 B:231	R:211 G:248 B:241	R:224 G:249 B:246	R:231 G:246 B:243	R:232 G:242 B:241	R:233 G:241 B:243	R:229 G:241 B:241	R:214 G:236 B:233	R:197 G:232 B:225

(b) RGB components of green laser dot centre region.

Figure 6.1: The intensity values of the RGB components for the centre of a green laser dot. The region enclosed in the small rectangle (bottom right of the left figure) corresponds to the RGB values shown in the right figure.

laser dots is expressed as white colour in the image where all three RGB components have values in the range of $230 - 255$ (Fig. 6.1b).

Based on the above observations, a vision algorithm was designed and implemented. The algorithm was first designed and tested in the Matlab environment using still images and then it was implemented in C/C++ using the OpenCV library. The vision algorithm comprises three steps:

- Laser Detection
- Colour Detection
- Final Selection

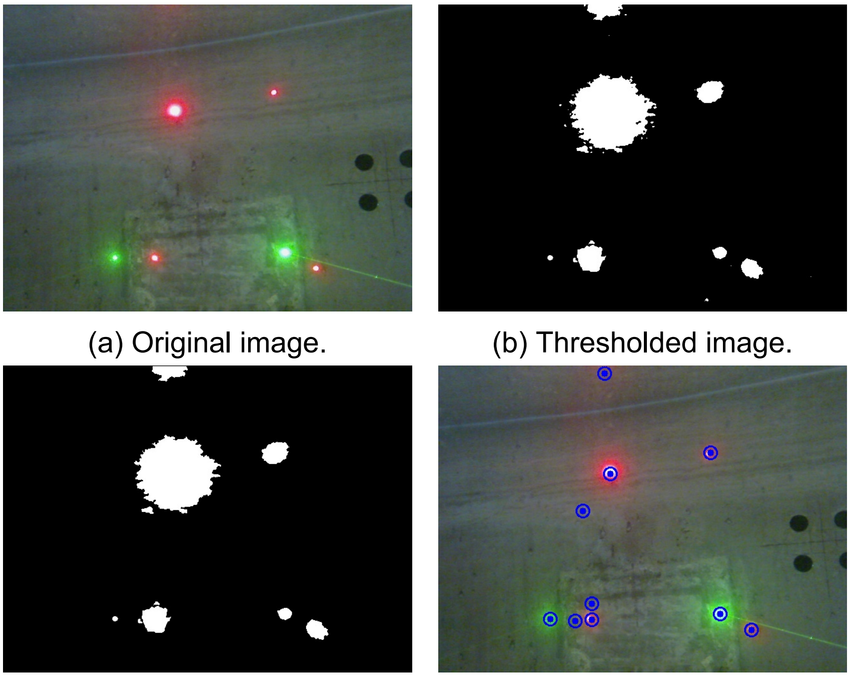
Each step will be described in detail in the next section.

6.2 Vision Algorithm Procedure

6.2.1 Laser Detection

The original image is first pre-processed using a smoothing filter to remove any noise produced by the camera sensor. The image is expessed in the RGB colour space (Fig. 6.2a). The red channel of the image is thresholded to produce a black and white image (Fig. 6.2b), where white pixels, i.e. those pixels that passed the threshold value, represent possible laser dots. The red channel was chosen because it is the only channel where both types of laser dots are adequately visible, taking also into consideration the above observation about green laser dots.

A blobing procedure (Fig. 6.2c) groups together adjacent white pixels and assigns to them properties such as blob centre and blob size. Groups of such pixels that are too small to represent laser dots are excluded from the subsequent steps of the processing. This last step removes any noise that originates from the segmentation process (i.e. the thresholding process). The result is



(a) Original image.

(b) Thresholded image.

(c) Thresholded image after blobing and removal of small sized blobs.

(d) Centres of possible laser dots detected in the image.

Figure 6.2: The laser detection operation.

the detection of areas in the image that may represent real laser dots based only on the intensity of these areas (Fig. 6.2d). The step that is described in the next section finalizes the detection procedure and divides laser dots based on their colour.

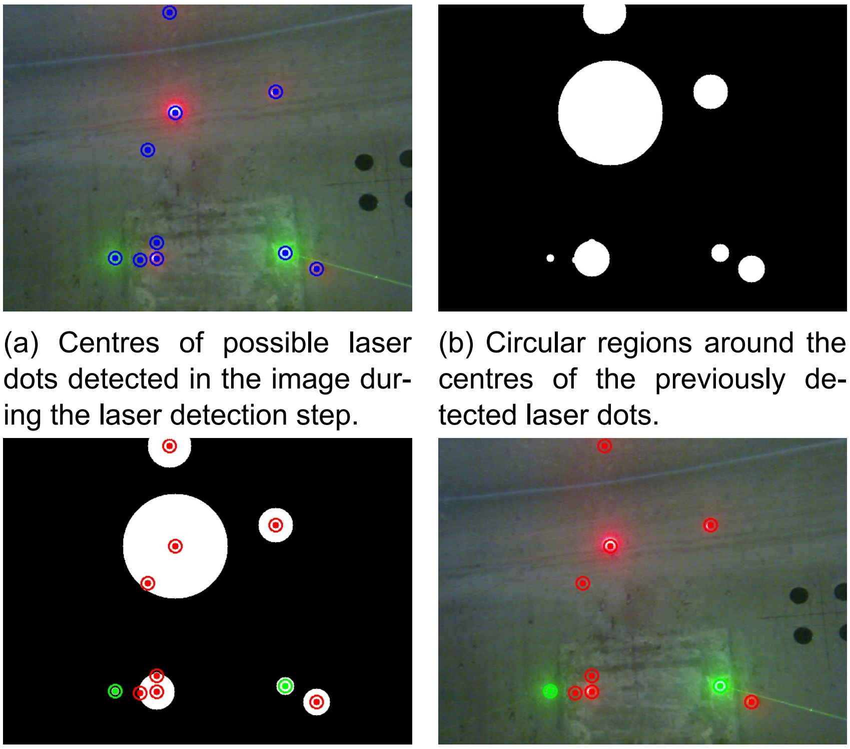
6.2.2 Colour Detection

In the previous step, certain areas of the image were classified as possible laser dots. Based on the size of each area, a radius for that area is calculated and a circular region that is slightly larger than the area is defined (Fig. 6.3b). The centre of the circular region coincides with the centre of the corresponding area and its radius is equal to the previously calculated radius. For each area, the algorithm calculates the mean colour value of the circular region based on the colour values of the pixels in the original image. A mean colour value is expressed as a triplet of mean values, where each member of the triplet corresponds to each channel of the RGB colour space.

The mean colour value of each circular region characterizes the corresponding blob area for which that region was defined (Fig. 6.3c). If a mean colour value lies inside certain limits, which are expressed as differences between the members of the triplet that represents it, then the corresponding blob area can be classified into three (3) categories: red laser, green laser or false positive. If a blob area is classified as a false positive, it means that the area does not correspond to a real laser dot because its colour is neither red nor green and it is discarded. Thus, this step finalizes the detection procedure and returns two structures, one that contains information about the red laser dots and the other about the green ones.

6.2.3 Final Selection

The vision algorithm must detect six (6) laser dots; four (4) of them must be red and the other two (2) must be green. This final step guarantees that the number of detected red or green laser



(a) Centres of possible laser dots detected in the image during the laser detection step.

(b) Circular regions around the centres of the previously detected laser dots.

(c) Circular regions that are classified as red or green laser dots.

(d) Centres of real laser dots detected in the image.

Figure 6.3: The colour detection operation.

dots will not exceed the appropriate value and it is executed only when the latter is true, i.e. when red > 4 and green > 2 (Fig. 6.4a).

The algorithm chooses the best correspondence among a group of blobs classified as laser dots that lie in the area of a real laser dot. These blobs are usually created during the thresholding as part of the segmentation procedure. They either belong to an actual laser dot or they represent false positives that the previous steps failed to recognize.

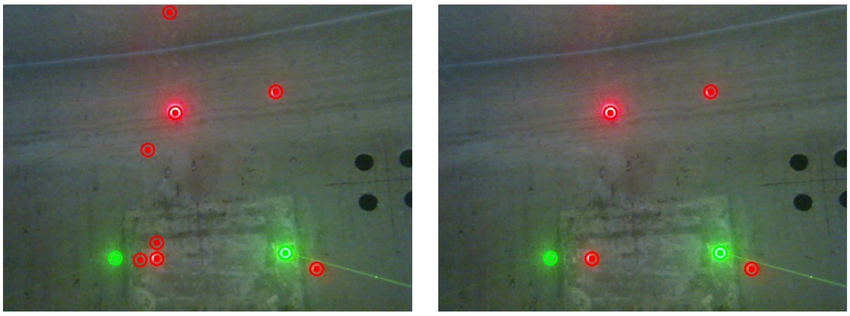
For each group of detected laser dots (red or green), the algorithm compares all laser dots pairwise based on their distance with respect to their size. At each iteration, it rejects from the pair the smaller laser dot. These procedure is executed iteratively until the desired number of laser dots is reached (i.e. red $= 4$ and green $= 2$) (Fig. 6.4b).

If the number of laser dots on a group (red or green) is less than the appropriate one, it means that the previous steps have failed to detect certain laser dots and the selection procedure is not executed.

6.3 Pros & Cons

The vision algorithm is accurate and robust under various lighting conditions. It is simple and as a result its runtime is small (approximately 100-150 ms). This means that the algorithm can produce results in real-time which is very important for the control of the vehicle. In addition, the algorithm produces no false positives when all six (6) laser dots lie inside the image frame of the vehicle.

On the other hand, the vision algorithm has certain limitations. First, its efficiency on detecting the laser dots depends heavily on the relative intensity of the laser dots with respect to their environment. This may be fixed by changing the threshold value in the thresholding procedure according to the ambient lighting conditions, although this solution does not provide any guaran-



(a) Centres of real laser dots detected in the image.

(b) Centres of real laser dots detected in the image after the selection procedure.

Figure 6.4: The selection operation.

tees. Second, when a red laser dot and a green one lie very close to each other or intersect, the algorithm will fail to detect either both or only the red laser dot. This is due to the high intensity of the green dot and the fact that, when two laser dots intersect, they form a single blob which cannot be separated into a red blob and a green one using the current procedure. Finally, there are certain circumstances under which the final selection procedure may produce misplaced laser dots or false positives. These circumstances relate to the presence of reflections of laser dots from the water surface in the image frame. When these reflections have greater size than the actual laser dots that lie nearby in the image, then the final selection procedure may falsely classify them as real and reject the actual ones.

Chapter 7

Cooperation & Control

The goal of this work is to navigate two underwater vehicles in a cooperative manner in order to inspect a flat surface using a Leader - Follower approach. The vehicles use a laser pointer setup in order to localize themselves with respect to the surface and to each other. The Leader moves independently following waypoints that form a meander-like trajectory. The Follower maintains a specific distance offset with respect to the surface and to the Leader, while at the same time its motion guarantees that the Leader laser dots do not leave its image plane. The motion control scheme analysis for the two vehicles is presented in this chapter. It is worth mentioning that the control scheme of the Leader is based on the localization principles of [22] with the difference that the vehicle receives only information of its range and orientation with respect to the surface since no specific target is present.

In Fig. 7.1, a random configuration of the vehicles is shown. The image also depicts four coordinate systems: the *World Coordinate System (WCS)* with origin O_W, which is fixed and plays the role of the inertial frame, the *Global Coordinate System (GCS)* with origin O_G, which is defined with respect to the formation of the two vehicles and its position depends on the motion of the Leader vehicle, the *Leader Coordinate System (LCS)* with origin O_L, which is attached to the Leader, and the *Follower Coordinate System (FCS)* with origin O_F, which is attached to the Follower.

The coordinate systems WCS and GCS are parallel to each other. As a result, the yaw orientation of a vehicle with respect to the surface has the same value when expressed on either of the two coordinate systems. Moreover, the y-axis of both systems lies on the surface. Based on that fact and on the parallelism of the two CSs, it can be concluded that the x coordinate of a vehicle is the same for both CSs. The above conclusions can be also mathematically stated as:

$$\begin{aligned} x_i &= {}^{G}x_i \\ \psi_i &= {}^{G}\psi_i \end{aligned} \qquad \text{where } i = L, F \tag{7.1}$$

The same applies also for relative sizes and velocities, e.g. $x_L^F = {}^{G}x_L^F$ and $\dot{x}_F = {}^{G}\dot{x}_F$.

In the following, the notation ${}^{j}z_i$ will denote a size that corresponds to i and is expressed with respect to the coordinate system j. For example, ${}^{G}\mathbf{n}_L$ denotes the pose (position and orientation) of the Leader (L) expressed with respect to the Global Coordinate System (G). The notation ${}^{j}z_i^k$ will denote a size of i relative to k and expressed in the coordinate system of j. For example, ${}^{G}\mathbf{n}_L^F$ denotes the relative pose of the Leader (L) with respect to the Follower (F) expressed in the GCS (G). When the above notation is used without the superscript j, the size is expressed with respect to the World Coordinate System (WCS).

The kinematic models of both vehicles, which respresent their movement with respect to the WCS, are already presented in Chapter 2 Section 2.2, but they are also shown in this chapter for readability purposes.

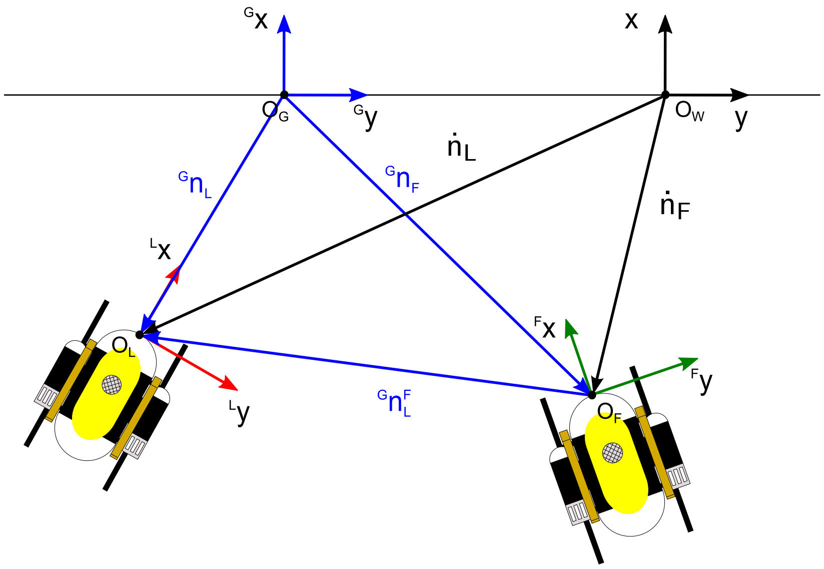


Figure 7.1: The vehicles presented in a random configuration. The *World Coordinate System*, the *Global Coordinate System*, the *Leader Coordinate System* and the *Follower Coordinate System* are also depicted.

$$\begin{aligned} \dot{x}_L &= u_L \cos\psi_L \\ \dot{y}_L &= u_L \sin\psi_L \\ \dot{z}_L &= w_L \\ \dot{\psi}_L &= r_L \end{aligned} \tag{7.2}$$

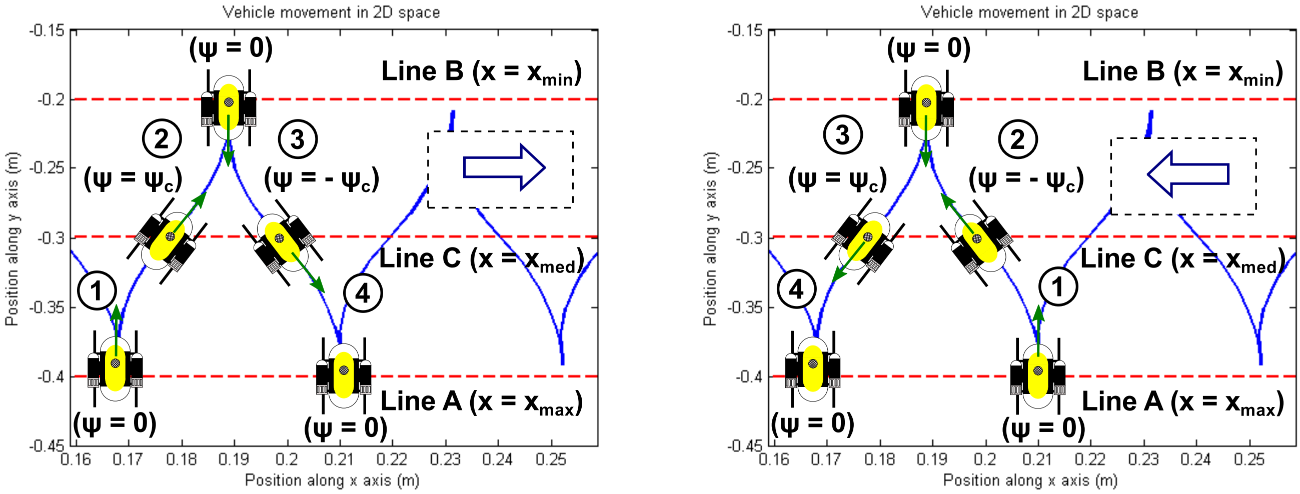
$$\begin{aligned} \dot{x}_F &= u_F \cos\psi_F - v_F \sin\psi_F \\ \dot{y}_F &= u_F \sin\psi_F + v_F \cos\psi_F \\ \dot{z}_F &= w_F \\ \dot{\psi}_F &= r_F \end{aligned} \tag{7.3}$$

7.1 Leader Motion Control Design

As it is already described in Chapter 3, the Leader vehicle moves in a saw-like manner in order to inspect the flat surface while maintaining an orientation that is almost perpendicular to it. In particular, as depicted in Fig. 7.2, the Leader moves back and forth reaching lines A, B and C in a cycling order (i.e. A-C-B-C-A) at a specific orientation each time. The lines A, B and C represent different distances from the surface along the x-axis of the WCS that the vehicle must reach. According to the desired direction of motion (left or right), the orientation at which the vehicle reaches lines A, B or C may differ each time.

	Right Motion		Left Motion	
	$\mathbf{x_d}$	$\boldsymbol{\psi_d}$	$\mathbf{x_d}$	$\boldsymbol{\psi_d}$
Mode 1	x_{med}	ψ_c	x_{med}	$-\psi_c$
Mode 2	x_{min}	0	x_{min}	0
Mode 3	x_{med}	$-\psi_c$	x_{med}	ψ_c
Mode 4	x_{max}	0	x_{max}	0

Table 7.1: Modes of motion based on the desired direction.



(a) Vehicle motion to the right.

(b) Vehicle motion to the left.

Figure 7.2: Control scheme for each direction of motion (more detailed).

This leads to the definition of four (4) modes of motion for each direction (see Table 7.1). Each mode is characterised by two desired states that the vehicle must reach, namely, the desired position along the x-axis x_d and the desired orientation ψ_d. Each line in Fig. 7.2a and 7.2b corresponds to a vertical distance from the surface. In particular, lines A, B and C correspond to the distances d_{max}, d_{min} and d_{med}. Thus, the potential values of x_d are denoted as x_{min}, x_{max} and x_{med}, which are related to each of the previously defined distances, respectively. The desired orientation ψ_d may take the values 0, ψ_c or $-\psi_c$, where ψ_c is a positive constant ($\psi_c > 0$).

During the control operation, a switching scheme is feeding a controller with values for the desired states x_d and ψ_d based on specific criteria, as depicted in Fig. 7.3. More precisely, these criteria refer to whether the vehicle lies between two lines or on the vicinity of a particular line, its orientation and the desired direction of motion.

In order to clarify the operation of the control scheme, the mode description of Chapter 3 shall be revisited. Let us assume again that the vehicle must move to the right, but it has a negative yaw angle (Mode 4 as depicted in Fig. 7.2a). The vehicle will move towards line A and will try to attain a zero yaw angle, as imposed by the desired values of Mode 4. When it reaches close to line A ($x_L \approx x_{max}$, $\psi_L \approx 0$), the switching scheme will feed the controller with the desired values of Mode 1 (see also Table 7.1 for the right direction). The switching scheme will continue to feed the controller with the same desired values (Mode 1) until the vehicle reaches line C with a positive yaw angle ψ_c. At the vicinity of line C ($x_L \approx x_{med}$, $\psi_L \approx \psi_c > 0$), the controller will be fed with the desired values of Mode 2, until it reaches line B ($x_L \approx x_{min}$, $\psi_L \approx 0$) where Mode 3 is activated. Finally, the vehicle will move to line C trying to attain a negative yaw angle $-\psi_c$. When the vehicle reaches line C ($x_L \approx x_{med}$, $\psi = -\psi_c < 0$) Mode 4 is activated. The vehicle then moves towards line A and the cycle is reinitialized. The same exact procedure is followed for the case of the left motion. The only difference lies in the desired values for each Mode.

The values of x_{min}, x_{med}, and x_{max} depend on the desired waypoints that the vehicle must reach. A waypoint can be denoted as $\mathbf{r_{dG}} = [\begin{array}{ccc} x_{dG} & y_{dG} & z_{dG} \end{array}]^T$ ("G" is for Global) in order to distinguish it from the desired values $\mathbf{r_d} = [\begin{array}{cc} x_d & \psi_d \end{array}]^T$ that are fed to the controller at each iteration and are related to its saw-like motion. Based on the above notation with regard to the waypoints, the values of x_{min}, x_{med}, and x_{max} can be written as:

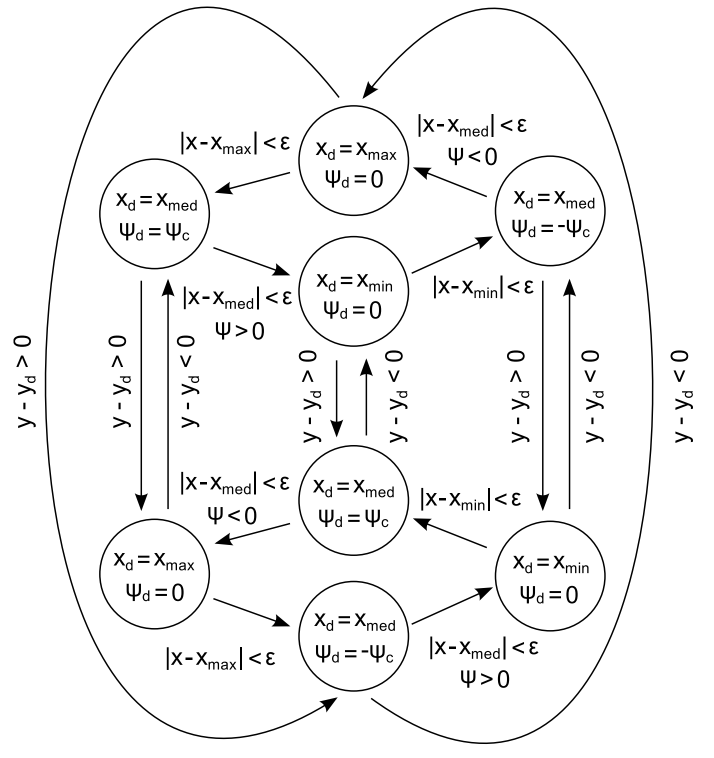


Figure 7.3: The switching scheme that feeds the motion controller of the Leader with desired states.

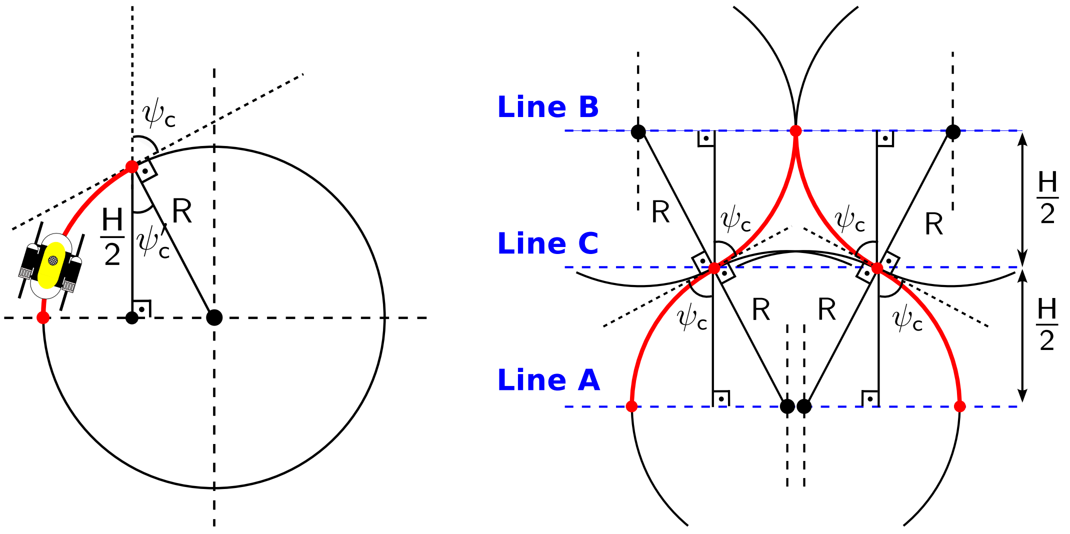
$$x_{min} = x_{dG} - \frac{H}{2}$$
$$x_{med} = x_{dG}$$
$$x_{max} = x_{dG} + \frac{H}{2}$$

where H denotes the distance between the lines A and B ($H > 0$). The value of ψ_c is chosen so as to provide a smooth motion during the control operation, especially with respect to the yaw rotation. This is due to the fact that in inspection operations abrupt yaw rotations may lead to blurry images or incomprehensible videos. For that reason, the motion of the Leader corresponds to a motion of a vehicle that moves on the circumference of a circle. Thus, the value of ψ_c is calculated based on Fig. 7.4a as:

$$\psi_c = \frac{\pi}{2} - \psi_c' = \frac{\pi}{2} - \cos^{-1}\left(\frac{H}{2R}\right) \tag{7.4}$$

where R is the radius of the circle and H is the distance that the vehicle traverses during its motion. This is the same parameter H that was used to calculate the values of x_{min}, x_{med} and x_{max}. As depicted in Fig. 7.4b, for each mode the centre of the corresponding circle is in a different position, but in all cases the values of R and H are the same and, thus, the parameter ψ_c remains the same. It is also worth mentioning that there is no guarantee that the vehicle will move on the circumference of the defined circle. The calculation of ψ_c provides only a desired value for the yaw angle. In order to navigate the vehicle on the circumference of the circle, the control gains of the surge and yaw motion must be carefully chosen. In this work, the goal is to move the Leader near the circumference (not exactly on it) without taking into account any deviation from it in order to navigate the vehicle parallel to the surface.

As already mentioned, a single controller is employed for all modes of motion of the Leader. Thus, in order to prove the stability of the proposed motion control scheme, it suffices to prove the



(a) The calculation of the constant ψ_c based on the premise that the vehicle moves on the circumference of a circle with radius R.

(b) The full desired trajectory of the Leader during one cycle of motion.

Figure 7.4: The desired motion of the Leader during the control procedure.

stability of the controller in only one mode of motion. The proof is identical for the other modes. The structure of the switching scheme guarantees that, when the vehicle reaches asymptotically the desired state of a mode, it will be led directly to the next mode. This will continue until the vehicle reaches the desired waypoint.

The above analysis referred only to the left and right motions of the Leader, while the vehicle must also reach waypoints along the vertical direction (z-axis). The motion of the vehicle on the xy-plane is independent from its motion along the z (heave) direction. This means that two separate controllers can be derived; one for its planar motion and another one for its motion along the z-axis.

Theorem 1. *The state vector $p_{xy} = [\ x_L \quad \psi_L\ ]^T$ converges asymptotically to the desired state vector $p_{xy,d} = [\ x_d \quad \psi_d\ ]^T$ under the control laws:*

$$\begin{aligned} u_L &= -k_u \frac{x_L - x_d}{\cos\psi_L} \\ r_L &= -k_r(\psi_L - \psi_d) \end{aligned} \tag{7.5}$$

where k_u, k_r are positive constants ($k_u, k_r > 0$).

Proof. First, let V_{xy} be a positive definite candidate Lyapunov function of the form:

$$V_{xy} = \frac{1}{2}(x_L - x_d)^2 + \frac{1}{2}(\psi_L - \psi_d)^2 \tag{7.6}$$

The derivative of V_{xy} is given by:

$$\begin{aligned} \dot{V}_{xy} &= (x_L - x_d)(\dot{x}_L - \dot{x}_d) + (\psi_L - \psi_d)(\dot{\psi}_L - \dot{\psi}_d) \Rightarrow \\ \dot{V}_{xy} &= (x_L - x_d)\dot{x}_L + (\psi_L - \psi_d)\dot{\psi}_L \end{aligned} \tag{7.7}$$

because each time x_d and ψ_d are contants. Using Eq. 7.2, Eq. 7.7 becomes:

$$\dot{V}_{xy} = (x_L - x_d)u_L\cos\psi_L + (\psi_L - \psi_d)r_L \tag{7.8}$$

Substituting the control laws 7.5 into Eq. 7.8 yields:

$$\dot{V}_{xy} = -k_u(x_L - x_d)^2 - k_r(\psi_L - \psi_d)^2 \leq 0 \tag{7.9}$$

and, thus, the proof is concluded. □

Theorem 2. *The state z_L converges asymptotically to the desired state z_d under the control law:*

$$w_L = -k_w(z_L - z_d) \tag{7.10}$$

where k_w is a positive constant ($k_w > 0$).

Proof. The proof is identical to that provided for the previous theorem. □

The above theorem refers to the simulation procedure. During the experiments, though, the controller should be able to compensate also for the buoyancy of the vehicle. The motion of the vehicle along the z-axis relates to low speeds and the system can be considered as linear. Thus, during the experimental procedure, the z motion can be controlled by a PID controller of the form:

$$w_L = -k_{pz}e_z - k_{dz}\dot{e}_z - k_{iz}\sum e_z dt \tag{7.11}$$

where $e_z = z_L - z_d$, $\dot{e}_z$ is the derivative of the error, dt is the time interval between two successive measurements and k_{pz}, k_{dz}, k_{iz} are positive constants ($k_{pz}, k_{dz}, k_{iz} > 0$).

In general, the goal of the Leader controller is to navigate the vehicle through certain waypoints that form a meander-like trajectory on the yz-plane in order to inspect the flat surface. Each time the vehicle reaches a waypoint, though, it is not necessary to have a specific distance offset from the surface as long as the trajectory in the yz-plane is well formed. This means that only the errors along the y- and z-directions with respect to each waypoint are considered. The latter will become clearer in Chapter 8 and Chapter 10.

7.2 Follower Motion Control Design

The localization system which is equipped on the Follower can provide the following information:

- ${}^{G}\mathbf{n}_L$: the pose of the Leader with respect to the GCS
- ${}^{G}\mathbf{n}_F$: the pose of the Follower with respect to the GCS
- ${}^{G}\mathbf{n}_L^F$: the relative pose of the Leader with respect to the Follower expressed in the GCS
- ${}^{G}\mathbf{n}_F^L$: the relative pose of the Follower with respect to the Leader expressed in the GCS
- $\mathbf{s}_{lc}$: the image space coordinates of the centre of the four laser dots produced by the Leader
- ${}^{cF}\mathbf{r}_{lc}$: the cartesian coordinates of the same centre expressed in the FcCS

As already mentioned, the pose $\mathbf{n}$ of a vehicle can be expressed as $\mathbf{n} = [\begin{array}{cccc} x & y & z & \psi \end{array}]^T$. The image space coordinates of the laser centre can be expressed as $\mathbf{s}_{lc} = [\begin{array}{cc} u_{lc} & v_{lc} \end{array}]^T$, while the corresponding cartesian coordinates can be expressed as ${}^{cF}\mathbf{r}_{lc} = [\begin{array}{ccc} {}^{cF}x_{lc} & {}^{cF}y_{lc} & {}^{cF}z_{lc} \end{array}]^T$.

By means of a simple differentiation and a subsequent application of a smoothing filter, the localization system can provide accurate estimates of the derivatives of the above sizes. For the analysis that follows, only the derivatives of the first four sizes are needed, i.e.:

- ${}^{G}\dot{\mathbf{n}}_L$: the linear and angular velocities of the Leader with respect to the GCS

- ${}^{G}\dot{\mathbf{n}}_F$: the linear and angular velocities of the Follower with respect to the GCS
- ${}^{G}\dot{\mathbf{n}}_L^F$: the relative linear and angular velocities of the Leader with respect to the Follower expressed in the GCS
- ${}^{G}\dot{\mathbf{n}}_F^L$: the relative linear and angular velocities of the Follower with respect to the Leader expressed in the GCS

The goal of the subsequent analysis is to derive a controller for the Follower based on the kinematic models of both vehicles that will drive the state vector $\mathbf{p} = [\ {}^{G}x_F \quad {}^{G}y_F \quad u_{lc} \quad v_{lc}\ ]^T$ to the desired state vector $\mathbf{p}_d = [\ {}^{G}x_d \quad {}^{G}y_d \quad u_d \quad v_d\ ]^T$. Each desired state can be expressed as:

$$\begin{aligned}
{}^{G}x_d &= c_x \\
{}^{G}y_d &= {}^{G}y_L + c_y \\
u_d &= \frac{\text{WIDTH} + 1}{2} \\
v_d &= \frac{\text{HEIGHT} + 1}{2}
\end{aligned}$$

where c_x is a negative constant ($c_x < 0$), c_y is a positive constant ($c_y > 0$) and the variables WIDTH and HEIGHT denote the width and the height of the image frame, respectively.

The Follower vehicle does not have any non-holonomic constraints (Eq. 7.3). Thus, the motion control scheme is based on the premise that the state vector $\mathbf{p}_{xy} = [\ {}^{G}x_F \quad {}^{G}y_F\ ]^T$ (first two states of $\mathbf{p}$) is controlled by the surge (u_F) and sway (v_F) motions of the vehicle, while the vector $\mathbf{p}_{uv} = [\ u_{lc} \quad v_{lc}\ ]^T$ (last two states of $\mathbf{p}$) is controlled by the heave (w_F) and yaw (r_F) motions. The control laws that drive the vector $\mathbf{p}_{xy}$ to its desired value $\mathbf{p}_{xy,d}$ are designed on the cartesian space, which is also the configuration space of the vehicle. The control laws that drive the vector $\mathbf{p}_{uv}$ to its desired value $\mathbf{p}_{uv,d}$ are designed on the image space.

Theorem 3. *The state vector $\mathbf{p}_{xy} = [\ {}^{G}x_F \quad {}^{G}y_F\ ]^T$ converges asymptotically to the desired state vector $\mathbf{p}_{xy,d} = [\ {}^{G}x_d \quad {}^{G}y_d\ ]^T$ under the control laws:*

$$\begin{aligned}
u_F &= -k_x({}^{G}x_F - {}^{G}x_d)\cos{}^{G}\psi_F - k_y({}^{G}y_F - {}^{G}y_d)\sin{}^{G}\psi_F + \dot{y}_L \sin{}^{G}\psi_F \\
v_F &= k_x({}^{G}x_F - {}^{G}x_d)\sin{}^{G}\psi_F - k_y({}^{G}y_F - {}^{G}y_d)\cos{}^{G}\psi_F + \dot{y}_L \cos{}^{G}\psi_F
\end{aligned} \tag{7.12}$$

where k_x, k_y are positive constants ($k_x, k_y > 0$) and $\dot{y}_L$ denotes the motion of the Leader along the y-axis of the WCS.

Proof. First, let V_{xy} be a positive definite candidate Lyapunov function of the form:

$$V_{xy} = \frac{1}{2}({}^{G}x_F - {}^{G}x_d)^2 + \frac{1}{2}({}^{G}y_F - {}^{G}y_d)^2 \tag{7.13}$$

The derivative of V_{xy} is given by:

$$\begin{aligned}
\dot{V}_{xy} &= ({}^{G}x_F - {}^{G}x_d)({}^{G}\dot{x}_F - {}^{G}\dot{x}_d) + ({}^{G}y_F - {}^{G}y_d)({}^{G}\dot{y}_F - {}^{G}\dot{y}_d) \Rightarrow \\
\dot{V}_{xy} &= ({}^{G}x_F - {}^{G}x_d){}^{G}\dot{x}_F + ({}^{G}y_F - {}^{G}y_d)({}^{G}\dot{y}_F - {}^{G}\dot{y}_L) \Rightarrow \\
\dot{V}_{xy} &= ({}^{G}x_F - {}^{G}x_d)\dot{x}_F + ({}^{G}y_F - {}^{G}y_d)(\dot{y}_F - \dot{y}_L)
\end{aligned} \tag{7.14}$$

based on the definitions of WCS and GCS, their parallelism and the definitions of ${}^{G}x_d$ and ${}^{G}y_d$. Using Eq. 7.3, Eq. 7.14 becomes:

$$\dot{V}_{xy} = ({}^{G}x_F - {}^{G}x_d)(u_F \cos\psi_F - v_F \sin\psi_F) + ({}^{G}y_F - {}^{G}y_d)(u_F \sin\psi_F + v_F \cos\psi_F - \dot{y}_L) \tag{7.15}$$

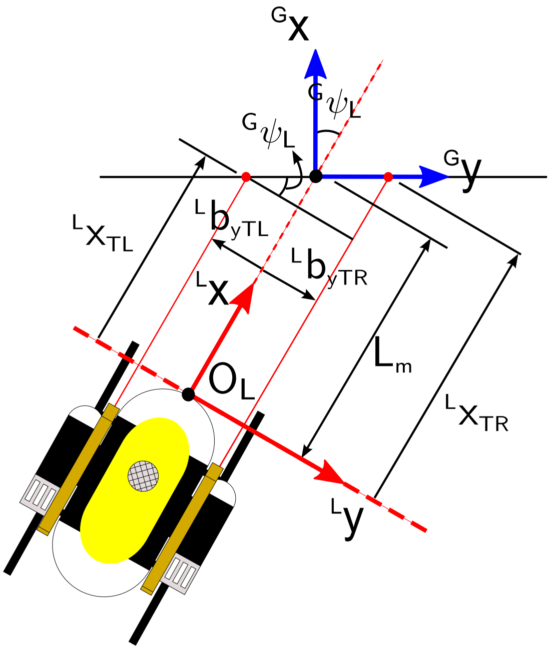


Figure 7.5: The x- and y-coordinates of the laser dots w.r.t. the LCS.

Substituting the control laws 7.12 into Eq. 7.15 and remembering Eq. 7.1 yields:

$$\dot{V}_{xy} = -k_x({}^G x_F - {}^G x_d)^2 - k_y({}^G y_F - {}^G y_d)^2 \leq 0 \tag{7.16}$$

and, thus, the proof is concluded. □

In the above Theorem, we used the fact that ${}^G\dot{y}_F -^G \dot{y}_L = \dot{y}_F - \dot{y}_L$. This assumption is based on the parallelism of the coordinate systems WCS and GCS, which leads to the fact that any relative size has the same value in both CSs.

In order to derive the corresponding controllers for the yaw and heave motion, a similar procedure as above is followed. Due to the fact that the heave and yaw motions depend on the errors $(u_{cl} - u_d)$ and $(v_{cl} - v_d)$, respectively, there is a need to express $\dot{u}_{cl}$ and $\dot{v}_{cl}$ as functions of the Follower control input $\mathbf{u}_F$. The corresponfing derivation is described in detail below.

First, we consider the cartesian coordinates of the centre of the four Leader laser dots with respect to the LCS. In order to do so, we must first calculate the cartesian coordinates of each laser dot of the Leader projected on the flat surface and then calculate their mean value.

Let us assume that the point of origin for a laser beam is denoted with respect to the LCS by a vector ${}^L b = [\ {}^L b_x \quad {}^L b_y \quad {}^L b_z\ ]^T$. Based on the configuration of the laser pointers equipped on the Leader vehicle, the following relations hold:

$$\begin{aligned} {}^L b_{xTL} &= {}^L b_{xTR} = {}^L b_{xBL} = {}^L b_{xBR} \\ {}^L b_{yTL} &= -{}^L b_{yTR} \\ {}^L b_{yBL} &= -{}^L b_{yBR} \\ {}^L b_{zTL} &= {}^L b_{zTR} \end{aligned} \tag{7.17}$$

where the subscripts TL, TR, BL, BR denote the Top Left, Top Right, Bottom Left and Bottom Right laser pointers as seen from the coordinate system of the Leader. The mean range of the Leader vehicle expressed as a function of its state is written as:

$$L_{m,L} = -\frac{{}^G x_L}{\cos {}^G\psi_L}$$

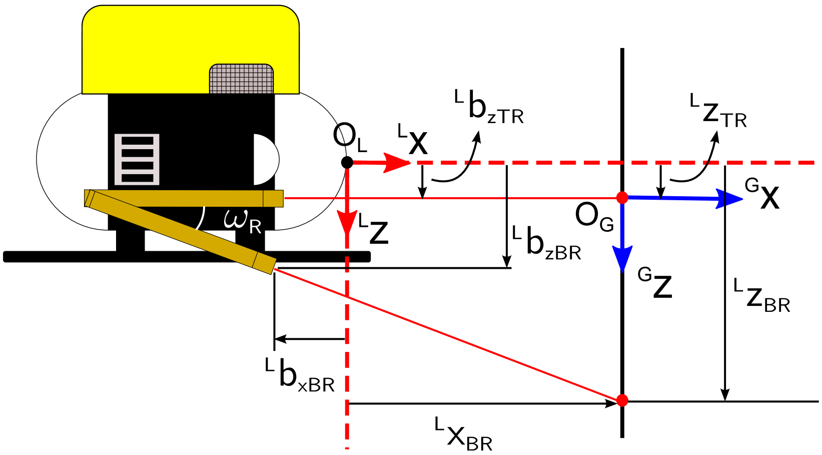


Figure 7.6: The x-coordinates of the laser dots w.r.t. the LCS computed as functions of the z-coordinates.

Then, as depicted roughly in Fig. 7.5 for the cases of TL and TR, the x-coordinates for the Leader laser dots can be expressed as:

$$\begin{aligned}
{}^{L}x_{TL} &= -\frac{{}^{G}x_{L}}{\cos {}^{G}\psi_{L}} + {}^{L}b_{yTL} \tan {}^{G}\psi_{L} \\
{}^{L}x_{TR} &= -\frac{{}^{G}x_{L}}{\cos {}^{G}\psi_{L}} + {}^{L}b_{yTR} \tan {}^{G}\psi_{L} \\
{}^{L}x_{BL} &= -\frac{{}^{G}x_{L}}{\cos {}^{G}\psi_{L}} + {}^{L}b_{yBL} \tan {}^{G}\psi_{L} \\
{}^{L}x_{BR} &= -\frac{{}^{G}x_{L}}{\cos {}^{G}\psi_{L}} + {}^{L}b_{yBR} \tan {}^{G}\psi_{L}
\end{aligned} \tag{7.18}$$

The corresponding y-coordinates (Fig. 7.5) are expressed as:

$$\begin{aligned}
{}^{L}y_{TL} &= {}^{L}b_{yTL} \\
{}^{L}y_{TR} &= {}^{L}b_{yTR} \\
{}^{L}y_{BL} &= {}^{L}b_{yBL} \\
{}^{L}y_{BR} &= {}^{L}b_{yBR}
\end{aligned} \tag{7.19}$$

As depicted in Fig. 7.6 for the case of BR, the x-coordinates of the Bottom Left and Bottom Right laser dots can also be calculated as:

$$\begin{aligned}
{}^{L}x_{BL} &= \frac{{}^{L}z_{BL} - {}^{L}z_{TL}}{\tan \omega_{L}} - \frac{{}^{L}b_{zBL} - {}^{L}b_{zTL}}{\tan \omega_{L}} + {}^{L}b_{xBL} \\
{}^{L}x_{BR} &= \frac{{}^{L}z_{BR} - {}^{L}z_{TR}}{\tan \omega_{R}} - \frac{{}^{L}b_{zBR} - {}^{L}b_{zTR}}{\tan \omega_{R}} + {}^{L}b_{xBR}
\end{aligned} \tag{7.20}$$

Thus, the z-coordinates for the four Leader laser dots (see also Fig. 7.6) can be expressed as:

$$\begin{aligned}
{}^{L}z_{TL} &= {}^{L}b_{zTL} \\
{}^{L}z_{TR} &= {}^{L}b_{zTR} \\
{}^{L}z_{BL} &= {}^{L}b_{zBL} + \left(-\frac{{}^{G}x_{L}}{\cos {}^{G}\psi_{L}} + {}^{L}b_{yBL} \tan {}^{G}\psi_{L} - {}^{L}b_{xBL} \right) \tan \omega_{L} \\
{}^{L}z_{BR} &= {}^{L}b_{zBR} + \left(-\frac{{}^{G}x_{L}}{\cos {}^{G}\psi_{L}} + {}^{L}b_{yBR} \tan {}^{G}\psi_{L} - {}^{L}b_{xBR} \right) \tan \omega_{R}
\end{aligned} \tag{7.21}$$

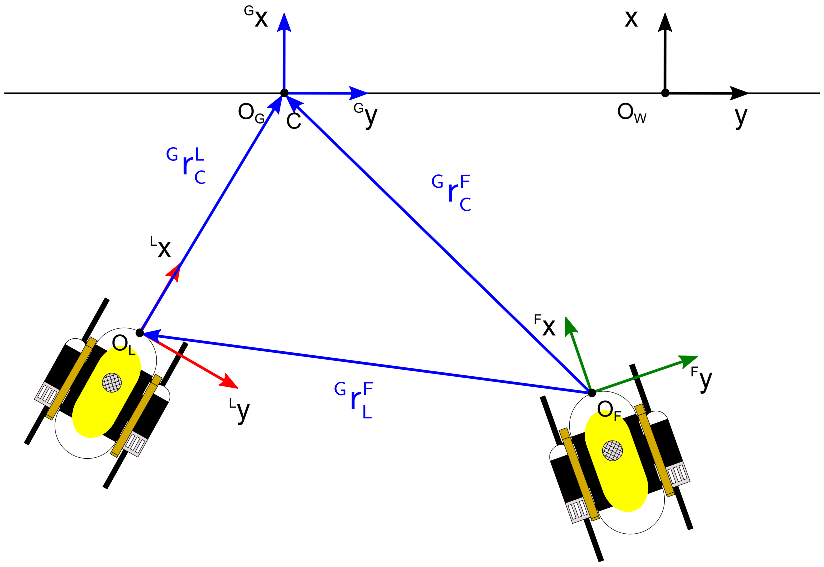


Figure 7.7: The position of the laser centre in cartesian space calculated w.r.t. FCS.

In order to calculate the cartesian coordinates of the Leader laser centre, we take the mean value of the x-, y- and z-coordinates that correspond to the Leader lasers, which, based also on Eq. 7.17, leads to:

$$\begin{aligned}
{}^{L}x_{lc} &= -\frac{{}^{G}x_L}{\cos {}^{G}\psi_L} \\
{}^{L}y_{lc} &= 0 \\
{}^{L}z_{lc} &= \frac{2{}^{L}b_{zTL} + {}^{L}b_{zBL} + {}^{L}b_{zBR}}{4} + \left(\frac{\tan\omega_L + \tan\omega_R}{4}\right)\left(-\frac{{}^{G}x_L}{\cos {}^{G}\psi_L} - {}^{L}b_{xBL}\right) \\
&\quad + \left(\frac{\tan\omega_L - \tan\omega_R}{4}\right){}^{L}b_{yBL}\tan {}^{G}\psi_L
\end{aligned} \tag{7.22}$$

In order to express the vector ${}^{L}\mathbf{r}_{lc} = [\ {}^{L}x_{lc} \quad {}^{L}y_{lc} \quad {}^{L}z_{lc}\ ]^T$ with respect to the GCS, we must apply a coordinate transformation using a rotation matrix:

$$ {}^{G}\mathbf{r}^{L}_{lc} = {}^{G}R_L {}^{L}\mathbf{r}_{lc} \tag{7.23}$$

where

$$ {}^{G}R_L = \begin{bmatrix} \cos {}^{G}\psi_L & -\sin {}^{G}\psi_L & 0 \\ \sin {}^{G}\psi_L & \cos {}^{G}\psi_L & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.24}$$

The localization system provides the relative pose of the Leader with respect to the Follower ${}^{G}\mathbf{n}^{F}_{L}$. Its relative position can be denoted as ${}^{G}\mathbf{r}^{F}_{L}$. Consequently, the position of the laser centre with respect to the Follower can be expressed as:

$$ {}^{G}\mathbf{r}^{F}_{lc} = {}^{G}\mathbf{r}^{L}_{lc} + {}^{G}\mathbf{r}^{F}_{L} \tag{7.25}$$

which is also depicted in Fig. 7.7.

The vector ${}^{G}\mathbf{r}^{F}_{lc}$ is expressed with respect to the FCS by applying another coordinate transformation as follows:

$$ {}^{F}\mathbf{r}_{lc} = {}^{F}R_G {}^{G}\mathbf{r}^{F}_{lc} \tag{7.26}$$

where

$$ {}^{F}R_G = {}^{G}{R_F}^{-1} = {}^{G}{R_F}^{T} = \begin{bmatrix} \cos{}^{G}\psi_F & \sin{}^{G}\psi_F & 0 \\ -\sin{}^{G}\psi_F & \cos{}^{G}\psi_F & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.27} $$

Finally, ${}^{F}\mathbf{r}_{lc}$ is expressed with respect to the coordinate system of the camera equipped on the Follower as:

$$ {}^{cF}\mathbf{r}_{lc} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} {}^{F}\mathbf{r}_{lc} \tag{7.28} $$

which yields the following equations:

$$ \begin{aligned} {}^{cF}x_{lc} &= {}^{G}x_F \sin{}^{G}\psi_F - {}^{G}x_L \tan{}^{G}\psi_L \cos{}^{G}\psi_F + ({}^{G}y_L - {}^{G}y_F)\cos{}^{G}\psi_F \\ {}^{cF}y_{lc} &= -{}^{G}x_F \cos{}^{G}\psi_F - {}^{G}x_L \tan{}^{G}\psi_L \sin{}^{G}\psi_F + ({}^{G}y_L - {}^{G}y_F)\sin{}^{G}\psi_F \\ {}^{cF}z_{lc} &= -\frac{2{}^{L}b_{zTL} + {}^{L}b_{zBL} + {}^{L}b_{zBR}}{4} - \left(\frac{\tan\omega_L + \tan\omega_R}{4}\right)\left(-\frac{{}^{G}x_L}{\cos{}^{G}\psi_L} - {}^{L}b_{xBL}\right) \\ &\quad - \left(\frac{\tan\omega_L - \tan\omega_R}{4}\right){}^{L}b_{yBL}\tan{}^{G}\psi_L - {}^{G}z_L + {}^{G}z_F \end{aligned} \tag{7.29} $$

Differentiation of the above equations yields:

$$ \begin{aligned} {}^{cF}\dot{x}_{lc} &= {}^{G}\dot{x}_F \sin{}^{G}\psi_F + {}^{G}x_F \cos{}^{G}\psi_F{}^{G}\dot{\psi}_F - {}^{G}\dot{x}_L \tan{}^{G}\psi_L \cos{}^{G}\psi_F - {}^{G}x_L \frac{{}^{G}\dot{\psi}_L}{(\cos{}^{G}\psi_L)^2}\cos{}^{G}\psi_F \\ &\quad + {}^{G}x_L \tan{}^{G}\psi_L \sin{}^{G}\psi_F{}^{G}\dot{\psi}_F + ({}^{G}\dot{y}_L - {}^{G}\dot{y}_F)\cos{}^{G}\psi_F - ({}^{G}y_L - {}^{G}y_F)\sin{}^{G}\psi_F{}^{G}\dot{\psi}_F \\ {}^{cF}\dot{y}_{lc} &= -{}^{G}\dot{x}_F \cos{}^{G}\psi_F + {}^{G}x_F \sin{}^{G}\psi_F{}^{G}\dot{\psi}_F - {}^{G}\dot{x}_L \tan{}^{G}\psi_L \sin{}^{G}\psi_F - {}^{G}x_L \frac{{}^{G}\dot{\psi}_L}{(\cos{}^{G}\psi_L)^2}\sin{}^{G}\psi_F \\ &\quad - {}^{G}x_L \tan{}^{G}\psi_L \cos{}^{G}\psi_F{}^{G}\dot{\psi}_F + ({}^{G}\dot{y}_L - {}^{G}\dot{y}_F)\sin{}^{G}\psi_F + ({}^{G}y_L - {}^{G}y_F)\cos{}^{G}\psi_F{}^{G}\dot{\psi}_F \\ {}^{cF}\dot{z}_{lc} &= \left(\frac{\tan\omega_L + \tan\omega_R}{4}\right)\left(\frac{{}^{G}\dot{x}_L \cos{}^{G}\psi_L + {}^{G}x_L \sin{}^{G}\psi_L{}^{G}\dot{\psi}_L}{(\cos{}^{G}\psi_L)^2}\right) - \left(\frac{\tan\omega_L - \tan\omega_R}{4}\right)\frac{{}^{L}b_{yBL}{}^{G}\dot{\psi}_L}{(\cos{}^{G}\psi_L)^2} \\ &\quad - {}^{G}\dot{z}_L + {}^{G}\dot{z}_F \end{aligned} \tag{7.30} $$

For the sake of simplicity, during the remainder of the analysis the vector $\mathbf{r}_c = [\ x_c \quad y_c \quad z_c\ ]^T$ will denote the vector ${}^{F}\mathbf{r}_{lc} = [\ {}^{cF}x_{lc} \quad {}^{cF}y_{lc} \quad {}^{cF}z_{lc}\ ]^T$, while the vector $\mathbf{s}_c = [\ u_c \quad v_c\ ]^T$ will denote the vector $\mathbf{s}_{lc} = [\ u_{lc} \quad v_{lc}\ ]^T$. The camera model can be described by the following two equations:

$$ \begin{aligned} u &= a_x \frac{x}{y} + u_0 \\ v &= -a_y \frac{z}{y} + v_0 \end{aligned} \tag{7.31} $$

The minus sign in the equation for v is used due to the fact that the z-axis in the Follower camera CS and the v-axis in the Follower image frame have opposite directions (see also Appendix B). Differentiation of the above equations yields:

$$ \begin{aligned} \dot{u} &= \frac{a_x}{y^2}(\dot{x}y - x\dot{y}) \\ \dot{v} &= -\frac{a_y}{y^2}(\dot{z}y - z\dot{y}) \end{aligned} \tag{7.32} $$

From Eq. 7.31, it is derived that:

$$\begin{aligned} x &= \frac{y}{a_x}(u-u_0) \\ z &= -\frac{y}{a_y}(v-v_0) \end{aligned} \tag{7.33}$$

By substituting the above equations to Eq. 7.32, it yields that:

$$\begin{aligned} \dot{u} &= \frac{a_x}{y}\dot{x} - \frac{u-u_0}{y}\dot{y} \\ \dot{v} &= -\frac{a_y}{y}\dot{z} - \frac{v-v_0}{y}\dot{y} \end{aligned} \tag{7.34}$$

Finally, substituting the realations 7.30 into Eq. 7.34 provides the velocity of the laser centre inside the image frame of the Follower in the following form:

$$\begin{bmatrix} \dot{u}_c \\ \dot{v}_c \end{bmatrix} = M\mathbf{u}_F + N\dot{\mathbf{x}}_L \tag{7.35}$$

which can be more analytically written as:

$$\begin{bmatrix} \dot{u}_c \\ \dot{v}_c \end{bmatrix} = \begin{bmatrix} m_1 & m_2 & m_3 & m_4 \\ m_5 & m_6 & m_7 & m_8 \end{bmatrix} \begin{bmatrix} u_F \\ v_F \\ w_F \\ r_F \end{bmatrix} + \begin{bmatrix} n_1 & n_2 & n_3 & n_4 \\ n_5 & n_6 & n_7 & n_8 \end{bmatrix} \begin{bmatrix} \dot{x}_L \\ \dot{y}_L \\ \dot{z}_L \\ \dot{\psi}_L \end{bmatrix} \tag{7.36}$$

where the terms in the above tables are written as:

$$\begin{aligned}
m_1 &= \frac{u_c-u_0}{y_c} \\
m_2 &= -\frac{a_x}{y_c} \\
m_3 &= 0 \\
m_4 &= {}^Gx_F\left(\frac{a_x\cos{}^G\psi_F}{y_c} - \frac{u_c-u_0}{y_c}\sin{}^G\psi_F\right) + \left({}^Gx_L\tan{}^G\psi_L - {}^Gy_L + {}^Gy_F\right)\left(\frac{a_x\sin{}^G\psi_F}{y_c} - \frac{u_c-u_0}{y_c}\right) \\
m_5 &= \frac{v_c-v_0}{y_c} \\
m_6 &= 0 \\
m_7 &= -\frac{a_y}{y_c} \\
m_8 &= -\frac{v_c-v_0}{y_c}\left[{}^Gx_F\sin{}^G\psi_F - {}^Gx_L\tan{}^G\psi_L\cos{}^G\psi_F + ({}^Gy_L - {}^Gy_F)\cos{}^G\psi_F\right] \\
n_1 &= -\tan{}^G\psi_L\left(\frac{a_x\cos{}^G\psi_F}{y_c} - \frac{u_c-u_0}{y_c}\sin{}^G\psi_F\right) \\
n_2 &= \left(\frac{a_x\cos{}^G\psi_F}{y_c} - \frac{u_c-u_0}{y_c}\sin{}^G\psi_F\right) \\
n_3 &= 0 \\
n_4 &= -\frac{{}^Gx_L}{(\cos{}^G\psi_L)^2}\left(\frac{a_x\cos{}^G\psi_F}{y_c} - \frac{u_c-u_0}{y_c}\sin{}^G\psi_F\right) \\
n_5 &= -\frac{a_y}{y_c}\left(\frac{\tan\omega_L + \tan\omega_R}{4\cos{}^G\psi_L} + \frac{v_c-v_0}{y_c}\tan{}^G\psi_L\sin{}^G\psi_F\right)
\end{aligned}$$

$$
\begin{aligned}
n_6 &= -\frac{(v_c - v_0)}{y_c}\sin {}^G\psi_F \\
n_7 &= \frac{a_y}{y_c} \\
n_8 &= \frac{{}^G x_L}{(\cos {}^G\psi_L)^2}\left[-\frac{a_y}{y_c}\left(\frac{\tan\omega_L + \tan\omega_R}{4}\right)\sin {}^G\psi_L + \frac{v_c - v_0}{y_c}\sin {}^G\psi_F\right] \\
&\quad + \frac{a_y}{y_c}\left(\frac{\tan\omega_L - \tan\omega_R}{4}\right)\frac{{}^L b_{yBL}}{(\cos {}^G\psi_L)^2}
\end{aligned}
$$

Based on the aforementioned analysis, the following theorem can be derived.

Theorem 4. *The state vector* $\boldsymbol{p}_{uv} = [\ u_c \quad v_c\ ]^T$ *converges asymptotically to the desired state vector* $\boldsymbol{p}_{uv,d} = [\ u_d \quad v_d\ ]^T$ *under the control laws:*

$$
\begin{aligned}
r_F &= \frac{-k_r(u_c - u_d) - m_1 u_F - m_2 v_F - n_1\dot{x}_L - n_2\dot{y}_L - n_4\dot{\psi}_L}{m_4} \\
w_F &= \frac{-k_w(v_c - v_d) - m_5 u_F - m_8 r_F - n_5\dot{x}_L - n_6\dot{y}_L - n_7\dot{z}_L - n_8\dot{\psi}_L}{m_7}
\end{aligned}
\tag{7.37}
$$

where k_r, k_w *are positive constants (*$k_r, k_w > 0$*) and the terms* m_1, m_2, m_4, m_5, m_7, m_8, n_1, n_2, n_4, n_5, n_6, n_7, n_8 *are already described in Eq. 7.36.*

Proof. First, let V_{uv} be a positive definite candidate Lyapunov function of the form:

$$
V_{uv} = \frac{1}{2}(u_c - u_d)^2 + \frac{1}{2}(v_c - v_d)^2 \tag{7.38}
$$

The derivative of V_{uv} is given by:

$$
\begin{aligned}
\dot{V}_{uv} &= (u_c - u_d)(\dot{u}_c - \dot{u}_d) + (v_c - v_d)(\dot{v}_c - \dot{v}_d) \Rightarrow \\
\dot{V}_{uv} &= (u_c - u_d)\dot{u}_c + (v_c - v_d)\dot{v}_c
\end{aligned}
\tag{7.39}
$$

based on the definitions of u_d and v_d. Using Eq. 7.36, Eq. 7.39 becomes:

$$
\begin{aligned}
\dot{V}_{uv} = (u_c - u_d)(m_1 u_F + m_2 v_F + m_4 r_F + n_1\dot{x}_L + n_2\dot{y}_L + n_4\dot{\psi}_L) + \\
(v_c - v_d)(m_5 u_F + m_7 w_F + m_8 r_F + n_5\dot{x}_L + n_6\dot{y}_L + n_7\dot{z}_L + n_8\dot{\psi}_L)
\end{aligned}
\tag{7.40}
$$

Substituting the control laws 7.37 into Eq. 7.40 yields:

$$
\dot{V}_{uv} = -k_r(u_c - u_d)^2 - k_w(v_c - v_d)^2 \leq 0 \tag{7.41}
$$

and, thus, the proof is concluded. □

In order to conclude this analysis, the calculation of the vector $\dot{\mathbf{x}}_L$ will be presented. Based on notions of differential kinematics, the following relation stands:

$$
{}^L\dot{\mathbf{r}}_L = {}^L\dot{\mathbf{r}}_G + {}^L R_G {}^G\dot{\mathbf{r}}_L + \left({}^L\omega_G\right)^x {}^G r_L \tag{7.42}
$$

where $\mathbf{r}$ denotes position, $\dot{\mathbf{r}}$ denotes linear velocity, $\boldsymbol{\omega}$ denotes here angular velocity and the vector ${}^L\dot{\mathbf{r}}_L = [\ u_L \quad v_L \quad w_L\ ]^T$ denotes the linear velocity of the Leader with respect to its own coordinate system. The terms u_L and w_L also happen to be its control inputs for the surge and heave motion, respectively. Also:

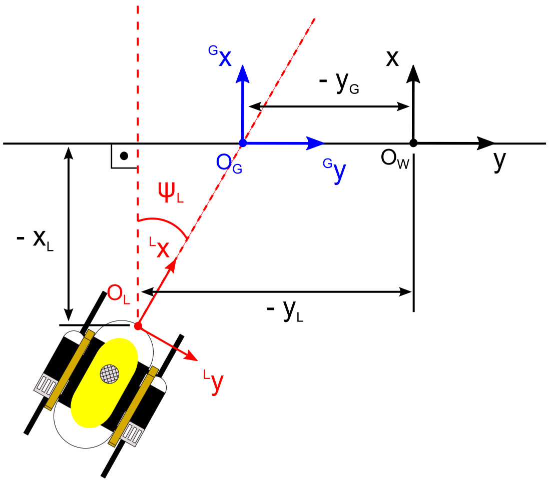


Figure 7.8: The derivation of the velocity $\dot{y}_G$ which corresponds to the lateral motion of the GCS as expressed w.r.t. the WCS. All sizes are expressed w.r.t. the WCS and the minus signs denote the fact that geometrical quantities are always positive.

$$ {}^{L}R_G = {}^{G}R_L^{\;-1} = {}^{G}R_L^{\;T} = \begin{bmatrix} \cos{}^{G}\psi_L & \sin{}^{G}\psi_L & 0 \\ -\sin{}^{G}\psi_L & \cos{}^{G}\psi_L & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.43} $$

The Eq. 7.42 can also be written as:

$$ {}^{L}\dot{\mathbf{r}}_L = {}^{L}R_G{}^{G}\dot{\mathbf{r}}_G + {}^{L}R_G{}^{G}\dot{\mathbf{r}}_L + {}^{L}R_G\left({}^{\boldsymbol{G}}\boldsymbol{\omega_G}\right)^{x}{}^{G}r_L \tag{7.44} $$

which, if solved for u_L, yields:

$$ \begin{aligned} u_L &= \cos{}^{G}\psi_L{}^{G}\dot{x}_G + \sin{}^{G}\psi_L{}^{G}\dot{y}_G + \cos{}^{G}\psi_L{}^{G}\dot{x}_L + \sin{}^{G}\psi_L{}^{G}\dot{y}_L \Rightarrow \\ u_L &= \sin{}^{G}\psi_L{}^{G}\dot{y}_G + \cos{}^{G}\psi_L{}^{G}\dot{x}_L + \sin{}^{G}\psi_L{}^{G}\dot{y}_L \end{aligned} \tag{7.45} $$

since ${}^{G}\dot{x}_G = 0$ (the GCS does not leave the surface) and $\left({}^{\boldsymbol{G}}\boldsymbol{\omega_G}\right) = \mathbf{0}_{3x1}$ (the GCS does not revolve). It can be considered that the linear velocity ${}^{G}\dot{y}_G$ as expressed with respect to the GCS is equal to the velocity $\dot{y}_G$ which is expressed to the WCS ($\dot{y}_G = {}^{G}\dot{y}_G$). According to Fig. 7.8, the latter can be computed as:

$$ \begin{aligned} y_G &= -(-y_L + x_L\tan\psi_L) = y_L - x_L\tan\psi_L \overset{\text{Differentiation}}{\Rightarrow} \\ \dot{y}_G &= \dot{y}_L - \dot{x}_L\tan\psi_L - x_L\frac{\dot{\psi}_L}{(\cos\psi_L)^2} \overset{7.2}{\Rightarrow} \\ \dot{y}_G &= u_L\sin\psi_L - u_L\cos\psi_L\tan\psi_L - \frac{x_L}{(\cos\psi_L)^2}r_L \end{aligned} \tag{7.46} $$

thus, leading to:

$$ {}^{G}\dot{y}_G = -\frac{{}^{G}x_L}{(\cos{}^{G}\psi_L)^2}r_L \tag{7.47} $$

where $r_L = \dot{\psi}_L = {}^{G}\dot{\psi}_L$ and $\psi_L = {}^{G}\psi_L$. Thus, based on Eq. 7.45, it stands that:

$$ u_L = -\frac{{}^{G}x_L\sin{}^{G}\psi_L}{(\cos{}^{G}\psi_L)^2}\,{}^{G}\dot{\psi}_L + \cos{}^{G}\psi_L{}^{G}\dot{x}_L + \sin{}^{G}\psi_L{}^{G}\dot{y}_L \tag{7.48} $$

Next, we solve Eq. 7.44 for w_L and we get $w_L = {}^G\dot{z}_G + {}^G\dot{z}_L$, while the same equation with respect to the Follower solved for w_F would lead to $w_F = {}^G\dot{z}_G + {}^G\dot{z}_F$. Finally, based on the previous relations and the kinematic models of the vehicles we get that:

$$\begin{aligned} \dot{x}_L &= u_L \cos {}^G\psi_L \\ \dot{y}_L &= u_L \sin {}^G\psi_L \\ \dot{z}_L &= \dot{z}_F - {}^G\dot{z}_F + {}^G\dot{z}_L \\ \dot{\psi}_L &= {}^G\dot{\psi}_L \end{aligned} \tag{7.49}$$

Chapter 8

Simulation

The theory developed in the preceding chapters was tested through a simulation procedure, which is presented below. The simulation testing was performed using Matlab. The main script that executes the simulation calls the following modules:

- **Leader Motion Controller**, which provides the control inputs for the Leader and is implemented by the `leaderController` function.
- **Follower Motion Controller**, which provides the control inputs for the Follower and is implemented by the `followerController` function.
- **Switching Scheme module**, which executes the switching scheme for the saw-like motion of the Leader vehicle and is implemented by the `switching_scheme` function.
- **Planner**, which feeds the Leader Motion Controller with the desired waypoints of the meander trajectory. The Planner does not exactly correspond to a distinct module but is rather implemented among the lines of the main script of the simulation.
- **State-to-Lasers module**, which provides the centres of the laser dots produced by both vehicles in the Follower image frame based on the configuration of the vehicles w.r.t. the world coordinate system (WCS); it is implemented by the `state2laserCentresFULL` function.
- **Lasers-to-State module**, which performs the state estimation procedure based on the results of the previous module and it is implemented by the `laserCentres2visionState` function.
- **State Velocity module**, which calculates the derivative of the states for both vehicles simply by differentiating and it is implemented by the `poseVelocity` function.
- **Filter module**, which filters out any spikes produced by the previous differentiation procedure using a simple averaging filter and it is implemented by the `simpleFilter` function.

The previous modules will be described in more detail in Appendix C, while their interconnection is presented in Fig. 8.1. Regarding the simulation procedure, three hypotheses were made:

- First, it is assumed that the control loop time is the same for both vehicles, although in reality the vehicles move independently and their control loop times may differ.
- Second, the control input at time i equals to the corresponding velocity at time $i+1$. This means that the vehicle acquires instantly the velocity imposed by the corresponding control input.

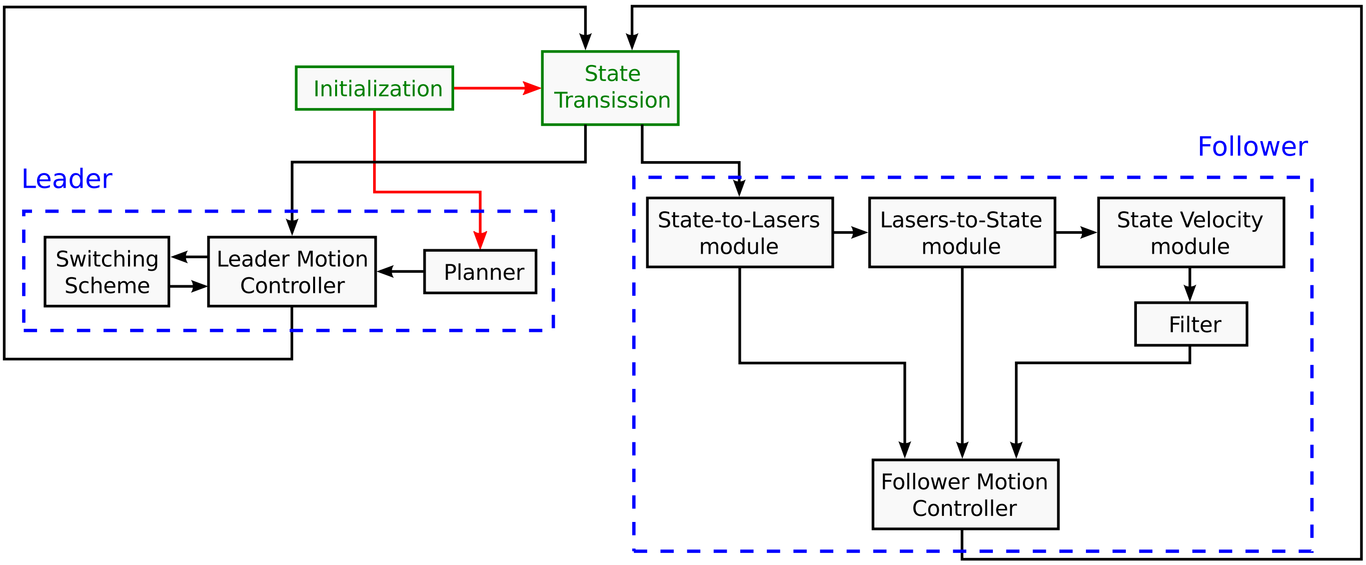


Figure 8.1: The block diagram of the simulation procedure. The interconnection between the modules of the simulation is also presented.

- Finally, it is assumed that the kinematics of the vehicles are completely described by the kinematic models provided at Chapter 2 (e.g. $\dot{z}_F = w_F$) and no dynamics are taken into consideration.

The mathematical background for the *Leader Motion Controller*, *Follower Motion Controller* and *Lasers-to-State* modules has been already presented in the corresponding chapters (Chapter 7, Section 7.1 and 7.2 and Chapter 5, respectively), while the mathematical analysis for the *State Velocity* and *Filter* modules is trivial. The mathematical analysis for the *State-to-Lasers* module will be presented in the next section.

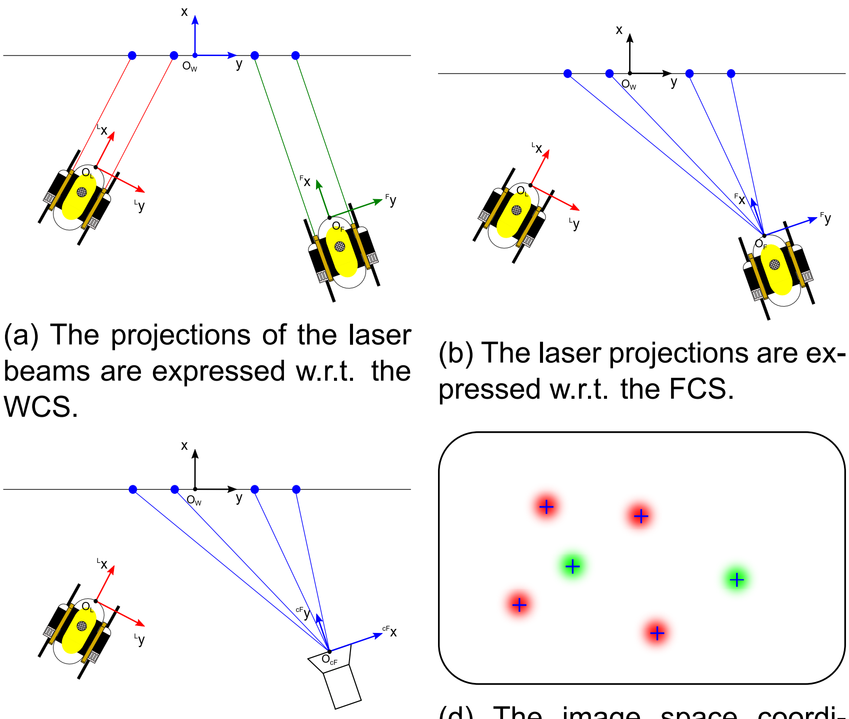
8.1 State-to-Lasers module Mathematical Analysis

During the experimental procedure, the sensing of the Follower vehicle is based purely on vision and no other information is used, e.g. from a position tracker. The purpose of the *State-to-Lasers* module is to provide to the Follower the same sensing information as that provided during the real experiment. The Follower controller uses the centre of the four Leader laser dots in order to provide control inputs with regard to the yaw and heave motion. Thus, the position of the laser dots and their centre inside the Follower's image frame must be calculated. At the same time, the module provides the opportunity to observe aptly and directly if the motion of the vehicles violates the vision constraints. For that purpose, during the simulation, a separate window depicts the position of the laser dots inside the image frame of the Follower. Finally, outside the scope of the simulation, the module provides a way to measure any error that might occur during the localization procedure, in the case that the roll and pitch angles of the vehicles are not exactly equal to zero. This is due to the fact that the module takes into account the whole pose of the vehicles (including their roll and pitch angles) to calculate the position of the laser dots inside the Follower's image frame. This will be made more clear in the following analysis.

The *State-to-Lasers* module follows a procedure that comprises four (4) stages (Fig. 8.2):

- Express the laser dots of the vehicles as projections of their laser beams onto a plane w.r.t. the World Coordinate System (WCS). The plane corresponds to the flat surface.

- Express the laser dots w.r.t. the Follower Coordinate System (FCS).
- Express the laser dots w.r.t. the Follower camera Coordinate System (FcCS).
- Calculate the position of the laser dots in the Follower's image frame based on its camera model.



(a) The projections of the laser beams are expressed w.r.t. the WCS.

(b) The laser projections are expressed w.r.t. the FCS.

(c) The laser projections are expressed w.r.t. the FcCS.

(d) The image space coordinates of the laser dots are calculated.

Figure 8.2: The four stages of the *State-to-Lasers* module.

Stage 1

In the first stage, each laser beam is expressed as a line in 3D-space, while the flat surface is expressed as a plane in 3D-space. The corresponding projections of the beams onto the surface are calculated by solving the system between each 3D-line and the plane (Fig. 8.3). The plane's equation in 3D-space is written in matrix form as:

$$\mathbf{N}^T\mathbf{r} = 0 \tag{8.1}$$

where $\mathbf{N} = [\begin{array}{ccc} 1 & 0 & 0 \end{array}]^T$ because the plane is perpendicular to the x-axis of the WCS (the surface lies in an upright position). The expression of a line in 3D-space is written in matrix form as:

$$\mathbf{r} = \mathbf{a}t + \mathbf{b} \tag{8.2}$$

where $\mathbf{a}$ is the direction vector of the line, while $\mathbf{b}$ is the position vector of a known point that lies on the line. The vector $\mathbf{r}$ in the previous two equations denotes any arbitrary point that belongs to the plane or to the line, respectively.

In general, the laser pointer configuration includes laser pointers that are parallel to the longitudinal axes of the vehicles as well as pointers that form a pitch angle with these axes, as in the case of the Leader. In order to calculate the direction vector of a laser beam, we use the product of two rotation matrices; one corresponds to the rotation of a laser pointer with respect to the vehicle's

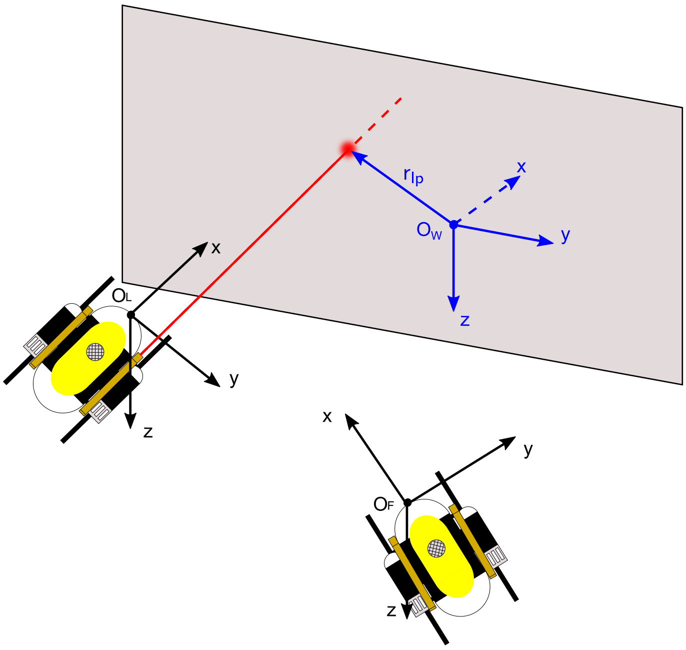


Figure 8.3: The projection of a laser beam onto the surface is calculated as the intersection of the line that represents it and the plane that represents the surface.

coordinate system (denoted as $R_{xyz,config}$) which relates to the pitch angle that is already mentioned and the other corresponds to the rotation of the vehicle with respect to the WCS (denoted as $R_{xyz,rot}$). The resulting rotation matrix can be written as:

$$R_{xyz} = R_{xyz,rot}\ R_{xyz,config} \tag{8.3}$$

The above expression can also be written as:

$$R_{xyz} = \underbrace{R_x(\phi)\ R_y(\theta)\ R_z(\psi)}_{R_{xyz,rot}}\ \underbrace{R_x(\phi_c)\ R_y(\theta_c)\ R_z(\psi_c)}_{R_{xyz,config}} \tag{8.4}$$

where the matrices R_x, R_y, R_z represent the roll, pitch and yaw rotations, respectively, the angles ϕ, θ, ψ correspond to the orientation of the vehicle w.r.t. the WCS, while the angles ϕ_c, θ_c, ψ_c correspond to the orientation of a laser pointer w.r.t. the vehicle's CS. The corresponding direction vector of a laser beam is calculated as:

$$\mathbf{a} = R_{xyz}\ \hat{\mathbf{x}} \tag{8.5}$$

where the vector $\hat{\mathbf{x}} = [\ 1 \quad 0 \quad 0\ ]^T$ represents the direction of the longitudinal axis (x-axis) of the vehicle. For example, the direction vector of a laser parallel to the x-axis of a vehicle can be calculated as:

$$\mathbf{a} = R_{xyz,rot}\ \hat{\mathbf{x}}$$

since $\phi_c = \theta_c = \psi_c = 0$ while the same vector for an inclined laser pointer ($\theta_c \neq 0$) is calculated as:

$$\mathbf{a} = R_{xyz,rot}\ R_y(\theta_c)\ \hat{\mathbf{x}}$$

The position vector $\mathbf{b}$ is chosen to correspond to the point where a laser beam is emitted from. For each laser beam, this point has a known position vector $\mathbf{b}_0$ with respect to the vehicle's coordinate system. The vector $\overrightarrow{b_0}$ must be expressed with respect to the WCS by means of the rotation matrix $R_{xyz,rot}$ and, then, it must be added to the position vector of the vehicle, which is also expressed with respect to the WCS. This is mathematically written as:

$$\mathbf{b} = R_{xyz,rot}\,\mathbf{b_0} + \mathbf{r_v} \tag{8.6}$$

where $\mathbf{r_v} = [\begin{array}{ccc} x & y & z \end{array}]^T$ represents the position vector of the vehicle.

Next, substituting Eq. 8.2 into Eq. 8.1 and solving for t yields:

$$\mathbf{N}^T\,(\mathbf{a}t + \mathbf{b}) = 0 \Rightarrow$$
$$t = \frac{-\mathbf{N}^T\mathbf{b}}{\mathbf{N}^T\mathbf{a}}$$

Substituting t into Eq. 8.2 provides the projection of a laser beam onto the surface with respect to the WCS as:

$$\mathbf{r} = \mathbf{a}\frac{-\mathbf{N}^T\mathbf{b}}{\mathbf{N}^T\mathbf{a}} + \mathbf{b} \tag{8.7}$$

Stage 2

In the second stage, the relative position (Fig. 8.4) between a laser projection (lp) and the Follower (F) w.r.t. the WCS is expressed as:

$$\mathbf{r_{lp}^F} = \mathbf{r_{lp}} - \mathbf{r_F} \tag{8.8}$$

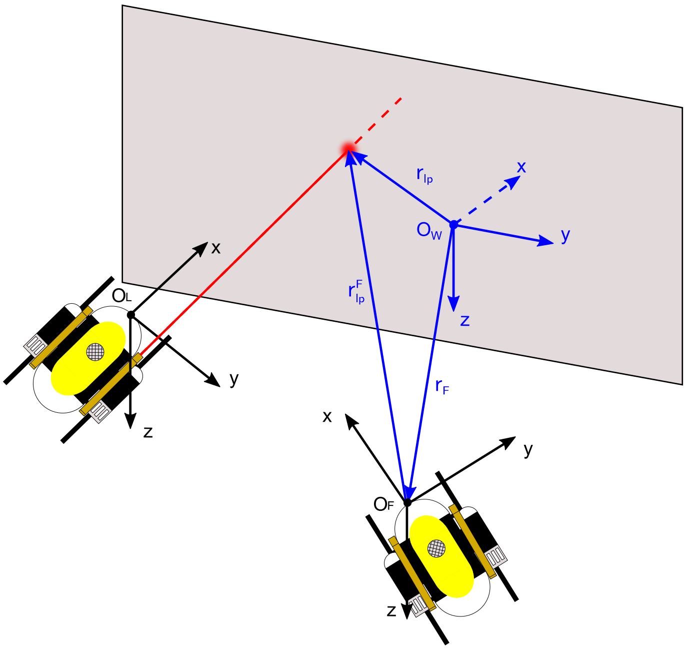


Figure 8.4: The relative position between a laser projection and the Follower with respect to the WCS.

The same relative position is then expressed with respect to the FCS using a rotation transformation as follows:

$${}^{\mathbf{F}}\mathbf{r_{lp}^F} = R_{zyx,F}^{-1}\,\mathbf{r_{lp}^F} = R_{zyx,F}^{T}\,\mathbf{r_{lp}^F} \tag{8.9}$$

where $R_{zyx,F}$ represents the rotation matrix of the Follower vehicle w.r.t. WCS and it is calculated as:

$$R_{zyx,F} = R_z(\psi_F)\,R_y(\theta_F)\,R_x(\phi_F) \tag{8.10}$$

where the angles ϕ_F, θ_F, ψ_F represent the roll, pitch and yaw angles of the Follower vehicle. Thus, the position vector of a laser projection with respect to the FCS is written as:

$$ {}^{\mathbf{F}}\mathbf{r}_{\mathbf{lp}} = R_{zyx,F}^{T}\left(\mathbf{r}_{\mathbf{lp}} - \mathbf{r}_{\mathbf{F}}\right) \tag{8.11} $$

Stage 3

In the third stage, the coordinates of a laser projection are expressed with respect to the FcCS as follows (see also Chapter 2 Section 2.1):

$$ {}^{\mathbf{cF}}\mathbf{r}_{\mathbf{lp}} = \begin{bmatrix} x_{lp} \\ y_{lp} \\ z_{lp} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} {}^{\mathbf{F}}\mathbf{r}_{\mathbf{lp}} \tag{8.12} $$

Stage 4

Finally, in the last stage, the pixel coordinates of each laser projection inside the image frame of the vehicle are calculated using the camera model and the corresponding intrinsic parameters (a_x, a_y, u_0, v_0) of the Follower camera system.

$$ \begin{aligned} u_{lp} &= a_x \frac{x_{lp}}{y_{lp}} + u_0 \\ v_{lp} &= -a_y \frac{z_{lp}}{y_{lp}} + v_0 \end{aligned} \tag{8.13} $$

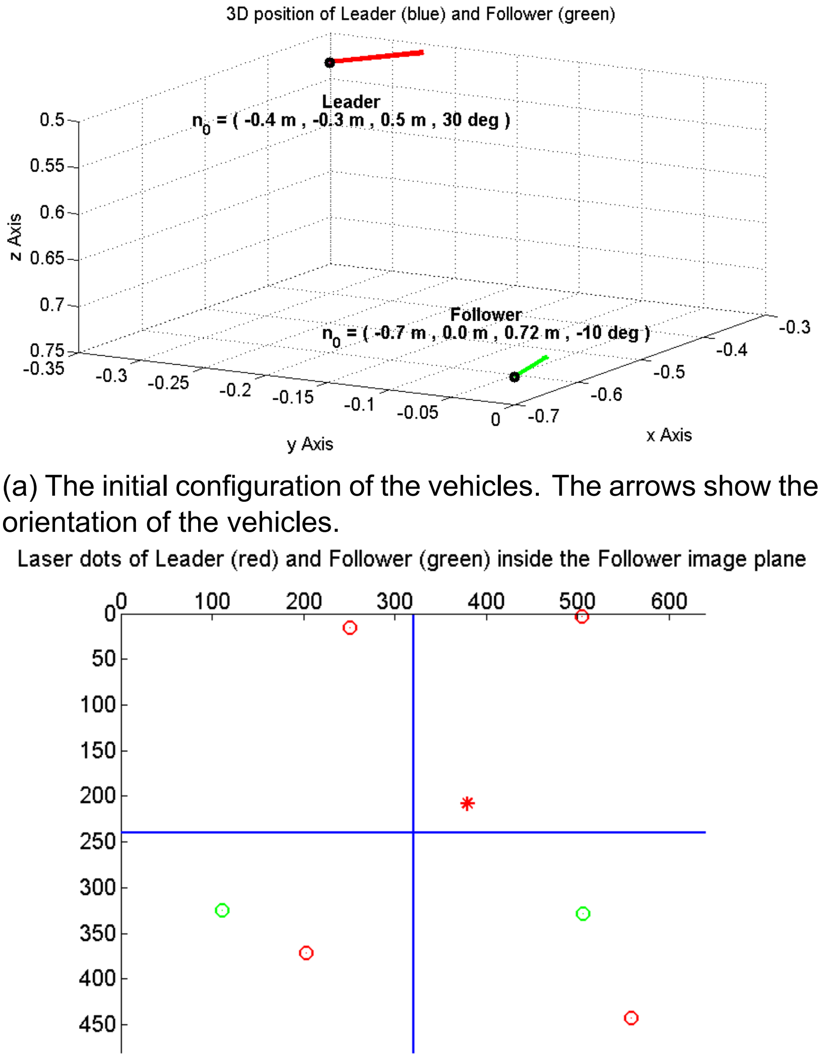
8.2 Simulation Results

This section presents the results of the simulation procedure. The Leader begins from an initial configuration $[\ -0.4\text{m} \quad -0.3\text{m} \quad 0.5\text{m} \quad 30° \ ]^T$ with respect to the WCS while the Follower's initial configuration is $[\ -0.7\text{m} \quad 0.0\text{m} \quad 0.72\text{m} \quad -10° \ ]^T$. The initial configuration of the vehicles is also presented in Fig. 8.5a while the laser dots inside the Follower's image frame for that initial configuration are shown in Fig. 8.5b.

The Leader must traverse a meander where each segment has a width of 0.2m and a height of 0.25m on the yz-plane. On the xy-plane, the Leader must move between two bounds as explained in Chapters 3 and 7 that are 0.2m apart. The Follower must maintain a vertical distance offset of 0.9m with respect to the surface (thus $x_d = -0.9$) and a lateral distance offset of 0.5m with respect to the Leader (thus $y_d = 0.5$). At the same time, the centre of the Leader laser dots must be kept at all times at the centre of the Follower image frame (thus $u_d =$ WIDTH$/2$ and $v_d =$ HEIGHT$/2$). The loop time is considered for both vehicle to be 150ms. The results of the simulation are presented in Fig. 8.6 through 8.11.

These figures prove the efficiency of the theory. From Fig. 8.7, it is concluded that the Leader vehicle can reach the desired waypoints with the requested accuracy, while moving almost in parallel to the surface, despite its nonholonomic constraints. In the same figure, the permissible error at which the Leader must reach a waypoint is denoted by a cylinder rather than a sphere. This is due to the fact that the proposed controller regulates only the errors along the y- and z-axes between the vehicle and the waypoint and, thus, only these errors are considered. In Fig. 8.8, it is shown that the motion of the Leader is strictly bounded between the defined bounds.

For the case of the Follower, it is shown in Fig. 8.9 that no laser dot leaves its image frame, while the centre of the Leader laser dots (small magenta square) converges near the centre of the image frame. Finally, from Fig. 8.10 and 8.11 it can be concluded that the Follower reaches



(a) The initial configuration of the vehicles. The arrows show the orientation of the vehicles.

(b) The position of the laser dots inside the Follower's image frame in the initial configuration.

Figure 8.5: Initial Configuration.

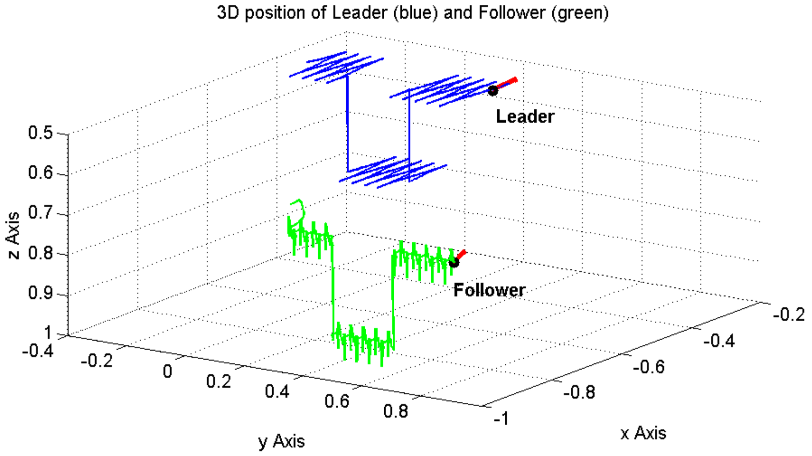


Figure 8.6: The motion of the Leader and the Follower in 3D-space.

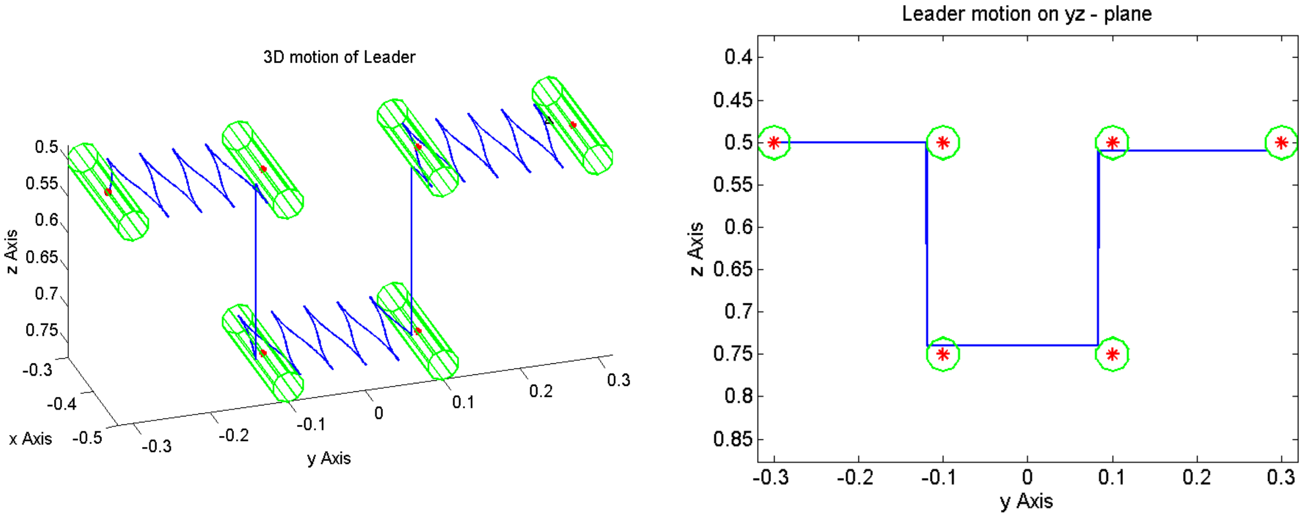


Figure 8.7: The motion of the Leader in 3D-space. Red circles denote the waypoints that the Leader must reach. The cylinders around them denote the error at which the Leader must approach the waypoints.

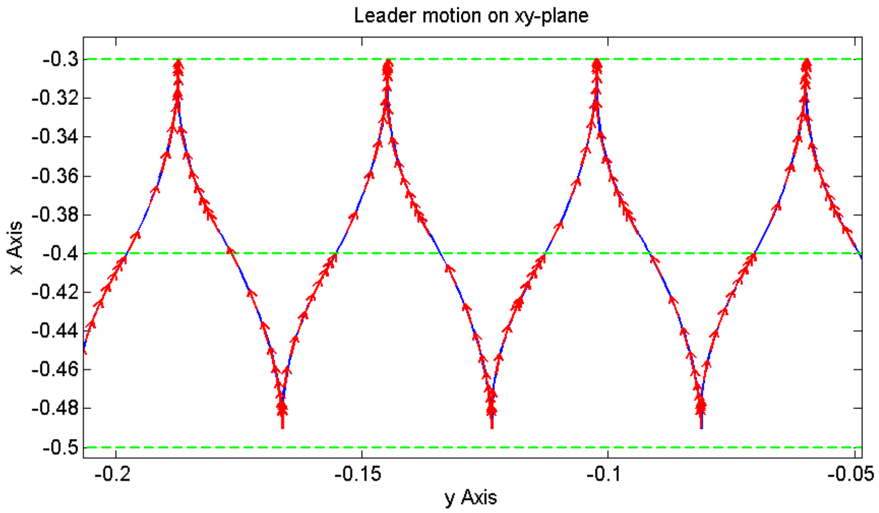


Figure 8.8: The motion of the Leader on the xy-plane. The red dashed lines represent the bounds between which the Leader must move.

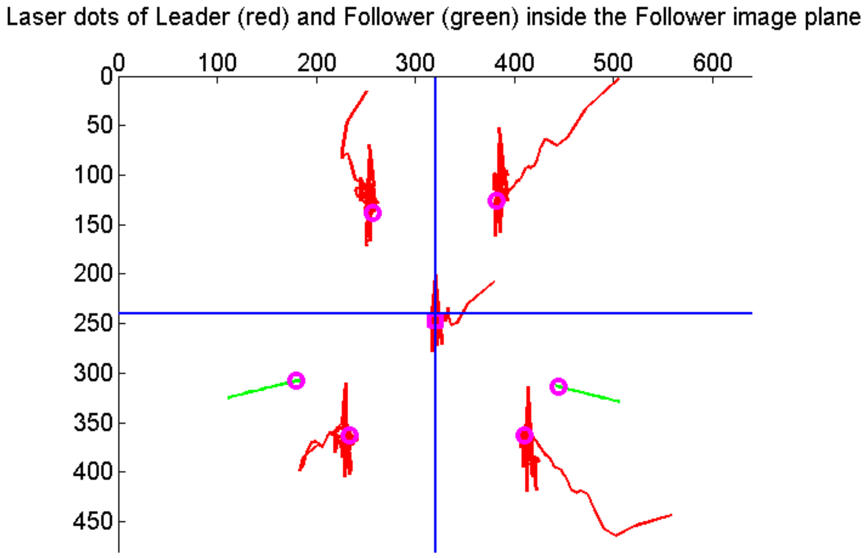


Figure 8.9: The positions of the laser dots inside the image frame of the Follower during the motion of the vehicles.

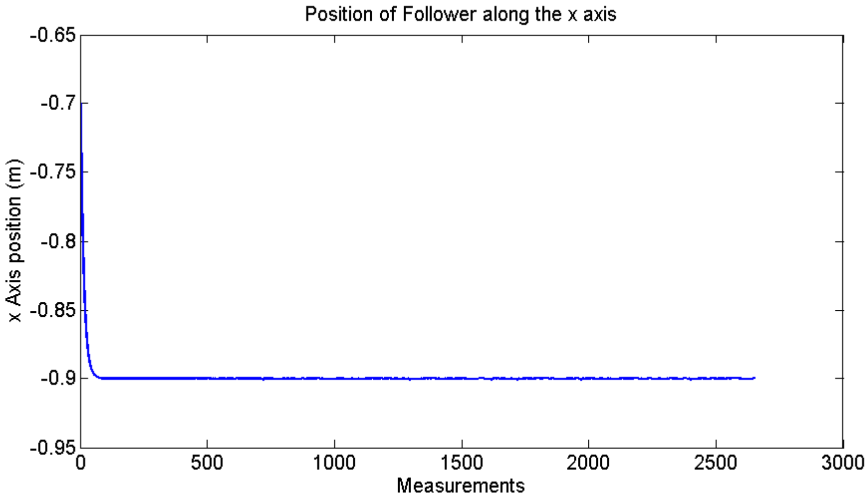


Figure 8.10: The position of the Follower along the x-axis of the WCS.

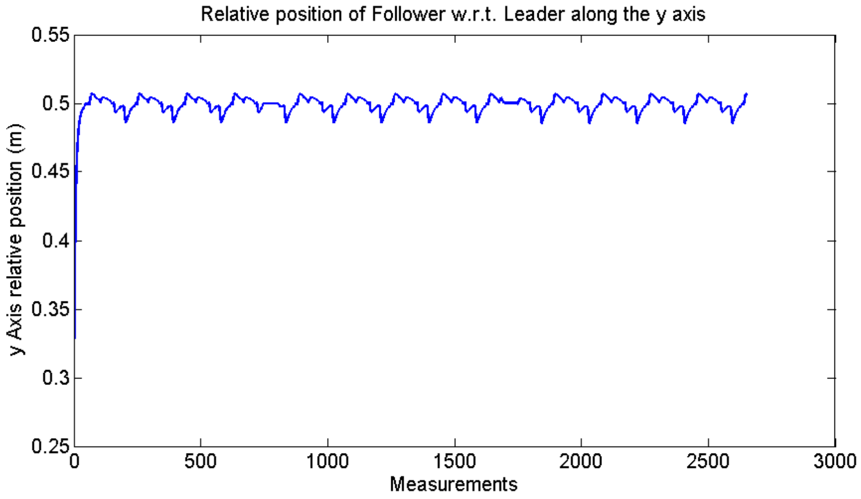


Figure 8.11: The relative position of the Follower w.r.t. the Leader along the y axis of the WCS.

successfully both the desired distance from the surface x_d and the relative lateral distance to the Leader y_d.

Chapter 9

Experimental Setup

The experimental setup can be examined by describing five (5) distinct entities. These are listed below and are described in detail in the following sections:

- Remotely Operated Vehicles (ROVs)
- Sensing Equipment
- Laser Pointer Setup
- Software Platform
- Environment of Operation

9.1 Remotely Operated Vehicles (ROVs)

During the experiments, two Remotely Operated Vehicles (ROVs) were used, namely a Videoray PRO (Fig. 9.1) and a Seabotix LBV (Fig. 9.2). The Videoray PRO ROV has three (3) thrusters, two of which provide motion along the longitudinal axis of the vehicle (x-axis), i.e. *surge* motion u and rotation about its vertical axis (z-axis), i.e. *yaw* motion r. The third thruster provides motion along the vertical axis of the vehicle (z-axis), i.e. *heave* motion w. The vehicle has no thruster to provide motion along its lateral axis (y-axis), i.e. *sway* motion v. As a result, it is underactuated along its sway axis and, thus, it is subject to nonholonomic constraints. Also, the vehicle is equipped with an on-board Charged-Coupled Device (CCD) camera, which is not used during the experiments.

The Seabotix LBV ROV has the same setup of thrusters as the Videoray PRO ROV plus an additional one for lateral motion (*sway* v), which leads to a total of four (4) thrusters. As the vehicle is actuated along its sway axis, it is not subject to nonholonomic constraints concerning its motion

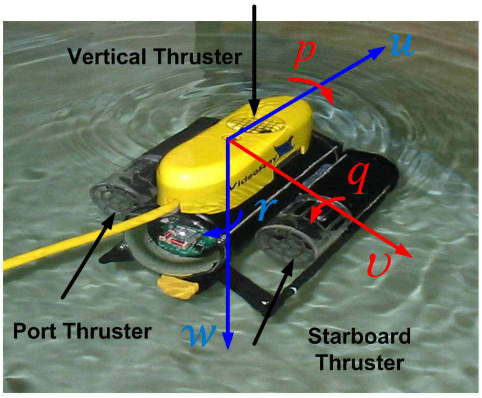


Figure 9.1: VideoRay PRO ROV. The underactuated Degrees-of-Freedom (DOFs) are denoted in red and the fully actuated DOFs in blue colour.

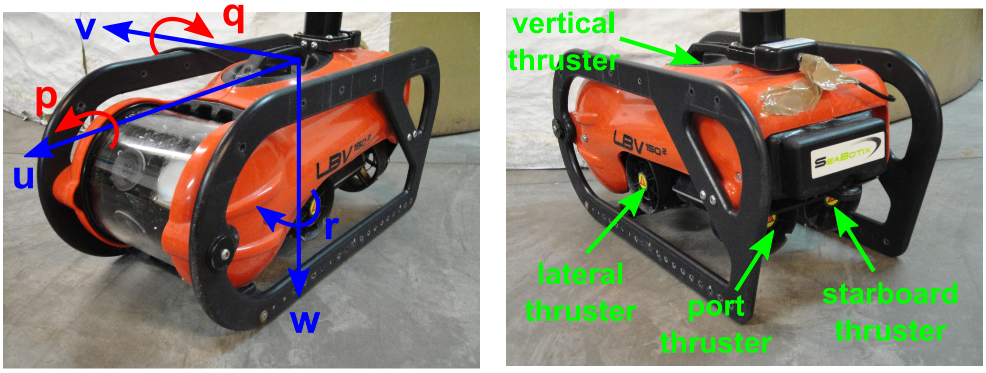


Figure 9.2: Seabotix LBV ROV. The underactuated DOFs are denoted in red and the fully actuated DOFs in blue colour.

on the xy-plane. There is a coupling though between its sway (lateral) and its yaw motion. The vehicle is also equipped with an on-board CCD camera as well as with red laser pointers which are placed in parallel and in close distance with each other. During the experiments, neither the camera nor the laser pointers are used. Instead, a webcam inside a well-insulated casing is used to improve the performance of the vision algorithm, while a new setup of laser pointers is equipped on the vehicle to improve the accuracy of the localization procedure.

Both vehicles are underactuated about surge and sway axis (pitch and roll motions), but they are designed to be statically stable about these axes and, thus, the corresponding roll and pitch angles are considered equal to zero at all times. The vehicles are also equipped with depth sensors and magnetic compasses.

9.2 Sensing equipment

The vehicles are equipped with sensors that provide positioning information that is used as input for the motion controllers. The Leader vehicle (Videoray PRO ROV) is equipped with a Polhemus Isotrak system (Fig. 9.3). The Polhemus Isotrak II position sensor is used as a reference sensor to evaluate the proposed control algorithm for the Leader. It is also used as a reference sensor for the motion of the Follower and the output of its vision system. The Isotrak system consists of a transmitter and a receiver and its operating principle is based on A/C magnetics. The transmitter is placed at known distance with respect to the surface (Fig. 9.4) and the two receivers are placed on the vehicles and at known distance from the centre of each camera (Fig. 9.5a and9.5b). The known distances provide the necessary offsets so that the comparison of the measurements from the systems and the reference can be as accurate as possible. The system provides accurate measurements within a sphere of radius of approximately 76cm at an update rate of 60 updates/second, if one receiver is attached, and 30 updates/second, if two receivers are used. The vehicle is also equipped with a setup of laser pointers which is described in Section 9.3.

The Follower vehicle (Seabotix LBV ROV) is equipped with a Laser Vision System that provides localization information with respect to the surface. In particular, the LVS of the Follower provides the mean range of the vehicle and its angle with respect to the surface (see also Chapter 5). The LVS of the Follower (Fig. 9.6a) comprises a USB camera that provides visual information at 25 fps and is placed inside a well-insulated casing, two green underwater laser pointers that are equipped in parallel to each other on the port and the starboard of the vehicle and the vision algorithm that was described in Chapter 6. When the Follower LVS is combined with the laser setup equipped on the Leader (Fig. 9.6b and 9.6c), then the entire laser system can be referred to as Cooperative Laser Vision System (CLVS), which provides to the Follower relative localization information with

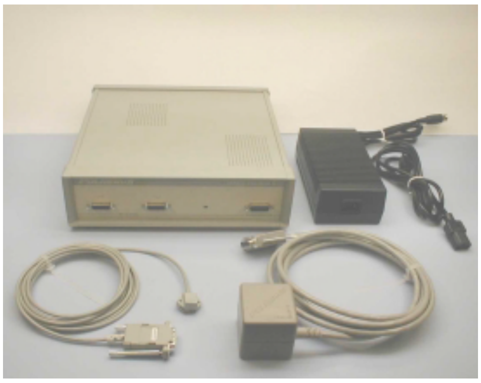


Figure 9.3: The Polhemus Isotrak system.

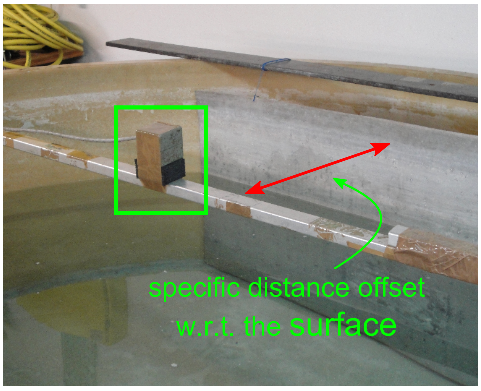
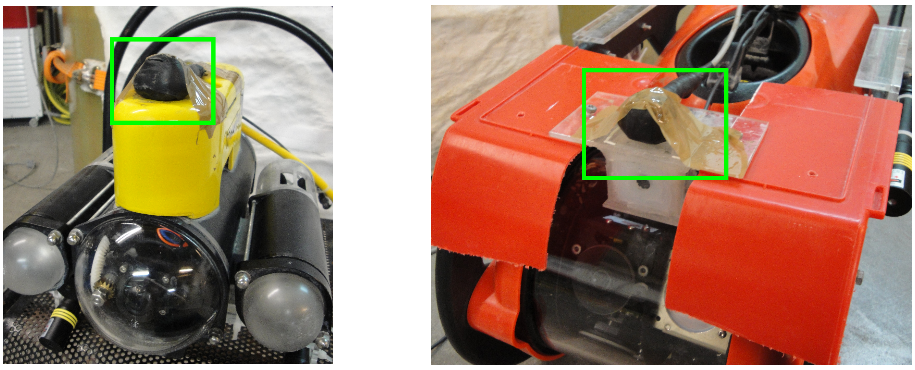


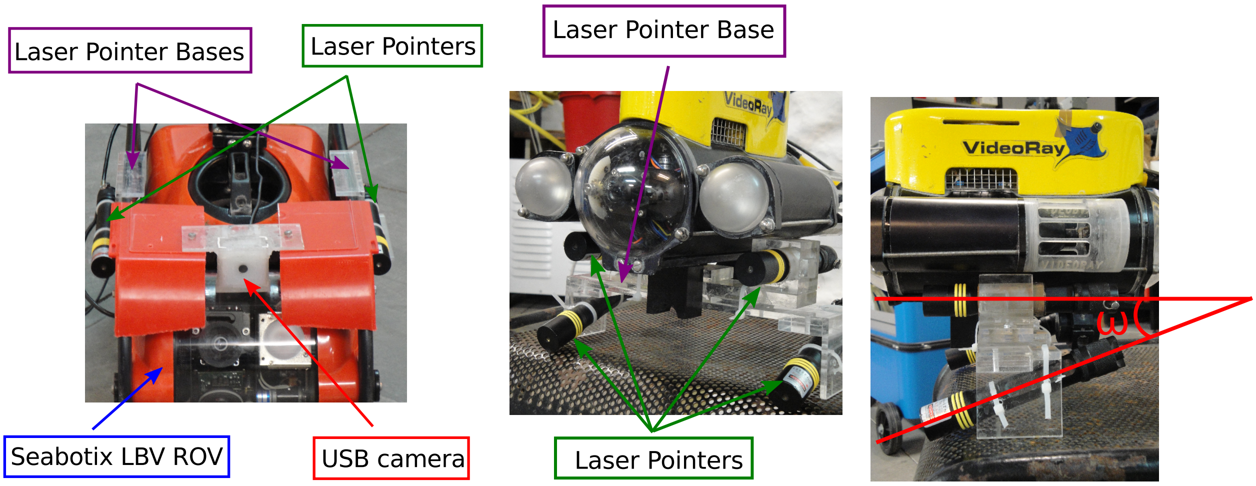
Figure 9.4: The Isotrak transmitter as placed with respect to the surface.



(a) An Isotrak receiver as placed with respect to the camera of the Leader.

(b) An Isotrak receiver as placed with respect to the camera of the Follower.

Figure 9.5: The Isotrak receivers setup.

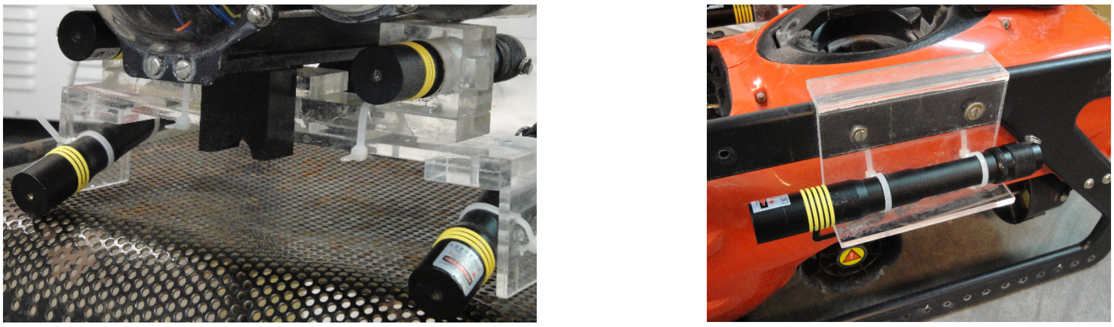


(a) The Laser Vision System (LVS) of the Follower.

(b) The laser setup equipped on the Leader.

(c) The Leader laser setup viewed from the side.

Figure 9.6: The Cooperative Laser Vision System (CLVS).



(a) The Leader laser pointer setup.

(b) The Follower laser pointer setup.

Figure 9.7: The laser pointer setup equipped on the vehicles.

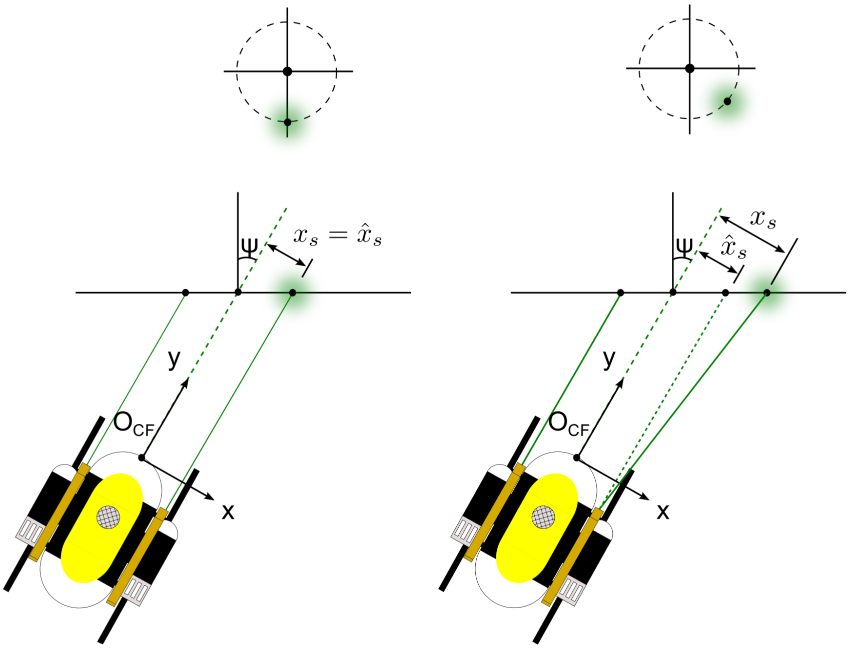
respect to the Leader. The laser pointer setup of the Follower is described in more detail in Section 9.3.

9.3 Laser Pointer Setup

As described in the previous section, each vehicle is equipped with a laser pointer setup that provides localization information to the Follower. The pointers that are used for the Leader setup have a wave length of 650 nm (red), while the laser pointers of the Follower setup have a wave length of 532 nm (green). All laser pointers have an output power of 5 mW.

In order to equip the laser pointers onto the vehicles, certain structures had to be made. These structures are shown in Fig. 9.7. The structure of the Leader comprised two parts. The upper part was already constructed within the scope of [22]. The lower part was constructed within the scope of this work in order to accomodate the two inclined laser pointers. The Follower construction was also part of this work and its shape guaranteed that the laser pointers would be equipped accurately at the desired position, i.e. in parallel to the longitudinal axis of the vehicle and its camera.

Another important issue that had to be taken into account, was the fact that the beams projected from the laser pointers deviated by a small angle from their intended course. The only way to alleviate this problem was to position the lasers as described below.



(a) The laser pointer is positioned in such a way that the dot lies vertically below the centre of the circular locus.

(b) The laser pointer has a random position.

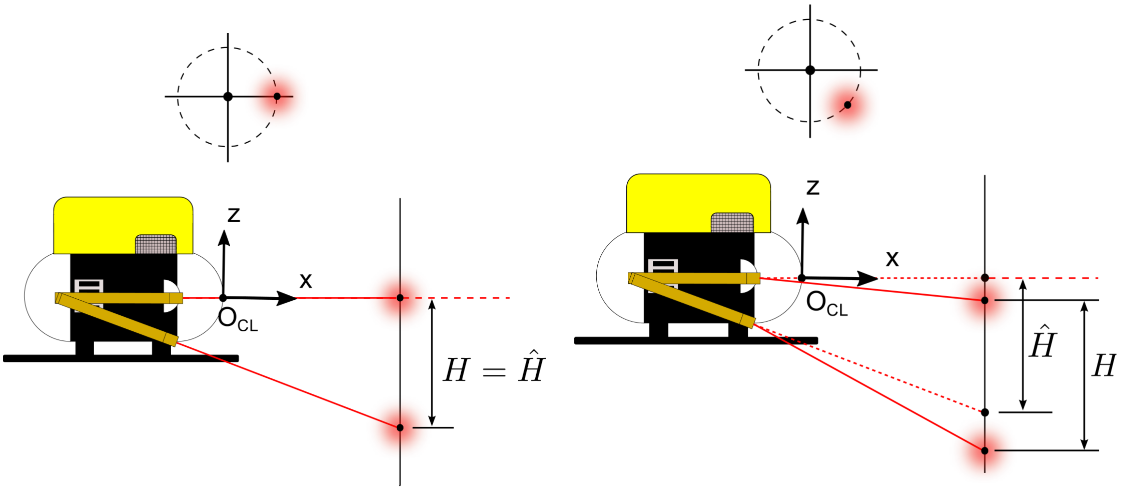
Figure 9.8: Laser deviation problem for the case of the Follower.

Let us assume that a laser pointer was placed at a horizontal position and projected its beam vertically on a flat surface. If the pointer was turned about its axis, the successive positions of its dot would form a locus of a circle. Now let us consider the case of the Follower depicted in Fig. 9.8. In this figure, $\hat{x}_s$ denotes the distance of a laser dot from the centre of the camera along the x-axis as used in the algorithms and x_s its real value.

If the laser pointer was placed as in Fig. 9.8b, x_s and $\hat{x}_s$ would not be equal and based on Chapter 5 and Eq. 5.3 the corresponding ranges would be false. Instead, the laser pointer was placed as in Fig. 9.8a, where x_s and $\hat{x}_s$ match. It is obvious that in the last case the z-coordinate of the dot would deviate from its real value, but z-coordinates are not used while calculating the ranges for the Follower and, thus, this error is ignored.

The case of the Leader is depicted in Fig. 9.9. Let $\hat{H}$ denote the vertical distance between the laser dots of a pair of laser pointers (equipped on port or staboard) if no deviation was present and H its real value. As in the previous case, if the lasers were placed at random (Fig. 9.9b), $\hat{H}$ and H would not be the same, which would lead to localization errors. The appropriate configuration is shown in Fig. 9.9a. In that case $\hat{H} = H$. Unfortunately, the complete localization procedure includes also the x- and y-coordinates of the laser dots with respect to the FcCS, which do not correspond to their ideal values, i.e. without deviation, in the proposed setup.

In particular, it is assumed that the centre between the two upper laser dots of the Leader coincides with point O_G (see also 2) and, thus, the camera axis passes through it. It is obvious that, due to the deviation of the laser beams, this was not the case. A solution to this problem is shown in Fig. 9.10. According to that figure, the two horizontal lasers were placed in symmetrical positions with each other. When $\psi = 0$, the error is nonexistent, because the deviation is symmetrical and the centre does not change. As the yaw angle ψ changes, the error augments, but, since the leader motion control scheme bounds the orientation of the Leader, the error is bounded.



(a) The laser pointer is positioned in such a way that the dot lies horizontally right to the centre of the circular locus.

(b) The laser pointer has a random position.

Figure 9.9: Laser deviation problem for the case of the Leader.

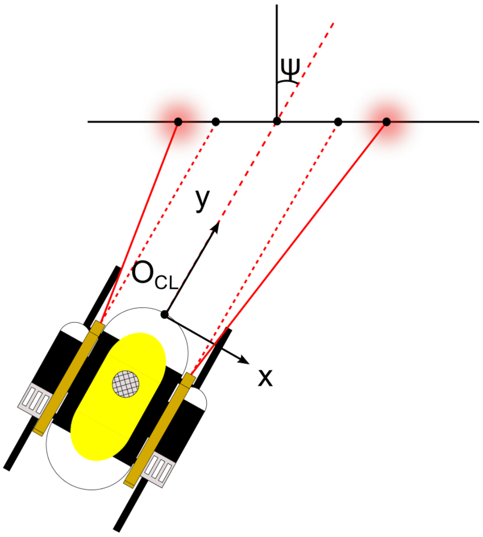


Figure 9.10: Proposed configuration of lasers for the Leader in order to improve localization results.

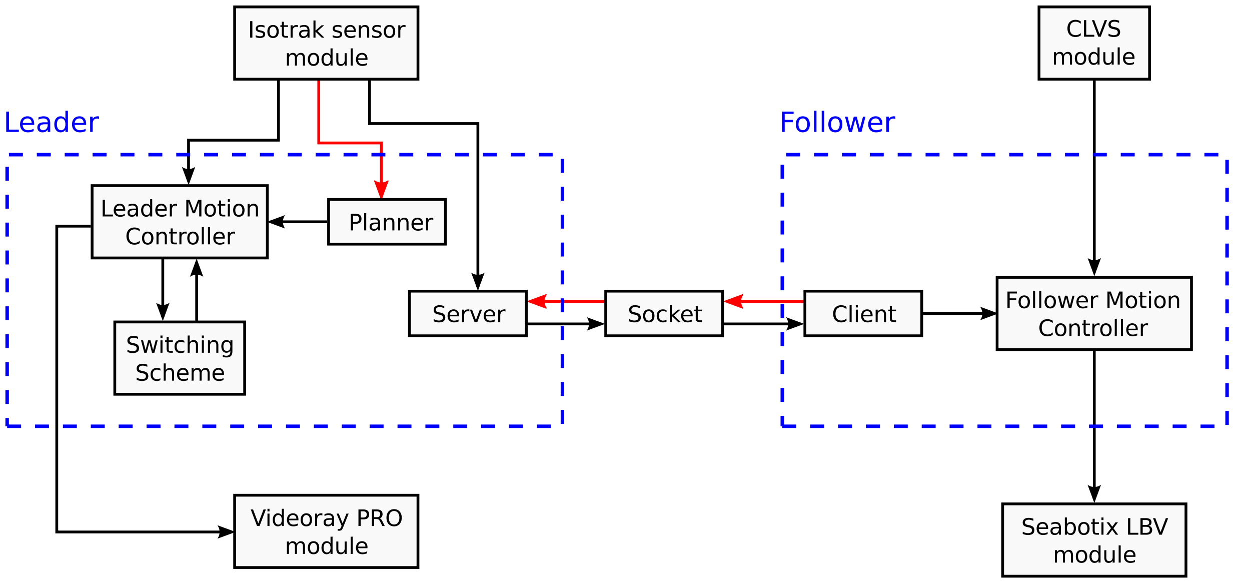


Figure 9.11: The interconnection between the modules of the software platform that was used during the experimental procedure. The red arrows denote interconnections that take place only during the initialization phase.

9.4 Software Platform

The software platform comprises several modules, as it is depicted in Fig. 9.11. These modules can be listed as follows:

- Isotrak sensor module
- CLVS module
- Leader Motion controller
- Follower Motion controller
- Planner
- Switching Scheme module
- Socket module
- Videoray PRO module
- Seabotix LBV module

The *Isotrak sensor module* initialises the Isotrak system and receives position measurements from the Isotrak sensor. The *Leader Motion Controller* receives these measurements and provides as output the control inputs for the motion of the Leader. In order to do so, it exchanges information with the *Switching Scheme module* which implements the saw-like motion of the Leader. The control inputs of the Leader provided by the Leader Motion Controller are fed to the *Videoray PRO module*. This module transforms the control inputs of the Leader to appropriate inputs for its thrusters and sends the information to the vehicle.

The *Planner* is not exactly a distinct module, but rather it is implemented among the lines of the Leader software code. It is initialized by the initial measurement of the Isotrak system. This initial measurement defines the waypoints that the Leader vehicle must follow. The module uses a for-loop structure in order to provide to the Leader Motion Controller a new waypoint at each loop. Each loop of the Planner begins when the Leader reaches the previous waypoint with the prescribed accuracy.

The *CLVS module* comprises several submodules, namely the *Laser Detection module*, the *Laser Sorting module*, the *State Estimation module* and the *Velocity Calculation module*. The

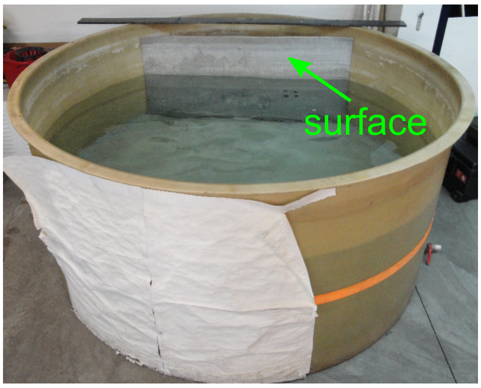


Figure 9.12: The water tank where the experiments took place.

Laser Detection module receives a frame from the USB camera of the Follower vehicle and detects the laser dots inside that image frame. It also filters any noise produced by the laser detection procedure. The Laser Sorting module receives as input the image frame coordinates of the detected laser dots. Then, it sorts the laser dots into Top Left, Bottom Left, Top Right and Bottom Right for the case of the Leader laser dots (4 laser dots) and into Left and Right for the case of the Follower (2 laser dots). The sorted image frame coordinates are fed to the State Estimation module which calculates the pose of the vehicles with respect to the GCS. The Velocity Calculation module calculates the derivatives of the poses of the vehicles through a simple differentiation procedure. The output of the module is passed through a simple averaging filter to eliminate any noise produced by the differentiation procedure.

The Laser Detection module, the State Estimation module and the Velocity Calculation module provide as information the image coordinates of the Leader laser dots centre, the states of the vehicles with respect to the GCS and the corresponding rates of change for these states, respectively. This information is then fed to the *Follower Motion Controller*, which calculates the control inputs for the motion of the Follower vehicle and feeds them to the *Seabotix LBV module*. The module translates these velocity inputs into inputs for the thrusters of the vehicle and sends them directly to the vehicle using its motor control interface.

The *Socket module* implements the socket interconnection between the Leader and the Follower software and it comprises the *Server* and the *Client* modules which correspond to the Leader and the Follower, respectively, as shown in Fig. 9.11. The module is used during the initialization step for the synchronisation of the two main programs in order to begin at the same time. It also provides to the Follower the positioning information from the Isotrak system in order to compare the measurements of the CLVS with a reference and to calculate the Follower heave velocity $\dot{z}_F$ with respect to the WCS, which is necessary for the derivation of the control inputs.

9.5 Environment of Operation

The environment of the experiments is that provided by the small water tank that is depicted in Fig. 9.12. The tank is cylindrical with a radius of 2.10m and a height of 1.0m. During the experiments, the tank was filled with water up to a height of 80cm. Inside the tank lies a flat aluminum plate, as depicted also in Fig. 9.12, with dimensions 130cmx90cm. The laser pointers of the vehicles project their beams onto the aluminum plate, thus producing distinct laser dots that the vision algorithm of the Follower can detect.

Chapter 10

Experimental Procedure & Results

This chapter presents the experimental procedure and the corresponding results related to the cooperation scheme analysed in the previous chapters. The final experiment involves the cooperative motion of two underwater vehicles inside a small water tank in front of a flat surface using a vision-based algorithm and a laser pointer setup in order to use implicit communications between the vehicles.

The task is complex and the system that implements the cooperation procedure comprises several subsystems and submodules. These refer to the laser vision system (also referred to as Cooperative Laser Vision System, CLVS), which includes both the vision algorithm and the laser pointer setup, the Leader controller, the Follower controller as well as the underwater vehicles themselves. A brief diagram that shows the interconnection between the various subsystems during the experimental procedure is given in Fig. 10.1.

In order to execute correctly and efficiently the experimental procedure, a unit test approach was followed. Each module of the subsystem was tested separately. Thus, the testing procedure was planned to comprise several stages, which can be described as follows:

- Testing of the **Laser Vision System**, in order to guarantee that it provides accurate localization information.
- Testing of the **Leader Motion Controller**, in order to guarantee that the Leader vehicle approaches accurately the prescribed waypoints and produces a well defined meander-like trajectory.
- Testing of the **Follower Motion Controller** when the Leader does not move, in order to guarantee that the Follower controller can, at least, stabilize the Follower around its desired pose with respect to the Leader and to the surface.
- **Final testing** of the entire system (Leader is moving).

The experimental procedure of each stage is described in the next sections. In each section, the experimental results are also presented, along with some comments. In this chapter, we present experimental results that relate only to the Laser Vision System and to the Leader controller. The experiments that concern the remaining stages can be considered as one joint experiment (Testing of the **Cooperation Scheme**) and are divided here for clarity. They also represent part of the future work related to this thesis.

10.1 Laser Vision System Experiments

The experimental procedure for the testing of the Laser Vision System comprised two distinct stages. In the first stage, only the localization of the Follower vehicle with respect to the surface

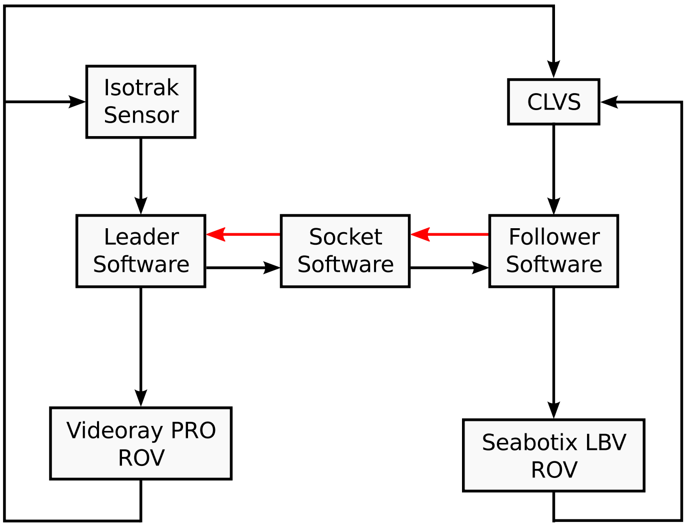


Figure 10.1: The interconnection between the systems during the experimental procedure.

was examined. In the second stage, the localization procedure for both vehicles was considered. The vehicles were placed inside a water tank where a flat surface was fixed in an upright position.

During both stages, the motion of the vehicles as that would be in the final experiments was not taken into consideration. The system is sensitive to roll and pitch deviations from 0, since the relative localization procedure is entirely designed under the premise that $\phi = \theta = 0$ for both vehicles. Thus, the laser vision system provides erroneous information if that premise is violated. For that reason, both vehicles were placed on stable bases inside the water tank, in such a way so as to guarantee that the roll and pitch angles of the vehicles were as close to zero as possible. Then, the system was activated and measurements were taken, first, while the vehicles were stationary and then, while moving each of the vehicles separately. In particular, during the first stage the Leader was not present inside the water tank and was not considered. Instead, only the Follower was moved in various positions with respect to the surface. In the second stage, both vehicles were placed inside the tank. The Follower remained almost stationary and the Leader was moved in various positions.

During the whole procedure, any motion of the vehicles was performed so that none of the laser dots would leave the Follower image frame. The corresponding results are presented in Fig. 10.5 through 10.12. In those figures, the results provided by the vision system itself (red colour) are compared directly to those provided by the Isotrak system (blue colour), which is used as a reference for the validation of the results.

Fig. 10.2 through 10.4 show results from the first stage (only Follower). Since the Leader vehicle was not considered, only the mean range, the yaw orientation and the x-axis coordinate of the Follower is presented (no relative y- or z-axis coordinates could be calculated). By observing these figures, it can be concluded that the laser setup of the Follower provided fairly accurate localization results. More precisely, the vision measurements of the mean range and the x-axis position deviate from the Isotrak measurements by $5-6$cm, while the corresponding deviation for the yaw orientation is $3-5°$.

On the other hand, Fig. 10.5 through 10.12 depict the results of the second stage (both vehicles). The deviations of the vision measurements from the Isotrak ones lie again in almost the same intervals. In particular, the deviation for the Leader mean range is $2-4$cm, for the Follower mean range $5-6$cm, while for the yaw orientations the error is $5-6°$. As for the relative position of the two vehicles (Leader x-axis position, Follower x-axis position, relative y-axis and relative z-axis position) the maximum error was not greater than 6cm.

The deviations presented in the above mentioned measurements can be considered as in-

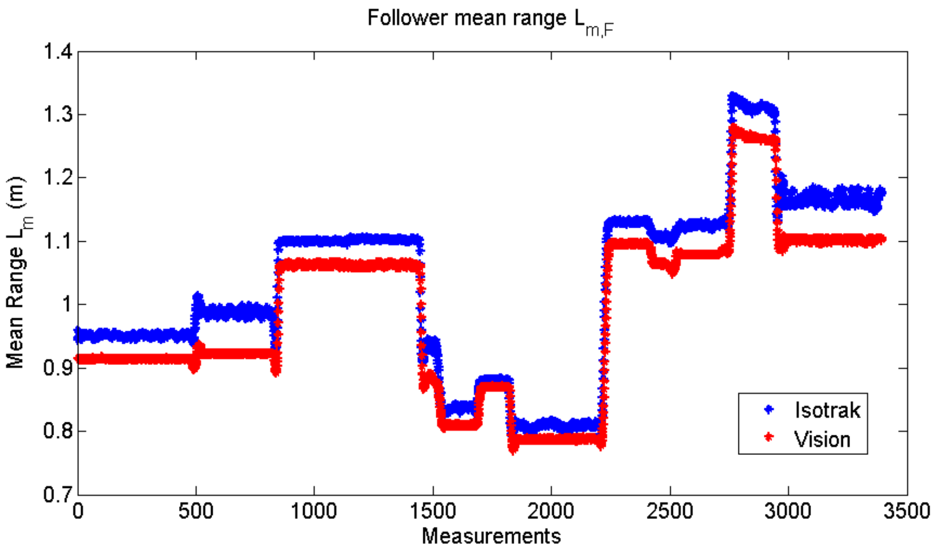


Figure 10.2: The mean range of the Follower Lm, F with respect to the surface when no Leader is considered.

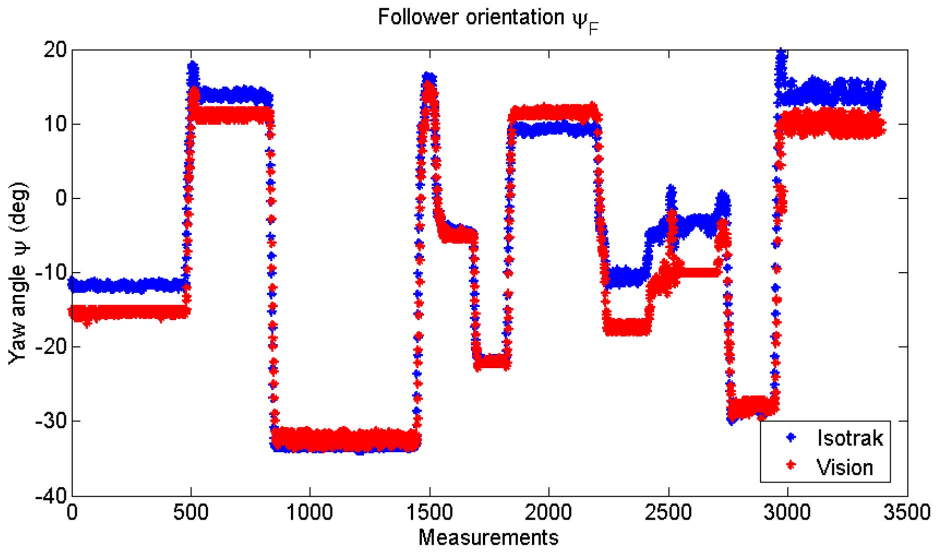


Figure 10.3: The orientation of the Follower ψ_F with respect to the surface when no Leader is considered.

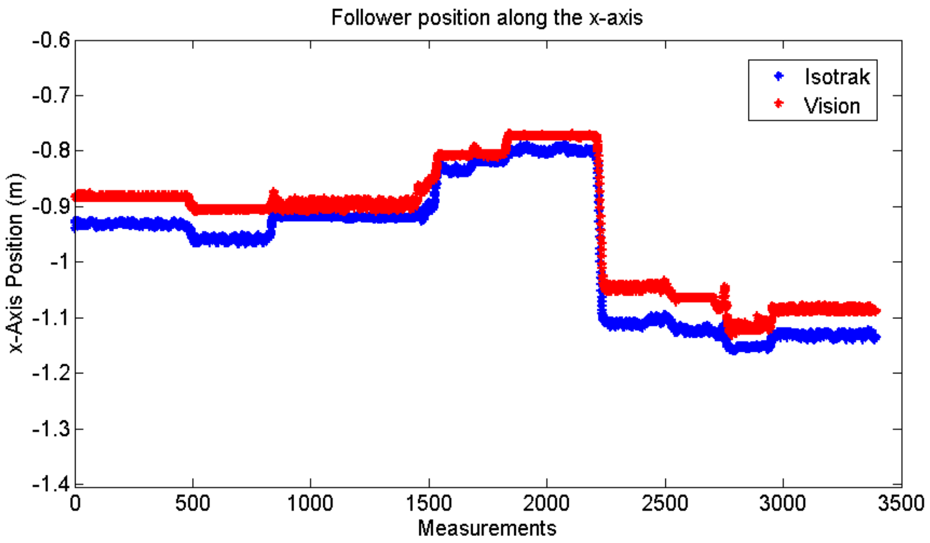


Figure 10.4: The position of the Follower along the x-axis of the WCS when no Leader is considered.

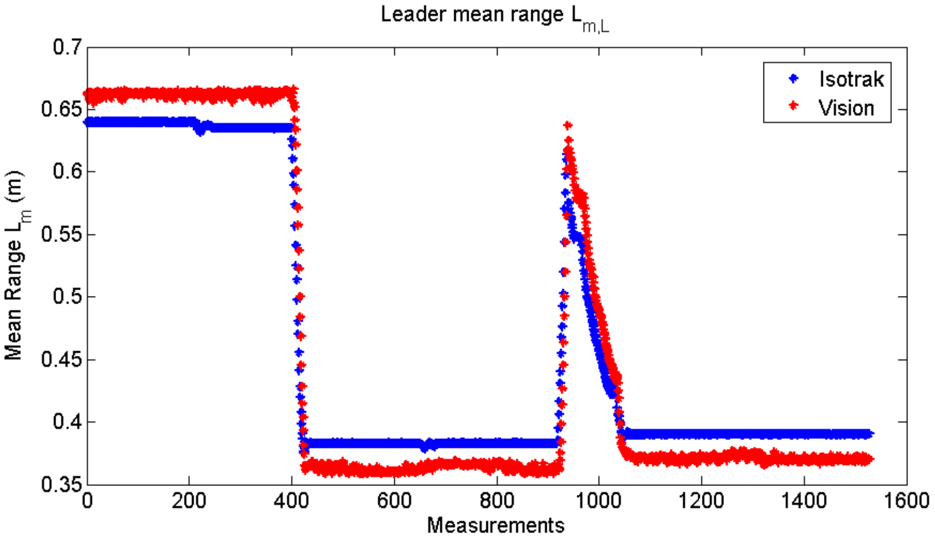


Figure 10.5: The mean range of the Leader Lm, L with respect to the surface.

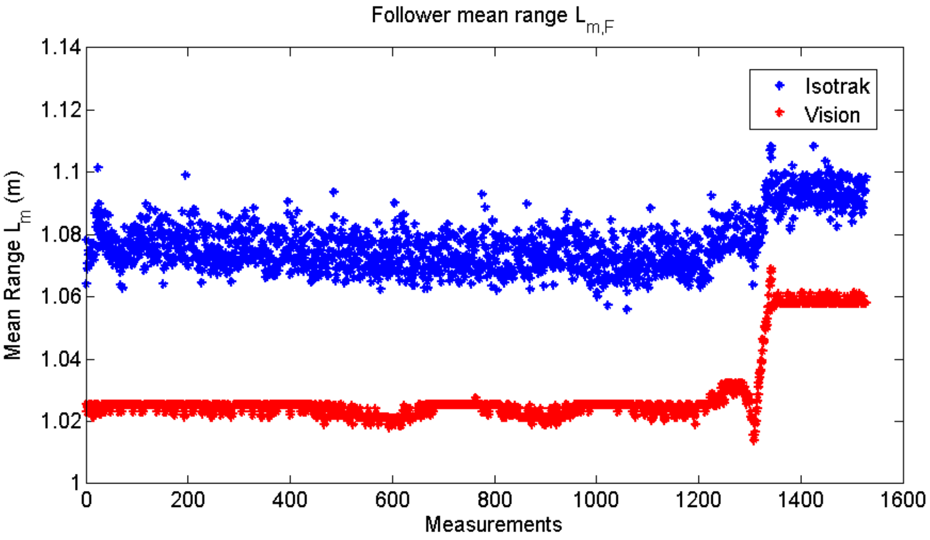


Figure 10.6: The mean range of the Follower Lm, F with respect to the surface.

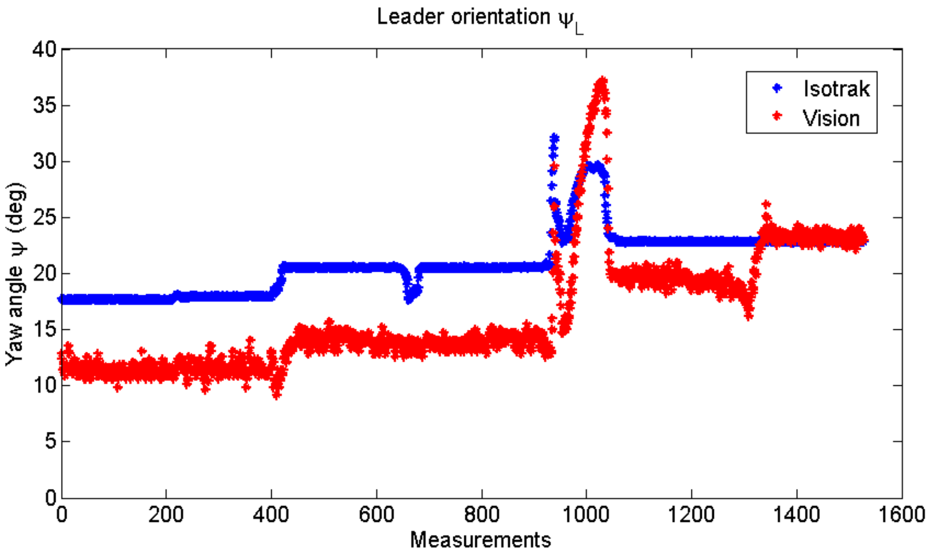


Figure 10.7: The orientation of the Leader ψ_L with respect to the surface.

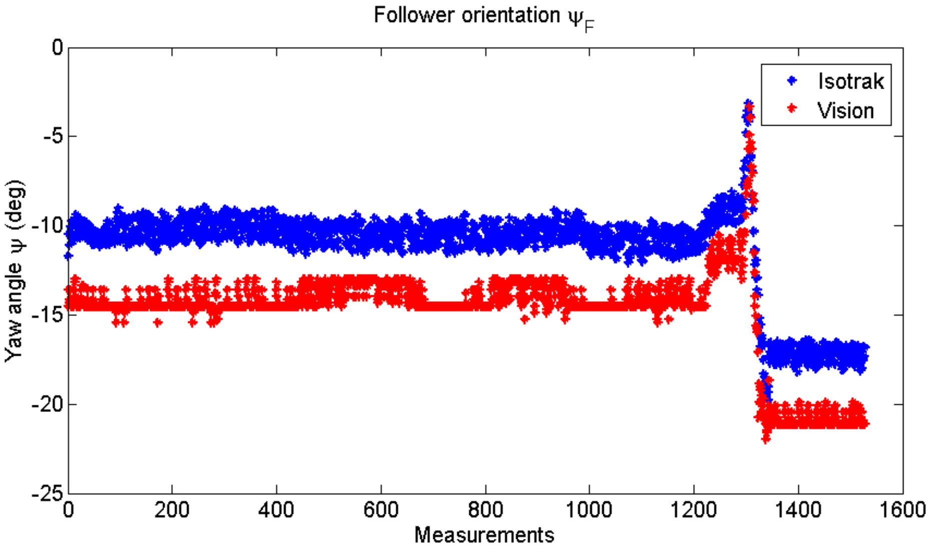


Figure 10.8: The orientation of the Follower ψ_F with respect to the surface.

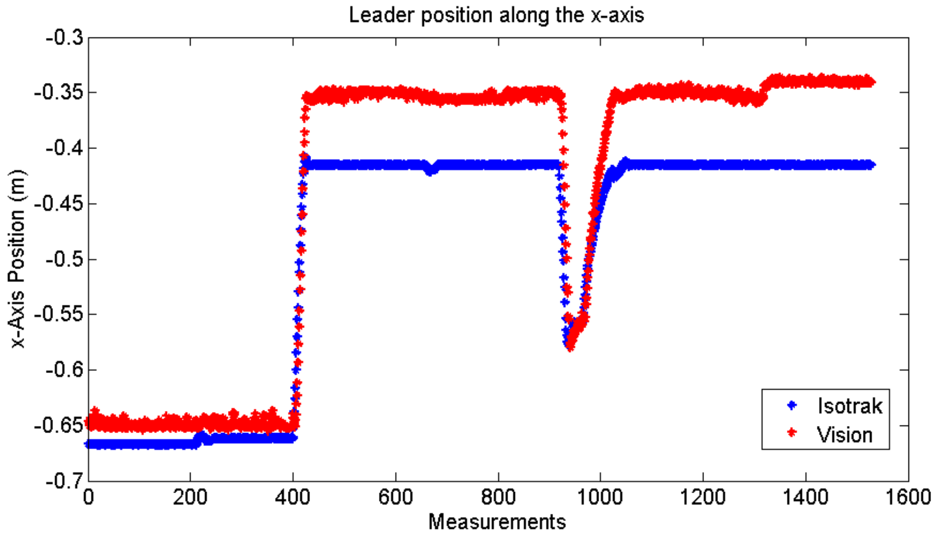


Figure 10.9: The position of the Leader along the x-axis of the WCS.

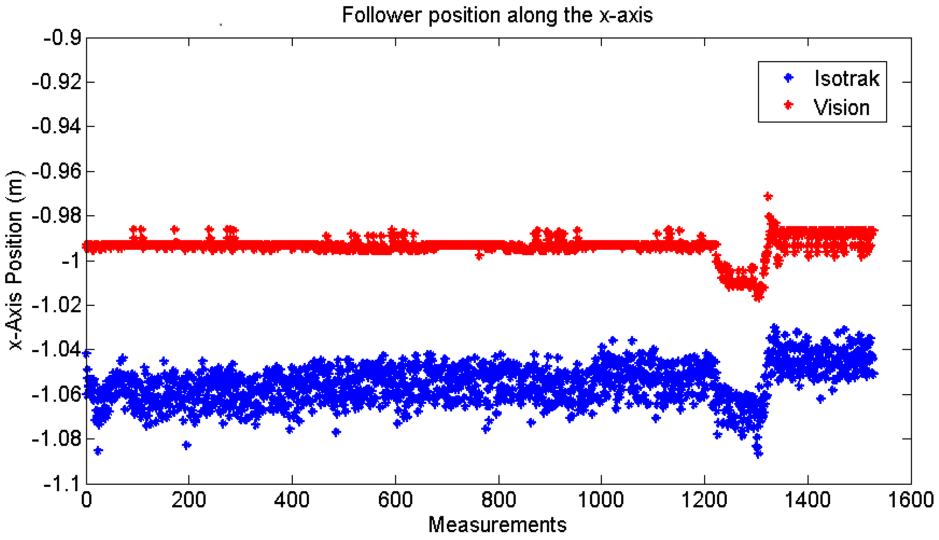


Figure 10.10: The position of the Follower along the x-axis of the WCS.

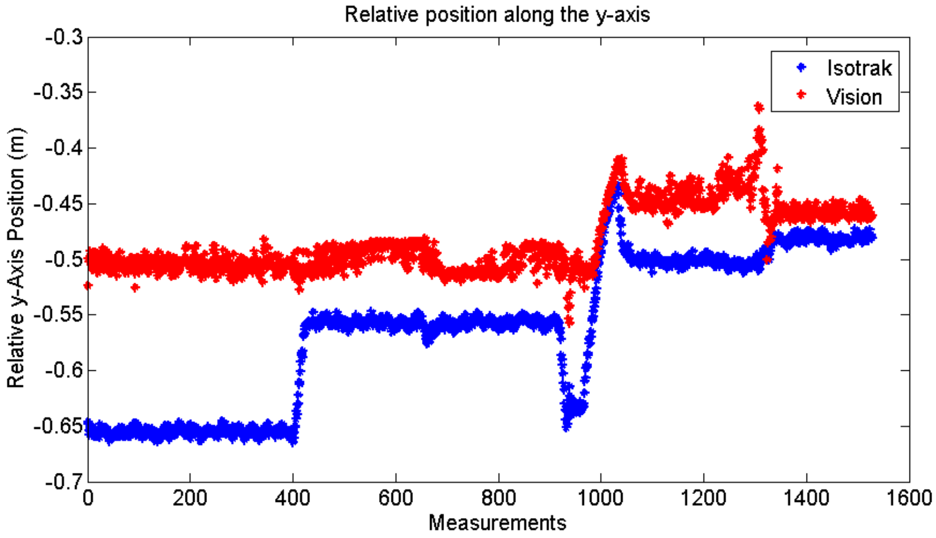


Figure 10.11: The relative position between the two vehicles along the y-axis.

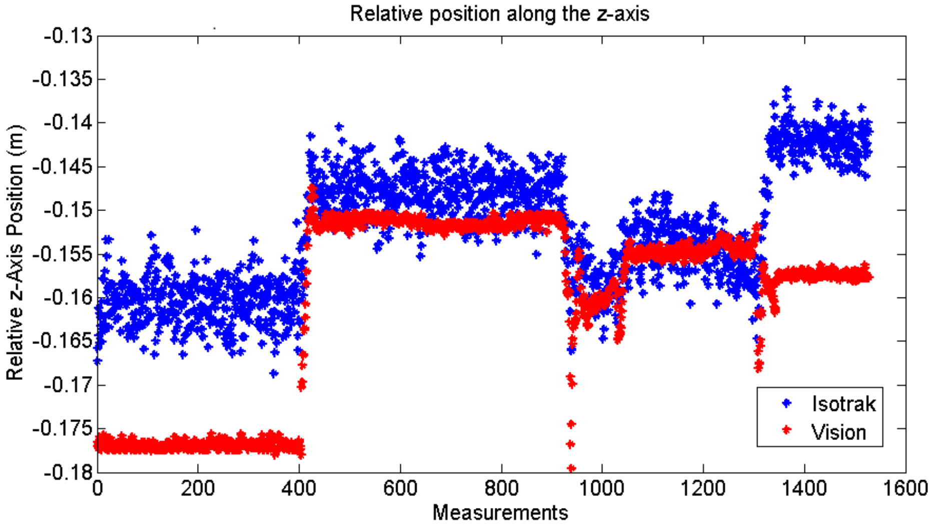


Figure 10.12: The relative position between the two vehicles along the z-axis.

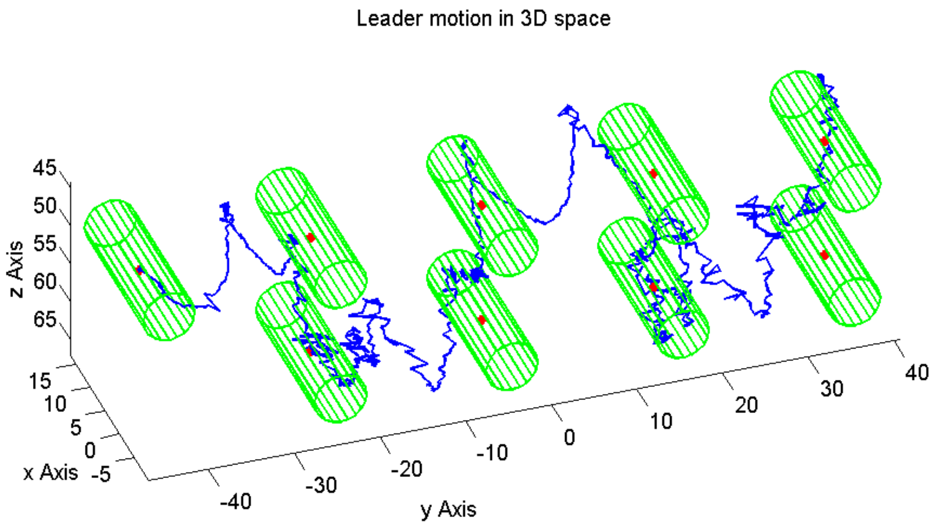


Figure 10.13: The motion of the Leader in 3D-space. Red dots denote the waypoints that the Leader must reach. The green cylinders around them denote the error at which the Leader must approach the waypoints.

significant based on the fact that, during the inspection task, the vehicles must only scan the surface. In addition, these deviations may originate from small errors during the positioning of the Isotrak sensors or from measurement errors of the Isotrak system itself.

10.2 Leader Motion Control Experiments

During the Leader controller testing, the Leader vehicle was placed at a random initial pose with respect to the surface. The position of the waypoints was calculated automatically by the algorithm based on this initial pose, in order to provide a meander-like trajectory with width of 0.2m and height of 0.15m on the yz-plane. Then, the vehicle began its motion using as input for the controller the position information received by the Isotrak system, which is described in Chapter 9. The vehicle had to reach 8 waypoints (Fig. 10.13) within a distance of 3cm. Each time the vehicle reached a waypoint, the algorithm fed to the controller the position of a new one. The experiment ended when the vehicle had reached all waypoints within the prescribed accuracy. The corresponding results are presented in Fig. 10.13 through 10.15.

As it can be seen from Fig. 10.14, the Leader's motion is not consistent on the xy-plane. This is primarily due to the influence of the tether cable during the experiments which hindered the Leader from following its planned trajectory. On the other hand, Fig. 10.15 shows that the Leader reaches the prescribed waypoints within the desired accuracy and that the vehicle traverses a well-shaped meander on the yz-plane. This was the goal of the experiment and, as far as the meander-like trajectory is concerned, it can be considered as successful.

The green cylinders in Fig. 10.13 denote the fact that, during the experimental procedure, the controller does not regulate the vertical distance (x distance) that the vehicle will have when reaching a waypoint. The vehicle must form a well-shaped meander trajectory and, thus, only the y and z position errors with respect to the waypoints are considered. This is expressed by cylinders whose bases have a radius of 3cm that represent the admissible error.

The above experiment was also performed for a rectangle trajectory. The vehicle began its motion from a random configuration, followed three waypoints as shown in Fig. 10.16 and returned to its initial position. The results of this experiment are shown in Fig. 10.16 through Fig. 10.18.

In Fig. 10.17, the motion of the vehicle is again influenced by the presence of the tether. In Fig.

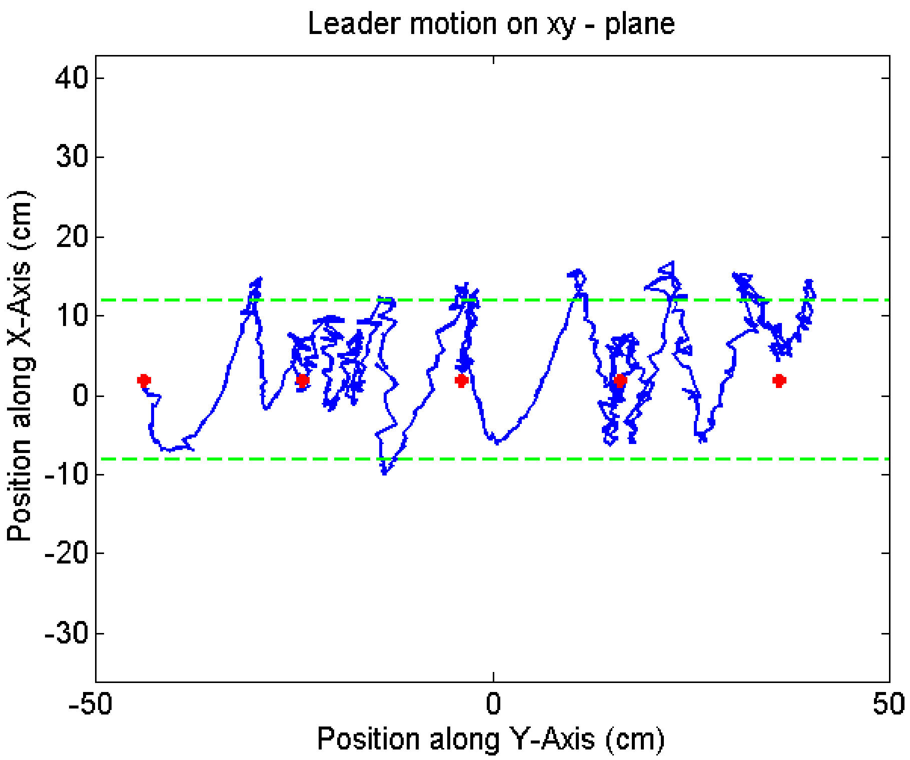


Figure 10.14: The motion of the Leader on the xy-plane. The green dashed lines denote lines A and B of the control procedure.

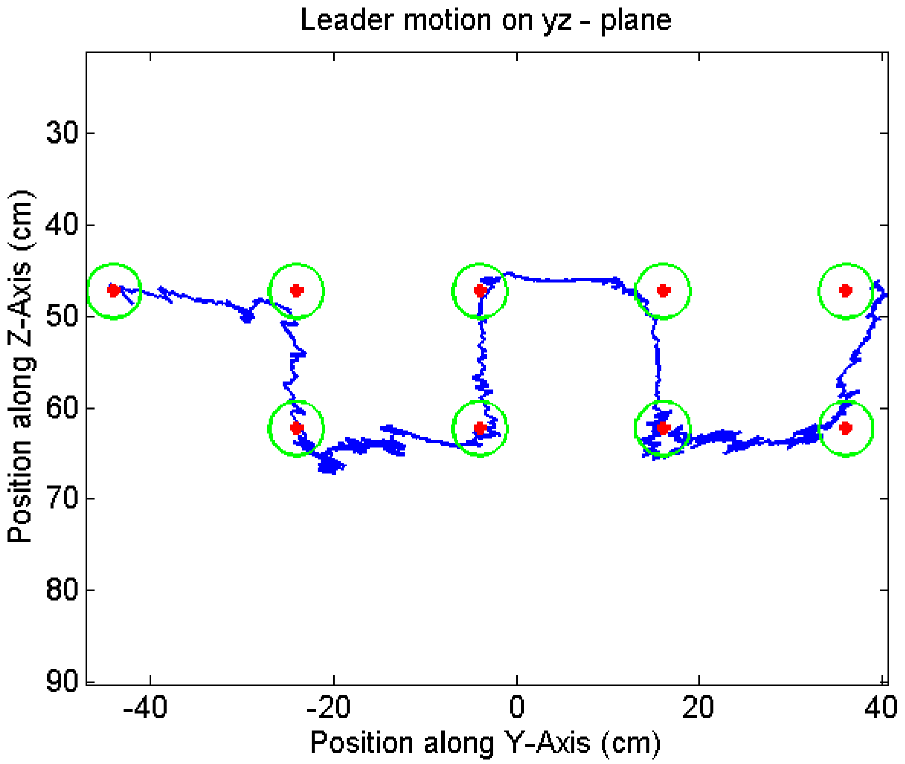


Figure 10.15: The motion of the Leader on the yz-plane. Red dots denote the waypoints that the Leader must reach. The circles around them denote the error at which the Leader must approach the waypoints.

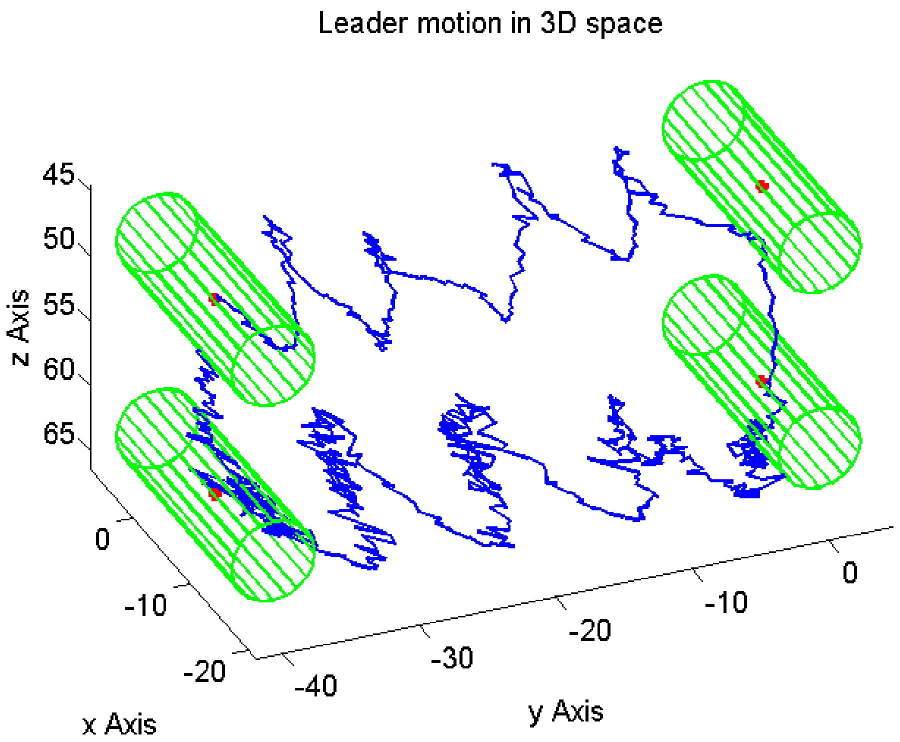


Figure 10.16: The motion of the Leader in 3D-space (rectangle trajectory). Red dots denote the waypoints that the Leader must reach. The green cylinders around them denote the error at which the Leader must approach the waypoints.

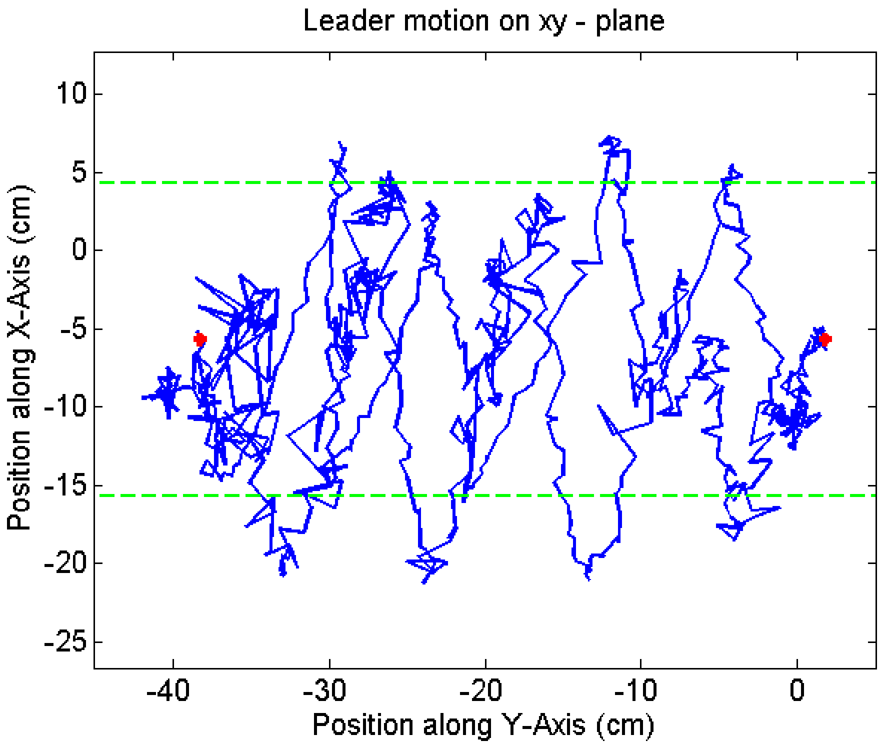


Figure 10.17: The motion of the Leader on the xy-plane (rectangle trajectory). The green dashed lines denote lines A and B of the control procedure.

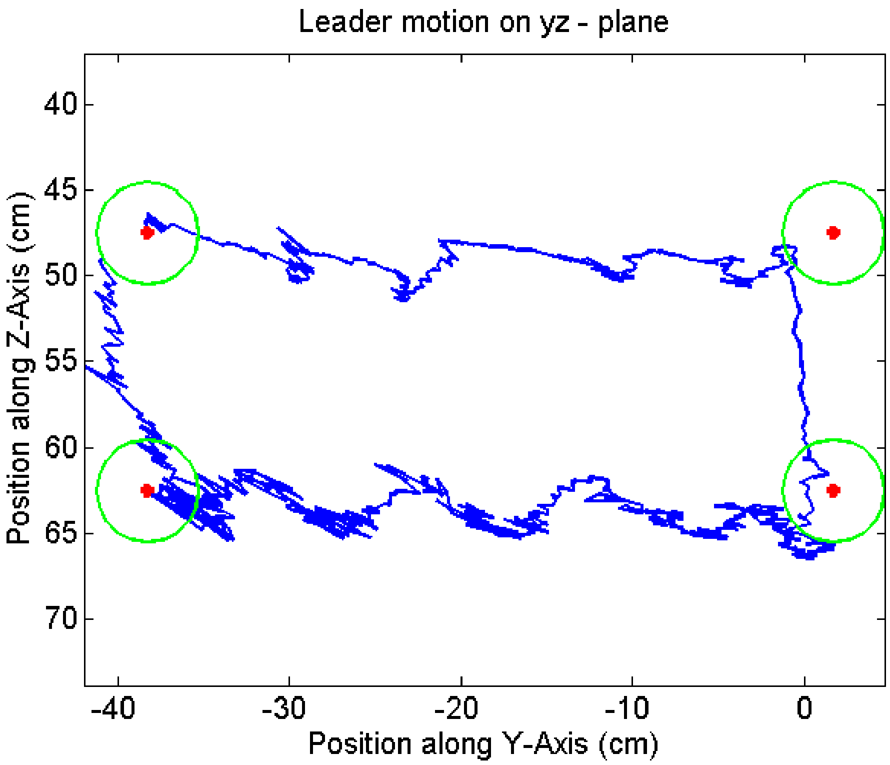


Figure 10.18: The motion of the Leader on the yz-plane (rectangle trajectory). Red dots denote the waypoints that the Leader must reach. The circles around them denote the error at which the Leader must approach the waypoints.

10.18 though, it is shown that the vehicle performs successfully the rectangle shaped trajectory. The purpose of this experiment was to examine the Leader motion controller for an alternative trajectory that contained all the primitive motions included in a meander-like one (i.e. left, right, upward and downward motions). At the same time, the Leader motion controller was tested with respect to both directions of saw-like motion, while the first experiment tested the controller only with respect to the right motion.

Chapter 11

Conclusions & Future Work

11.1 Discussion & Conclusions

In this work, we presented a cooperation scheme between two underwater vehicles. The vehicles had to inspect a flat surface without using any explicit communications. This was achieved by designing and implementing an implicit communications scheme. The communication scheme related to the exchange of localization information between the vehicles. First, a localization procedure based on a laser pointer configuration was designed and mathematically analyzed. Then, a suitable laser pointer setup was designed and constructed for both vehicles. An appropriate machine vision algorithm was also implemented in order to detect laser dots inside the image frame of a camera. The whole system was calibrated carefully in order to achieve maximum localization accuracy. The system was successfully tested through a number of experiments.

The next step was to design a motion control scheme for each of the vehicles in order to navigate them cooperatively in front of the surface. We chose to use a Leader - Follower approach due to the number of the vehicles and the fact that this approach provided simpler control design choices. The vehicles had different kinematic capabilities since one of them was subject to nonholonomic constraints with regard to its planar motion. This vehicle was chosen as the Leader, while the other acted as the Follower. It was more intuitive to make the less capable vehicle move on its own and the more capable one to follow it.

The motion of the vehicles had to comply to certain constraints that related either to the inspection task itself or to their cooperation. More specifically, these constraints related to the visibility of the Leader laser dots inside the Follower image frame and any conflict or collision that would occur between the vehicles. In addition, the vehicles had to face the surface at all times, which for the case of the nonholonomic vehicle posed a great difficulty. Two decentralized motion control schemes, one for each vehicle, were designed that addressed these problems. The control scheme for the Leader vehicle was tested and provided very good results.

Although a thorough analysis of the problem is provided in this work, there are still many issues to be examined and much more research to be done. These issues will be addressed in the next section.

11.2 Future Work

First, any future work must be related to the experimental testing of the Follower motion controller in order to assess its efficacy under real-life conditions. The same applies for the whole cooperation procedure where both motion controllers will be used. The experimental setup is prepared while the simulation provided very good results. The only thing that remains is the final experiment and the final assessment of the proposed algorithm.

Second, there are certain actions that could improve the results of the existing work. The first is the enhancement of the vision algorithm in order to render it more robust to a broader range of lightning conditions and to adress some of its problems described in Chapter 6, while at the same time keeping its efficiency, its robustness and its computational simplicity at the same levels. Another action would be to define mathematically some optimum or near optimum parameter values for the laser pointer setup in order to address more efficiently the visibility constraints with regard to the Leader laser dots. The goal would be to keep the corresponding angles and distances as small as possible, but acquire localization information with high discretization.

Second, there is some research to be done with respect to the kinematic capabilities of the Leader and the Follower vehicle and how these can influence the efficiency of the cooperation scheme. For example, the possibility of chosing the nonholonomic vehicle as the Follower and the holonomic one as the Leader has not been examined. The same applies for the case where both vehicles are nonholonomic. Although, there is some relevant research for the last case (e.g. in [14]), this issue has not been studied for an inspection task or for the case of using the implicit communications scheme proposed in this work.

Another interesting topic of research would be the case where both vehicles receive localization information with respect to the other. As it is already mentioned, in this work, only the case where the Follower vehicle localizes itself with respect to the Leader has been considered. Thus, if a laser setup similar to that of the Leader was equipped on the Follower, a new control problem could arise where both vehicles attempt to overcome their motion and visibility constraints with regard to the same or to a similar task. This approach could provide greater robustness to the system since both vehicles will try to maintain their formation and address the corresponding constraints, but it could also lead to limit cycles which is not rare in these occasions.

Finally, the constraints could be addressed more successfully if the motion controllers were dynamic and not kinematic. In that way, the dynamic characteristics of the vehicles would be also considered. Another approach would be to design the motion control scheme for one or both vehicles based on the Model Predictive Control (MPC) methodology. This would ensure that the motion constraints would be explicitly taken into account and, at the same time, that the control strategy would be optimized.

Appendix A

Notes on Kinematics & Dynamics of Marines Vehicles

This section will briefly present the basic static and dynamic modelling for marine vehicles based on the mathematical approach presented in [15]. *Statics* is related with the equilibrium of bodies when they are at rest or they move with constant velocity, while *dynamics* concern bodies that have accelerated motion. The following analysis concerns bodies that move in 6 Degrees-of-Freedom (DOFs), since 6 independent coordinates are necessary to describe the motion of a body in 3D-space. The first three coordinates (x,y,z) and their derivatives (u,v,w) refer to the position and translational motion along the x-, y- and z- axes. The last three coordinates (ϕ,θ,ψ) and their derivatives (p,q,r) refer to the orientation and rotational motion of the body about the same axes. For marine vehicles, the six velocities u, v, w, p, q, r are defined according to [39] as *surge*, *sway*, *heave*, *roll*, *pitch* and *yaw*, respectively, as shown in Fig. A.1.

A.1 Marine Vehicle Kinematics

In order to analyse the motion of marine vehicles in 3D-space, two coordinate frames are usually defined, as shown in Fig. A.2. The first is attached to the vehicle and it is called the *body-fixed* coordinate frame $X_0Y_0Z_0$. It is moving and its origin O is usually chosen to coincide with the center of gravity of the vehicle. The body axes X_0, Y_0 and Z_0 coincide with the principal axes of inertia and are defined as:

- X_0 - longitudinal axis (directed from aft to fore, i.e. from back to front)
- Y_0 - transverse or lateral axis (directed from port to starboard, i.e. from left to right)
- Z_0 - normal or vertical axis (directed from top to bottom)

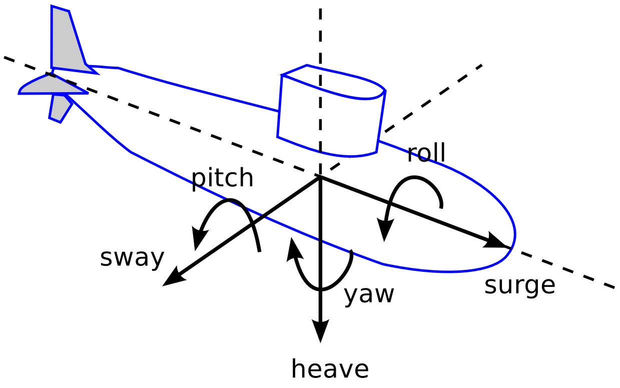


Figure A.1: The 6 components of motion as defined by [39].

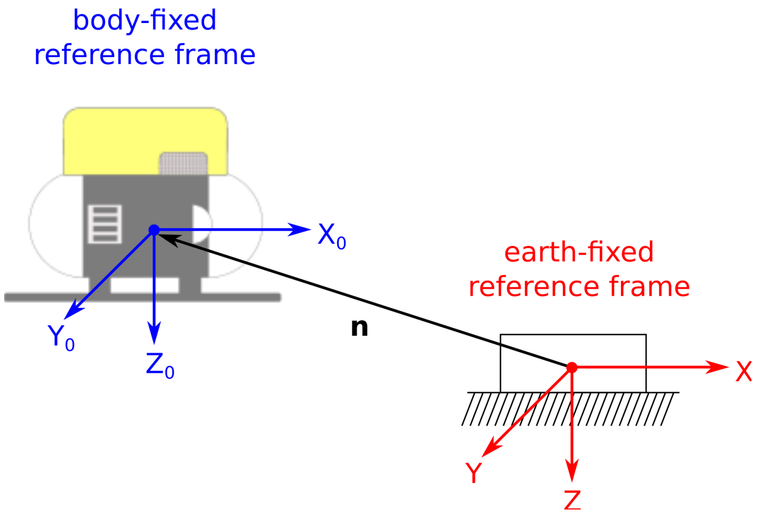


Figure A.2: The two coordinate systems used when examining the kinematics of marine vehicles.

The motion of the moving coordinate frame is studied with respect to an inertial frame. The earth-fixed frame XYZ can be considered as inertial, since it is fairly logical to assume that the motion of the Earth hardly affects low speed marine vehicles. Thus, the position and the orientation is described relative to the inertial frame XYZ, while the linear and angular velocities of the vehicle are expressed with respect to the body-fixed coordinate frame $X_0Y_0Z_0$.

The general motion of a marine vehicle can be described by the following vectors:

$$\begin{aligned}
&\boldsymbol{\eta} = [\boldsymbol{\eta_1}^T, \boldsymbol{\eta_2}^T]^T; \quad &&\boldsymbol{\eta_1} = [x, y, z]^T; \quad &&\boldsymbol{\eta_2} = [\phi, \theta, \psi]^T;\\
&\mathbf{v} = [\mathbf{v_1}^T, \mathbf{v_2}^T]^T; \quad &&\mathbf{v_1} = [u, v, w]^T; \quad &&\mathbf{v_2} = [p, q, r]^T;\\
&\boldsymbol{\tau} = [\boldsymbol{\tau_1}^T, \boldsymbol{\tau_2}^T]^T; \quad &&\boldsymbol{\tau_1} = [X, Y, Z]^T; \quad &&\boldsymbol{\tau_2} = [K, M, N]^T;
\end{aligned} \tag{A.1}$$

where $\boldsymbol{\eta}$ represents the position ($\boldsymbol{\eta_1}$) and orientation ($\boldsymbol{\eta_2}$) of the vehicle with coordinates in the inertial frame, $\mathbf{v}$ represents the linear ($\mathbf{v_1}$) and angular velocities ($\mathbf{v_2}$) of the vehicle expressed in the body-fixed frame and $\boldsymbol{\tau}$ denotes the forces ($\boldsymbol{\tau_1}$) and the moments ($\boldsymbol{\tau_2}$) acting on the vehicle in the body-fixed frame. The orientation of the vehicle $\boldsymbol{\eta_2}$ is represented by means of Euler angles.

A.1.1 Linear Velocities Transformation

The relation between the derivative of $\boldsymbol{\eta_1}$ and the velocity vector $\mathbf{v_1}$ is expressed as:

$$\dot{\boldsymbol{\eta}}_\mathbf{1} = \boldsymbol{J_1}(\boldsymbol{\eta_2})\ \mathbf{v_1} \Rightarrow \mathbf{v_1} = \boldsymbol{J_1}(\boldsymbol{\eta_2})^{-1}\ \dot{\boldsymbol{\eta}}_\mathbf{1} \tag{A.2}$$

where $\boldsymbol{J_1}(\boldsymbol{\eta_2})$ is a transformation matrix which depends on the Euler angles ϕ (roll), θ (pitch) and ψ (yaw). The matrix $\boldsymbol{J_1}(\boldsymbol{\eta_2})$ is derived as a result of three successive rotations of the frame $X_3Y_3Z_3$ which lead to the body-fixed frame $X_0Y_0Z_0$. The frame $X_3Y_3Z_3$ is parallel to the inertial frame XYZ but its origin coincides with the origin of the body-fixed frame. The corresponding rotations are:

- *yaw* rotation of $X_3Y_3Z_3$ about axis Z_3 by an angle ψ in order to obtain the frame $X_2Y_2Z_2$
- *pitch* rotation of $X_2Y_2Z_2$ about axis Y_2 by an angle θ in order to obtain the frame $X_1Y_1Z_1$
- *roll* rotation of $X_1Y_1Z_1$ about axis X_1 by an angle ϕ in order to obtain the frame $X_0Y_0Z_0$

These rotations can be expressed by the simple rotation matrices:

$$R_{z,\psi} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_{y,\theta} \;=\; \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \quad R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \tag{A.3}$$

where $c\,\cdot$ denotes the cosine of an angle and $s\,\cdot$ denotes its sine. The matrices $R_{z,\psi}$, $R_{y,\theta}$, $R_{x,\phi}$ correspond to the yaw, pitch and roll rotations, respectively. The transformation matrix $\boldsymbol{J_1}(\boldsymbol{\eta_2})$ can then be written as:

$$\boldsymbol{J_1}(\boldsymbol{\eta_2}) = R_{z,\psi}\; R_{y,\theta}\; R_{x,\phi} \tag{A.4}$$

which equals to:

$$\boldsymbol{J_1}(\boldsymbol{\eta_2}) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{A.5}$$

A.1.2 Angular Velocities Transformation

The relation between the derivative of $\boldsymbol{\eta_2}$ and the velocity vector $\mathbf{v_2}$ is written as:

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{2}} = \boldsymbol{J_2}(\boldsymbol{\eta_2})\; \mathbf{v_2} \tag{A.6}$$

The transformation matrix $\boldsymbol{J_2}(\boldsymbol{\eta_2})$ can be calculated based on the following expression:

$$\mathbf{v_2} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R^T_{x,\phi} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R^T_{x,\phi} R^T_{y,\theta} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = \boldsymbol{J_2}(\boldsymbol{\eta_2})^{-1} \dot{\boldsymbol{\eta}}_{\boldsymbol{2}} \tag{A.7}$$

Expanding the previous equation yields:

$$\boldsymbol{J_2}(\boldsymbol{\eta_2})^{-1} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix} \Rightarrow \boldsymbol{J_2}(\boldsymbol{\eta_2}) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & {}^{s\phi}\!/_{c\theta} & {}^{c\phi}\!/_{c\theta} \end{bmatrix} \tag{A.8}$$

where $s\cdot = \sin(\cdot)$, $c\cdot = \cos(\cdot)$ and $t\cdot = \tan(\cdot)$. Finally, the kinematic equations of a marine vehicle can be expressed concisely as follows:

$$\begin{bmatrix} \boldsymbol{\eta_1} \\ \boldsymbol{\eta_2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J_1}(\boldsymbol{\eta_2}) & \mathbf{0_{3\times3}} \\ \mathbf{0_{3\times3}} & \boldsymbol{J_2}(\boldsymbol{\eta_2}) \end{bmatrix} \begin{bmatrix} \mathbf{v_1} \\ \mathbf{v_2} \end{bmatrix} \Leftrightarrow \dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\; \mathbf{v} \tag{A.9}$$

A.2 Marine Vehicle Dynamics

The 6 DOF nonlinear dynamic equations of motion of a marine vehicle can be expressed in matrix form as:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{A.10}$$

where

- M is the *inertia matrix* (including the added mass term)
- $\mathbf{C}(\mathbf{v})$ is the *matrix of Coriolis and centripetal terms* (including the added mass terms)
- $\mathbf{D}(\mathbf{v})$ is the *dumping matrix*
- $\boldsymbol{g}(\boldsymbol{\eta})$ is the *vector of gravitational forces and moments*

- $\boldsymbol{\tau}$ is the *vector of control inputs*

The subsequent analysis aims only to explain and analyse the terms of the previous equation but it will not study their derivation and their mathematical expression in detail. The only purpose of this analysis is to briefly present and describe the terms that are related to the dynamics of the vehicle.

A.2.1 Rigid-Body Dynamics

The *Equations of Motion* for a 6-DOF rigid body can be written based on the Newtonian analysis in the following form:

$$\begin{aligned}
X &= m\left[\dot{u} - vr + wq - x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})\right] \\
Y &= m\left[\dot{v} - wp + ur - y_G(r^2 + p^2) + z_G(qr - \dot{p}) + x_G(qp + \dot{r})\right] \\
Z &= m\left[\dot{w} - uq + vp - z_G(p^2 + q^2) + x_G(rp - \dot{q}) + y_G(rq + \dot{p})\right] \\
K &= I_x\dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} \\
&\quad + m\left[y_G(\dot{w} - uq + vp) - z_G(\dot{v} - wp + ur)\right] \\
M &= I_y\dot{q} + (I_x - I_z)rp - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} \\
&\quad + m\left[z_G(\dot{u} - vr + wq) - x_G(\dot{w} - uq + vp)\right] \\
N &= I_z\dot{r} + (I_y - I_x)pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{zx} \\
&\quad + m\left[x_G(\dot{v} - wp + ur) - y_G(\dot{u} - vr + wq)\right]
\end{aligned} \tag{A.11}$$

where $\mathbf{r_G} = [x_G, y_G, z_G]^T$ denotes the coordinates of the vehicle's centre of gravity with respect to the body-fixed coordinate system and the terms I_x, I_y, I_z, $I_{xy} = I_{yx}$, $I_{xz} = I_{zx}$, $I_{yz} = I_{zy}$ are components of the body's inertia tensor I_0 which refers to the body-fixed coordinate frame $X_0Y_0Z_0$:

$$\mathbf{I_0} \triangleq \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & I_{zy} & I_z \end{bmatrix} \tag{A.12}$$

The above equations can be expressed in a more convenient form as:

$$\mathbf{M_{RB}}\dot{\mathbf{v}} + \mathbf{C_{RB}}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau_{RB}} \tag{A.13}$$

where $\mathbf{v} = [u, v, w, p, q, r]^T$ is the linear and angular velocity vector, $\boldsymbol{\tau_{RB}} = [X, Y, Z, K, M, N]^T$ is a generalized vector of forces and moments, $\mathbf{M_{RB}}$ is the rigid-body inertia matrix and $\mathbf{C_{RB}}(\mathbf{v})$ is the Coriolis and centripetal matrix.

The rigid-body equations of motion can be significantly simplified if the origin of the body-fixed frame is chosen to coincide with the centre of gravity, which means that $\mathbf{r_G} = [0, 0, 0]^T$, and the body axes coincide with the principal axes of inertia. The latter implies that the inertia tensor is diagonal ($\mathbf{I_0} =$ diag$\{I_x, I_y, I_z\}$).

A.2.2 Hydrodynamic Forces and Moments

The hydrodynamic forces and moments is assumed to be a linear superposition of two categories of forces, namely *radiation induced forces* and *Froude-Kriloff and diffraction forces* (for more information refer to [15]). The radiation induced forces can be considered as the sum of three components, namely:

- *Added mass* due to the inertia of the surrounding fluid

- *Radiation-induced potential dumping* due to energy carried away by generated surface waves
- *Restoring forces* due to Archimedes' principle (i.e. weight and buoyancy)

which can be mathematically expressed as:

$$\boldsymbol{\tau_R} = -\underbrace{\mathbf{M_A}\dot{\mathbf{v}} - \mathbf{C_A}(\mathbf{v})\mathbf{v}}_{\text{added mass}} - \underbrace{\mathbf{D_P}(\mathbf{v})\mathbf{v}}_{\text{potential damping}} - \underbrace{\boldsymbol{g(\eta)}}_{\text{restoring forces}} \tag{A.14}$$

Other dumping effects include skin friction, wave drift damping and damping due to vortex shedding which leads to:

$$\boldsymbol{\tau_D} = -\underbrace{\mathbf{D_S}(\mathbf{v})\mathbf{v}}_{\text{skin friction}} - \underbrace{\mathbf{D_W}(\mathbf{v})\mathbf{v}}_{\text{wave drift damping}} - \underbrace{\mathbf{D_M}(\mathbf{v})\mathbf{v}}_{\text{damping due to vortex shedding}} \tag{A.15}$$

Thus, the hydrodynamic forces and moments $\boldsymbol{\tau_H}$ can be written as the sum of $\boldsymbol{\tau_R}$ and $\boldsymbol{\tau_D}$ which is:

$$\boldsymbol{\tau_H} = -\mathbf{M_A}\dot{\mathbf{v}} - \mathbf{C_A}(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} - \boldsymbol{g(\eta)} \tag{A.16}$$

where the total hydrodynamic damping matrix is defined as:

$$\mathbf{D}(\mathbf{v}) \triangleq \mathbf{D_P}(\mathbf{v})\mathbf{v} + \mathbf{D_S}(\mathbf{v})\mathbf{v} + \mathbf{D_W}(\mathbf{v})\mathbf{v} + \mathbf{D_M}(\mathbf{v})\mathbf{v} \tag{A.17}$$

Ignoring the Froude-Kriloff and diffraction terms, the previous analysis leads to:

$$\mathbf{M_{RB}}\dot{\mathbf{v}} + \mathbf{C_{RB}}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau_{RB}} \tag{A.18}$$

$$\boldsymbol{\tau_{RB}} = \boldsymbol{\tau_H} + \boldsymbol{\tau_E} + \boldsymbol{\tau} \tag{A.19}$$

where $\boldsymbol{\tau_E}$ denotes the *environmental* forces and moments that act on the vehicle and $\boldsymbol{\tau}$ denotes the *propulsion* forces and moments. Thus, it stands that:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} + \boldsymbol{g(\eta)} = \boldsymbol{\tau_E} + \boldsymbol{\tau} \tag{A.20}$$

where

$$\mathbf{M} \triangleq \mathbf{M_{RB}} + \mathbf{M_A};\;\; \mathbf{C}(\mathbf{v})\mathbf{v} \triangleq \mathbf{C_{RB}}(\mathbf{v})\mathbf{v} + \mathbf{C_A}(\mathbf{v})\mathbf{v} \tag{A.21}$$

A.2.3 Added Mass and Inertia

The added mass matrix $\mathbf{M_A}$ is written as:

$$\mathbf{M_A} \triangleq -\begin{bmatrix} X_{\dot u} & X_{\dot v} & X_{\dot w} & X_{\dot p} & X_{\dot q} & X_{\dot r} \\ Y_{\dot u} & Y_{\dot v} & Y_{\dot w} & Y_{\dot p} & Y_{\dot q} & Y_{\dot r} \\ Z_{\dot u} & Z_{\dot v} & Z_{\dot w} & Z_{\dot p} & Z_{\dot q} & Z_{\dot r} \\ K_{\dot u} & K_{\dot v} & K_{\dot w} & K_{\dot p} & K_{\dot q} & K_{\dot r} \\ M_{\dot u} & M_{\dot v} & M_{\dot w} & M_{\dot p} & M_{\dot q} & M_{\dot r} \\ N_{\dot u} & N_{\dot v} & N_{\dot w} & N_{\dot p} & N_{\dot q} & N_{\dot r} \end{bmatrix} \tag{A.22}$$

where, e.g., $X_{\dot u} = \frac{\partial Y}{\partial \dot u}$, while the hydrodynamic Coriolis and centripetal matrix $\mathbf{C_A}$ is written as:

$$\mathbf{C_A} \triangleq \begin{bmatrix} 0 & 0 & 0 & 0 & -a_3 & a_2 \\ 0 & 0 & 0 & a_3 & 0 & -a_1 \\ 0 & 0 & 0 & -a_2 & a_1 & 0 \\ 0 & -a_3 & a_2 & 0 & -b_3 & b_2 \\ a_3 & 0 & -a_1 & b_3 & 0 & -b_1 \\ -a_2 & a_1 & 0 & -b_2 & b_1 & 0 \end{bmatrix} \tag{A.23}$$

where

$$\begin{aligned}
a_1 &= X_{\dot{u}}u + X_{\dot{v}}v + X_{\dot{w}}w + X_{\dot{p}}p + X_{\dot{q}}q + X_{\dot{r}}r \\
a_2 &= X_{\dot{v}}u + Y_{\dot{v}}v + Y_{\dot{w}}w + Y_{\dot{p}}p + Y_{\dot{q}}q + Y_{\dot{r}}r \\
a_3 &= X_{\dot{w}}u + Y_{\dot{w}}v + Z_{\dot{w}}w + Z_{\dot{p}}p + Z_{\dot{q}}q + Z_{\dot{r}}r \\
b_1 &= X_{\dot{p}}u + Y_{\dot{p}}v + Z_{\dot{p}}w + K_{\dot{p}}p + K_{\dot{q}}q + K_{\dot{r}}r \\
b_2 &= X_{\dot{q}}u + Y_{\dot{q}}v + Z_{\dot{q}}w + K_{\dot{q}}p + M_{\dot{q}}q + M_{\dot{r}}r \\
b_3 &= X_{\dot{r}}u + Y_{\dot{r}}v + Z_{\dot{r}}w + K_{\dot{r}}p + M_{\dot{r}}q + N_{\dot{r}}r
\end{aligned} \tag{A.24}$$

In the case of fully-submerged underwater vehicles that move at low speed and have three planes of symmetry, the off-diagonal terms of the added mass matrix $\mathbf{M_A}$ are neglected and, hence, the matrices $\mathbf{M_A}$ and $\mathbf{C_A}$ can be written as:

$$\mathbf{M_A} = -diag\{X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}\} \tag{A.25}$$

$$\mathbf{C_A} = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix} \tag{A.26}$$

A.2.4 Hydrodynamic Damping

As it was previously described, the hydrodynamic damping on marine vehicles may include terms that relate to:

- $\mathbf{D_P}(\mathbf{v})$ which concerns radiation-induced potential damping due to forced body oscillations
- $\mathbf{D_S}(\mathbf{v})$ which concerns linear skin friction due to laminar boundary layers and quadratic skin friction due to turbulent boundary layers
- $\mathbf{D_W}(\mathbf{v})$ which concerns wave drift damping
- $\mathbf{D_M}(\mathbf{v})$ which concerns damping due to vortex shedding (Morrison's equation)

Thus, as it is already mentioned, the total hydrodynamic damping matrix can be written as:

$$\mathbf{D}(\mathbf{v}) = \mathbf{D_P}(\mathbf{v})\mathbf{v} + \mathbf{D_S}(\mathbf{v})\mathbf{v} + \mathbf{D_W}(\mathbf{v})\mathbf{v} + \mathbf{D_M}(\mathbf{v})\mathbf{v} \tag{A.27}$$

In general, hydrodynamic damping for marine vehicles that travel at high speed is highly non-linear and coupled. The hydrodynamic damping matrix, though, can be significantly simplified under certain assumptions. According to these assumptions, the vehicle performs a non-coupled motion, it has three planes of symmetry and terms higher than second order are negligible. This leads to a diagonal form of $\mathbf{D}(\mathbf{v})$ where there are only linear and quadratic terms on the diagonal as shown below:

$$\mathbf{D}(\mathbf{v}) = -diag\{X_u, Y_v, Z_w, K_p, M_q, N_r\} - diag\{X_{u|u|}|u|, Y_{v|v|}|v|, Z_{w|w|}|w|, K_{p|p|}|p|, M_{q|q|}|q|, N_{r|r|}|r|\} \tag{A.28}$$

A.2.5 Restoring Forces

The restoring forces refer to the gravitational and buoyant forces acting on the vehicle. These forces can be mathematically expressed as:

- $W = mg$ (weight) which acts on the centre of gravity $\mathbf{r_G} = [x_G, y_G, z_G]^T$ of the vehicle
- $B = \rho g V$ (buoyancy) which acts on the centre of buoyancy $\mathbf{r_B} = [x_B, y_B, z_B]^T$ of the vehicle

where m is the mass of the vehicle, V is the volume of fluid displaced by the vehicle, g is the gravitational acceleration and ρ is the fluid density. It must be noted that the gravitational acceleration is considered positive towards the downward direction. Next, the weight and buoyancy forces are transformed to the body-fixed coordinate system using the transformation matrix $\boldsymbol{J_1}(\boldsymbol{\eta_2})$ as follows:

$$\boldsymbol{f_G}(\boldsymbol{\eta_2}) = \boldsymbol{J_1^{-1}}(\boldsymbol{\eta_2}) \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} \; ; \quad \boldsymbol{f_B}(\boldsymbol{\eta_2}) = -\boldsymbol{J_1^{-1}}(\boldsymbol{\eta_2}) \begin{bmatrix} 0 \\ 0 \\ B \end{bmatrix} \tag{A.29}$$

and by applying Newton's second law it yields that:

$$\boldsymbol{g}(\boldsymbol{\eta}) = \begin{bmatrix} (W-B)s\theta \\ -(W-B)c\theta s\phi \\ -(W-B)c\theta c\phi \\ -(y_G W - y_B B)c\theta c\phi + (z_G W - z_B B)c\theta s\phi \\ (z_G W - z_B B)s\theta + (x_G W - x_B B)c\theta c\phi \\ -(x_G W - x_B B)c\theta s\phi - (y_G W - y_B B)s\theta \end{bmatrix} \tag{A.30}$$

A.3 Equations of Motion

In this section, the equations of motion for a marine vehicle will be expressed and presented with regard to both the body-fixed and the inertial coordinate system.

A.3.1 Body-Fixed Vector Representation

As it is clear from the previous analysis, the kinematic and dynamic equations of motion for a marine vehicle can be expressed with regard to the body-fixed coordinate frame as follows:

$$\boxed{\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\,\mathbf{v}} \tag{A.31}$$

$$\boxed{\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau_E} + \boldsymbol{\tau}} \tag{A.32}$$

where

$$\mathbf{M} = \mathbf{M_{RB}} + \mathbf{M_A} \tag{A.33}$$

$$\mathbf{C}(\mathbf{v}) = \mathbf{C_{RB}}(\mathbf{v}) + \mathbf{C_A}(\mathbf{v}) \tag{A.34}$$

$$\mathbf{D}(\mathbf{v}) = \mathbf{D_P}(\mathbf{v}) + \mathbf{D_S}(\mathbf{v}) + \mathbf{D_W}(\mathbf{v}) + \mathbf{D_M}(\mathbf{v}) \tag{A.35}$$

A.3.2 Earth-Fixed Body Representation

The earth-fixed representation is obtained by applying the following kinematic trasformations (assuming $\boldsymbol{J}(\boldsymbol{\eta})$ is not singular):

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{v} && \Leftrightarrow \mathrm{v} = \boldsymbol{J}^{-1}(\boldsymbol{\eta})\dot{\boldsymbol{\eta}} \\ \ddot{\boldsymbol{\eta}} &= \boldsymbol{J}(\boldsymbol{\eta})\dot{\mathrm{v}} + \dot{\boldsymbol{J}}(\boldsymbol{\eta})\mathrm{v} && \Leftrightarrow \dot{\mathrm{v}} = \boldsymbol{J}^{-1}(\boldsymbol{\eta})\left[\ddot{\boldsymbol{\eta}} - \boldsymbol{J}^{-1}(\boldsymbol{\eta})\dot{\boldsymbol{\eta}}\right] \end{aligned} \tag{A.36}$$

Defining:

$$\begin{aligned} \boldsymbol{M}_{\eta}(\boldsymbol{\eta}) &= \boldsymbol{J}^{-T}(\boldsymbol{\eta})\boldsymbol{M}\boldsymbol{J}^{-1}(\boldsymbol{\eta}) \\ \boldsymbol{C}_{\eta}(\mathrm{v},\boldsymbol{\eta}) &= \boldsymbol{J}^{-T}(\boldsymbol{\eta})\left[\mathrm{C}(\mathrm{v}) - \boldsymbol{M}\boldsymbol{J}^{-1}(\boldsymbol{\eta})\dot{\boldsymbol{J}}(\boldsymbol{\eta})\right]\boldsymbol{J}^{-1}(\boldsymbol{\eta}) \\ \boldsymbol{D}_{\eta}(\mathrm{v},\boldsymbol{\eta}) &= \boldsymbol{J}^{-T}(\boldsymbol{\eta})\mathrm{D}(\mathrm{v})\boldsymbol{J}^{-1}(\boldsymbol{\eta}) \\ \boldsymbol{g}_{\eta}(\boldsymbol{\eta}) &= \boldsymbol{J}^{-T}(\boldsymbol{\eta})\boldsymbol{g}(\boldsymbol{\eta}) \\ \boldsymbol{\tau}_{\eta}(\boldsymbol{\eta}) &= \boldsymbol{J}^{-T}(\boldsymbol{\eta})(\boldsymbol{\tau} + \boldsymbol{\tau}_{\boldsymbol{E}}) \end{aligned} \tag{A.37}$$

results in the following expression for the earth-fixed representation:

$$\boxed{\boldsymbol{M}_{\eta}(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \boldsymbol{C}_{\eta}(\mathrm{v},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \boldsymbol{D}_{\eta}(\mathrm{v},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \boldsymbol{g}_{\eta}(\boldsymbol{\eta}) = \boldsymbol{\tau}_{\eta}(\boldsymbol{\eta})} \tag{A.38}$$

Appendix B

Notes on Visual Servoing, Computer Vision and Image Processing

B.1 Visual Servoing

Over the last decades, *visual servoing* has been examined in numerous works and used as a motion control strategy for many applications. In visual servoing, the task consists on using visual information provided by machine vision algorithms to control the *pose* (position and orientation) of a manipulator's end-effector or the pose of the robot itself, if autonomous vehicles (mobile, marine or aerial) are considered. Visual servoing includes the results from many different disciplines such as image processing, kinematics, dynamics, control theory and real-time computing. In the next sections, *Position-Based* and *Image-Based Visual Servoing* are presented briefly. The sole purpose of this presentation is to introduce the reader to the main idea of each methodology and discuss some of their main advantages and disadvantages.

B.1.1 Position-Based Visual Servoing (PBVS)

In PBVS (Fig. B.1), certain features of the scene are extracted from the image using a computer vision algorithm. Then, the pose of the target with respect to the camera (or vice versa) is estimated based on the geometric model of the target or the geometric relations between its features and on the camera model. The corresponding control law aims to minimize the error between the estimated pose and the desired one.

One of the key advantages of the PBVS is that it separates the control problem from the feature extraction and the estimation problems. Thus, it is possible to construct a control law without taking

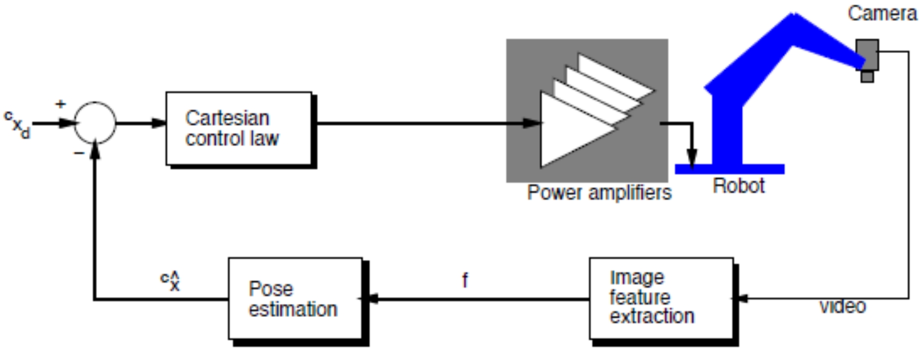


Figure B.1: Position-Based Visual Servoing (PBVS) closed-loop diagram. *Taken from [20].*

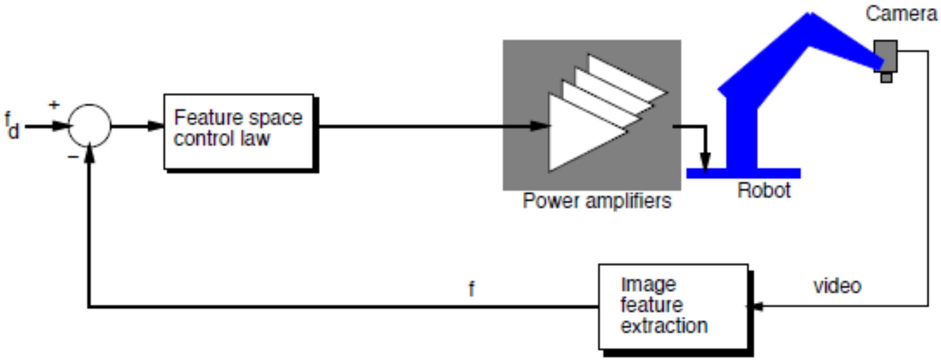


Figure B.2: Image-Based Visual Servoing (PBVS) closed-loop diagram. *Taken from [20].*

into consideration the exact process of feature extraction or pose estimation. This provides a more clear approach for the design of the visual servoing scheme. Another advantage of the PBVS is that it is possible to describe tasks in Cartesian space which is usually the task space of a robotic platform.

One primary disadvantage of the method is the fact that the feedback signal is based on estimated quantities which depend heavily on the calibration parameters of the system. Thus, pose-based methods can be extremely sensitive to calibration errors. In addition, these methods require an accurate model of the target object which may not be always available.

B.1.2 Image-Based Visual Servoing (IBVS)

In IBVS (Fig. B.2), the error signal, and thus the control inputs, is calculated directly with respect to image features. The error signal can be defined in two ways. One approach is to use the camera model in order to calculate the desired image feature parameters at the desired pose. There is also the ``teach-by-showing" approach, in which the robot is placed at a desired configuration and the corresponding image is used to compute the desired image feature parameter vector.

Although the error signal is defined in the image feature parameter space $\mathcal{F}$, the control inputs are usually defined either in joint space for the case of a manipulator or in cartesian (task) space $\mathcal{T}$. Thus, it is necessary to relate changes in the image feature parameters to changes in the robot position. In other words, there must be a way to relate the image space to the task space. This is done by using the image Jacobian (also known as *feature sensitivity matrix* or *interaction matrix*) which relates the rate of change of the end-effector position to the rates of change of the image feature parameters as shown below:

$$\dot{\mathbf{f}} = \mathbf{J_v}(\mathbf{r})\dot{\mathbf{r}} \tag{B.1}$$

where $\mathbf{f}$ denotes the image feature parameter vector and $\mathbf{r}$ denotes the state of the end-effector in some parameterization of the task space. In the case of a camera mounted on the end-effector of a manipulator, the vector $\mathbf{r}$ expresses the velocity screw of the camera with regard to its body-fixed frame. In the case of a camera equipped on an autonomous vehicle, the vector $\mathbf{r}$ can simply denote the linear and angular velocities of the vehicle expressed in the camera frame. The image Jacobian can be mathematically written as:

$$\mathbf{J_v}(\mathbf{r}) = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{r}}\right] = \begin{bmatrix} \frac{\partial f_1(\mathbf{r})}{\partial r_1} & \cdots & \frac{\partial f_1(\mathbf{r})}{\partial r_m} \\ \vdots & & \vdots \\ \frac{\partial f_k(\mathbf{r})}{\partial r_1} & \cdots & \frac{\partial f_k(\mathbf{r})}{\partial r_m} \end{bmatrix} \tag{B.2}$$

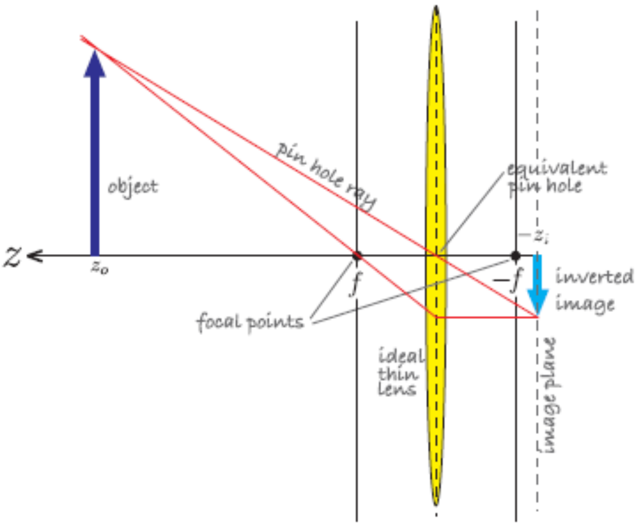


Figure B.3: An object in the scene is tranformed due to the camera lens to a 2D projection on the camera sensor. *Taken from [9]*

where m is the dimension of the task space $\mathcal{T}$, k is the dimension of the image feature parameter space $\mathcal{F}$ and therefore $\mathbf{J}_\mathrm{v} \in \mathbb{R}^{k \times m}$.

One key advantage of the image-based methods is that the efficiency of the system is completely independent from errors due to sensor modelling or camera calibration. This means that, even if the system is miscalibrated, the kinematic error will tend to zero when the image error tends to zero under the premise that the feedback system is asymptotically stable. This is not the case for position-based methods. Another advantage relates to the fact that image-based methods reduce the computational delay, while they obviate the need for image interpretation.

On the other hand, image-based techniques may pose certain difficulties during the control design, since the system becomes non-linear and highly coupled. In addition, there may be singularities in the feature mapping function which correspond to instabilities in the control law.

B.2 Camera Model

This section presents the camera modelling that is used in most visual servoing applications. Every camera contains a lens that forms a 2D projection of the scene on the image plane where the sensor is located (Fig. B.3). During the imaging process, this projection causes direct depth information to be lost. Thus, each 2D point on the image plane corresponds to a ray in the 3D space. In order to calculate the 3D coordinates of an image space point, some additional information is required. This information can originate from multiple cameras or multiple views from the same camera or geometric relations between feature points in the scene. In the subsequent analysis, three camera models will be presented; the *perspective projection* model, the *affine projection* model and the model that is used in this work.

In the following, in order to preserve a consistent notation, the x-axis of the camera coordinate system will correspond to the horizontal axis of the image plane, the z-axis to its vertical axis and the y-axis will be perpendicular to the other two and will coincide with the optical axis of the camera, as shown in Fig. B.4. The origin of the camera coordinate system will be located at distance f behind the image plane, where f is the focal length.

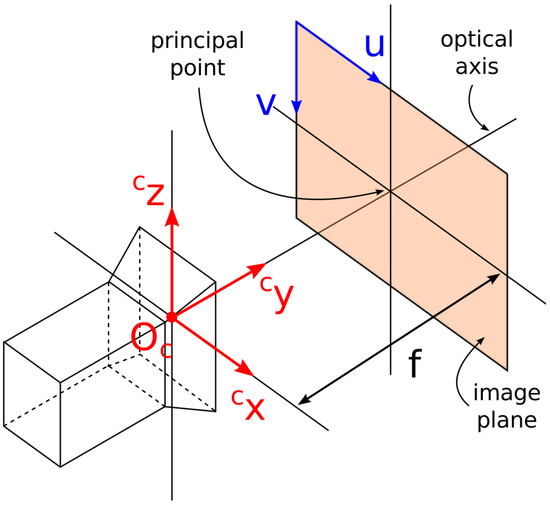


Figure B.4: Camera modelling representation.

B.2.1 Perspective Projection Model

Let ${}^{\mathbf{c}}\mathbf{P} = [x, y, z]^T$ denote the coordinates of a point in 3D space with respect to the camera. If the imaging process is modeled by perspective projection, the point will project on the image plane with coordinates $\mathbf{p} = [u, v]^T$ as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{y} \begin{bmatrix} x \\ z \end{bmatrix} \tag{B.3}$$

If the coordinates of the point $\mathbf{P}$ are expressed to another system, then a coordinate transformation must be performed in order to be expressed in the coordinate system of the camera.

B.2.2 Affine Projection Model

Assuming that the imaging process is now modeled by an affine projection, which is a linear approximation to the perspective projection, the image plane coordinates of a point in 3D space are given by:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{A}{}^{\mathbf{c}}\mathbf{P} + \mathbf{c} \tag{B.4}$$

where $\mathbf{A}$ is an arbitrary 2×3 matrix and $\mathbf{c}$ is an arbitrary 2×1 vector.

B.2.3 Linear Perspective Projection Model

According to the camera model that is used in this work, the projection of a 3D point onto the image plane is calculated as:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_x & s & u_0 & 0 \\ 0 & a_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ z \\ y \\ 1 \end{bmatrix} \tag{B.5}$$

where

$$a_x = fk_x$$
$$a_y = fk_y$$

while $[u_0, v_0]^T$ are the coordinates of the image center inside the image frame, s is the skew factor, f is the camera's focal length and k_x, k_y are the scaling factors along the u and v image axes respectively. The parameters a_x, a_y, u_0, v_0 are called the *intrinsic parameters* of the camera.

In this work, the skew factor is assumed to be zero ($s = 0$) and, thus, the above equation can be written in the form:

$$\begin{aligned} u = a_x \frac{x}{y} + u_0 \Rightarrow u - u_0 = a_x \frac{x}{y} \\ v = a_y \frac{z}{y} + v_0 \Rightarrow v - v_0 = a_y \frac{z}{y} \end{aligned} \tag{B.6}$$

which if written as:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_x & 0 & 0 & 0 \\ 0 & a_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ z \\ y \\ 1 \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \tag{B.7}$$

it resembles the affine projection model, while if written in the form:

$$\begin{bmatrix} u - u_0 \\ v - v_0 \end{bmatrix} = \frac{f}{y} \begin{bmatrix} k_x \, x \\ k_y \, z \end{bmatrix} \tag{B.8}$$

it resembles the perspective projection model. In essence, it is the perspective projection model expressed in a form that includes the intrinsic parameters of the camera.

B.3 Camera Calibration Procedure

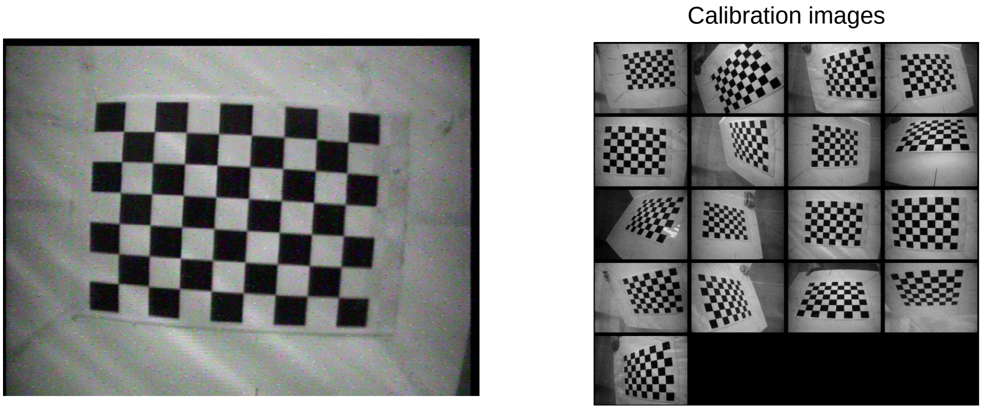
The intrinsic parameters of a camera (a_x, a_y, u_0, v_0) are usually not known beforehand. In order to define these parameters, a camera calibration procedure is necessary. In this work, two methodologies for the calibration of the camera were used (see Table B.1 for results). The first approach, which is the most common one, employed a checkerboard pattern, also referred to as *Tsai grid*. The second approach was task specific and it was based on the laser setup of the Follower vehicle. The steps of each approach are presented in more detail in the following sections. Although they both provided very good results, only the results from the second approach were used since they corresponded more to the task at hand. The camera calibration procedure was performed to the camera equipped on the Follower vehicle, since this was the vehicle that performed the relative localization procedure.

	Tsai Grid	Laser-based
$\mathbf{a_x}$	943.39141	967.9143
$\mathbf{a_y}$	950.20958	1144.9760
$\mathbf{u_0}$	319.5	300.9616
$\mathbf{v_0}$	239.5	278.1550

Table B.1: The results produced by the two calibration methods.

B.3.1 Camera Calibration with Tsai Grid

In this procedure, a checkerboard pattern was used in order to calibrate the camera. The steps are presented below and correspond to the steps followed using the Camera Calibration Toolbox in Matlab which can be found and downloaded from [7].



(a) Checkerboard pattern.

(b) Multiple images of the pattern taken at close proximity to the camera.

Figure B.5: Data collection using the checkerboard pattern (Tsai grid).

Step 1: Take images

The pattern (Fig. B.5a) was printed on a piece of paper. The paper was waterproofed and placed on a wide and hard surface in order to prevent bending of the pattern. The vehicle was placed on a stable base inside a small water tank. Then, the pattern was placed inside the field of fiew of the vehicle's camera at different angles and distances and a batch of 15 to 20 pictures was taken (Fig. B.5b). During this step, the pattern was placed close to the camera so that it could occupy as much space inside the image frame as possible. This was crucial for the accurate derivation of the intrinsic parameters later on.

Step 2: Corner Detection

The images were inserted in the toolbox. In the first image, the edge corners of the pattern were defined, thus, defining also the coordinate system. The tile size and the number of tiles for each direction of the pattern were also defined. Then the algorithm detected all the edges in the pattern (Fig. B.6).

Step 3: Pre-process all images

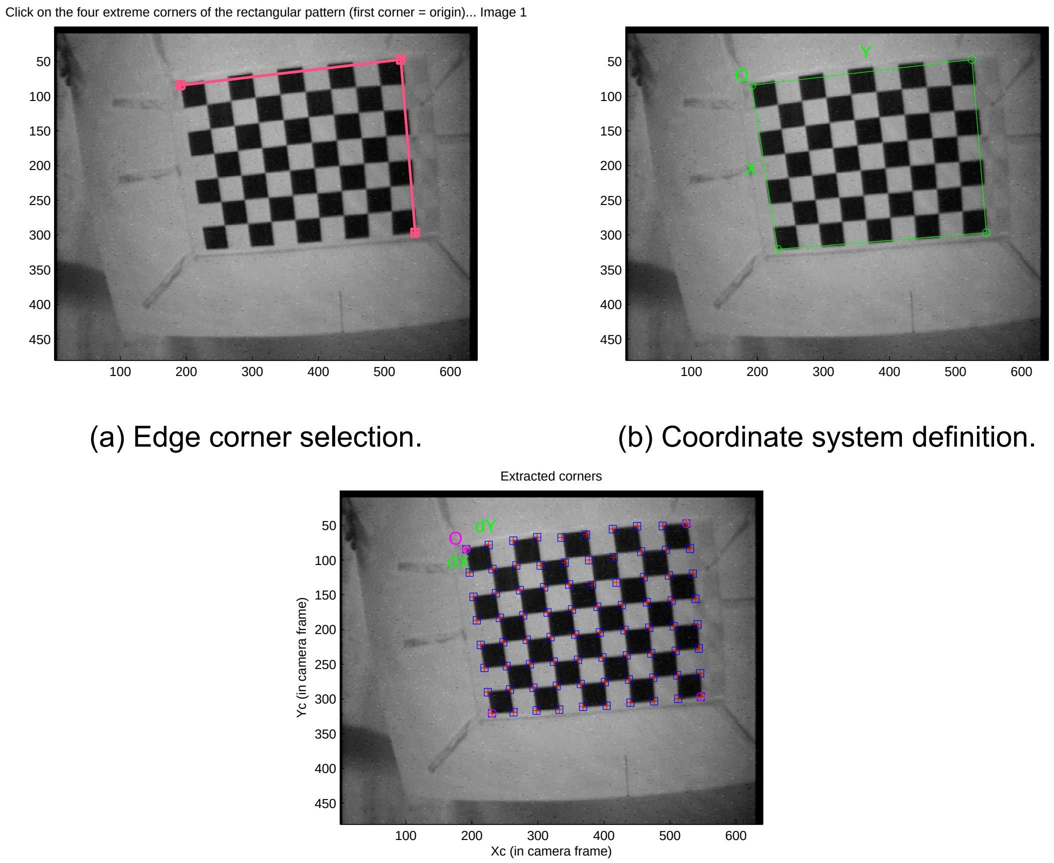
The previous step was repeated for each image. For the subsequent images, the tile size and tile number did not have to be defined unless the edge corners were poorly selected.

Step 4: Calibration Procedure

After the corner detection step was finished for all images, the actual calibration procedure was performed. The toolbox used the results from the previous steps in order to estimate the correct values of the intrinsic parameters of the camera, while it optimized the result.

Step 5: Error check and further calibration

After the calibration step, the user of the toolbox can see if there are any calibration errors, as shown in Fig. B.7a. The user can fix these errors by repeating one or more of the above steps using different parameter values if needed. Thus, the final step was to examine the final results and to apply any necessary corrections to the calibration results in order to ensure the accuracy of the procedure (the user can also see the extrinsic parameters of the camera as in Fig. B.7b, i.e. where the pattern lies with respect to the camera and vice versa). The necessary corrections can refer to the number of distortion coefficients to be computed or even the computation of the


Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1

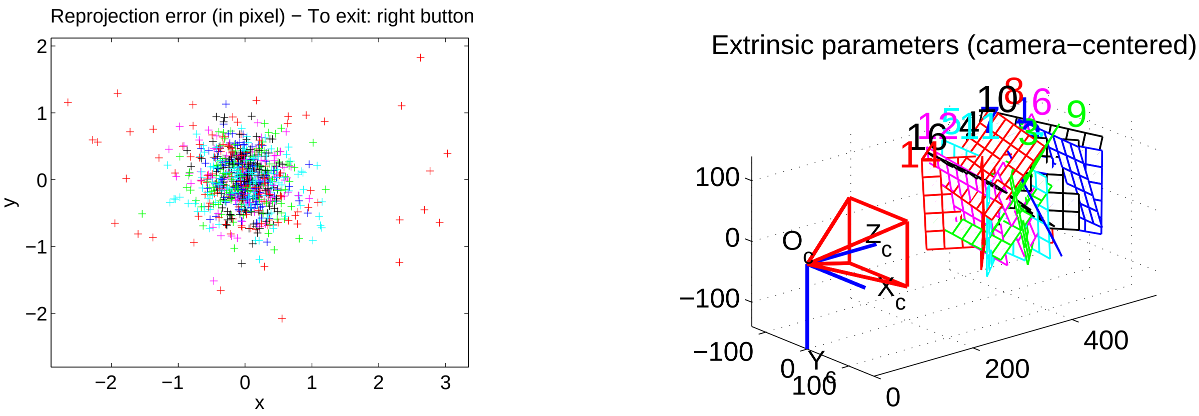
Extracted corners

(a) Edge corner selection.

(b) Coordinate system definition.

(c) Corner detection.

Figure B.6: Corner detection procedure.



(a) Calibration error graph.

(b) Extrinsic parameters graph.

Figure B.7: Error check procedure.

principal point (more detail on the functionality of the toolbox can be found in [7]).

It must be noted that the above procedure can be also executed using the corresponding C/C++ equivalent which uses OpenCV libraries and functions. The various parameters of the calibration can be set using an XML file. The exact procedure can be found in [1] as well as in other places.

B.3.2 Camera Calibration based on the Laser Setup Configuration

This procedure was based on the specific laser configuration that was equipped on the Follower vehicle. The main idea was to compute the intrinsic parameters of the vehicle's camera based on the position of the vehicle with respect to the surface and the position of the laser dots produced by its laser pointers inside the image frame. The necessary steps are described below in more detail.

Step 1: Take images and pose measurements

The vehicle was placed inside a water tank on a stable base, as in the previous procedure. A computer vision algorithm detected the laser dots of the vehicle inside its image frame and saved their image space coordinates (i.e. their position in the image frame) in a file. At the same time, the exact pose of the vehicle (position and yaw orientation) with respect to the surface was measured by the Isotrak II position tracker (see also Chapter 9). In this particular method, the base ensured that the roll and pitch angles of the vehicle were as close to zero as possible, which was crucial for the correct interpretation of the measurements taken.

Step 2: 3D coordinates of laser dots w.r.t. vehicle

In this step, the pose measurements of the Isotrak sensor were first filtered using a Butterworth filter. Then, they were transformed into 3D coordinates of the laser dots with respect to the vehicle. The computation was based on the laser configuration of the Follower. The coordinates were expressed with respect to the coordinate system of the camera. As explained in Chap. 9, the beams ejected from the laser pointers have a small deviation in comparison to their desired trajectory. Therefore, during the computation of the laser dot 3D coordinates, this deviation was also taken into account.

Step 3: Camera model expression

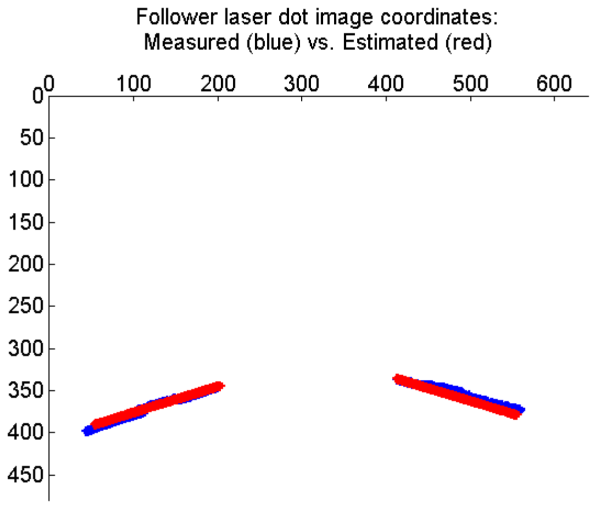


Figure B.8: The image space coordinates of the laser dots based on the estimated intrinsic parameters in comparison to the initial measurements.

In this step, the camera model was brought to a form where the parameter vector $\mathbf{x_p} = [a_x, u_0, a_y, v_0]^T$ expressed the solution of a linear equation as shown below:

$$\mathbf{C}\,\mathbf{x_p} = \mathbf{b} \tag{B.9}$$

The above equation can be more analytically written in the form:

$$\begin{bmatrix} \hat{\mathbf{x}}_\mathbf{L} & \mathbf{1_{n\times1}} & \mathbf{0_{n\times1}} & \mathbf{0_{n\times1}} \\ \mathbf{0_{n\times1}} & \mathbf{0_{n\times1}} & -\hat{\mathbf{z}}_\mathbf{L} & \mathbf{1_{n\times1}} \\ \hat{\mathbf{x}}_\mathbf{R} & \mathbf{1_{n\times1}} & \mathbf{0_{n\times1}} & \mathbf{0_{n\times1}} \\ \mathbf{0_{n\times1}} & \mathbf{0_{n\times1}} & -\hat{\mathbf{z}}_\mathbf{R} & \mathbf{1_{n\times1}} \end{bmatrix} \begin{bmatrix} a_x \\ u_0 \\ a_y \\ v_0 \end{bmatrix} = \begin{bmatrix} \mathbf{u_{L\,n\times1}} \\ \mathbf{v_{L\,n\times1}} \\ \mathbf{u_{R\,n\times1}} \\ \mathbf{v_{R\,n\times1}} \end{bmatrix} \tag{B.10}$$

where n is the number of measurements, the subscripts R and L refer to the right and the left laser pointer respectively and the elements of the vector $\hat{\mathbf{x}}_\mathbf{L}$ are of the form $\hat{x}_{L,i} = {}^{x_{L,i}}/_{y_{L,i}}(i = 1 \ldots n)$ while the elements of the vector $\hat{\mathbf{z}}_\mathbf{L}$ are of the form $\hat{z}_{L,i} = {}^{z_{L,i}}/_{y_{L,i}}(i = 1 \ldots n)$. The same applies for the vectors $\hat{\mathbf{x}}_\mathbf{R}$ and $\hat{\mathbf{z}}_\mathbf{R}$. The vectors $\mathbf{u_L}$, $\mathbf{v_L}$, $\mathbf{u_R}$, $\mathbf{v_R}$ represent the image space coordinates of the Follower laser dots inside its image frame for all measurements taken.

Step 4: Intrinsic Parameters Estimation and Verification

In this final step, the equation produced in the previous step is solved using the least-squares approximation method. This was done using the function `lsqlin` of the Matlab platform. Finally, based on the estimated intrinsic parameters, the image space coordinates of the laser dots were recalculated and compared to the initial measurements for validation purposes, as shown in Fig. B.8. The mean range and yaw orientation of the vehicle with respect to the surface were also calculated using the estimated parameters and compared to the corresponding values provided by the Isotrak sensor, as shown in Fig. B.9 and B.10.

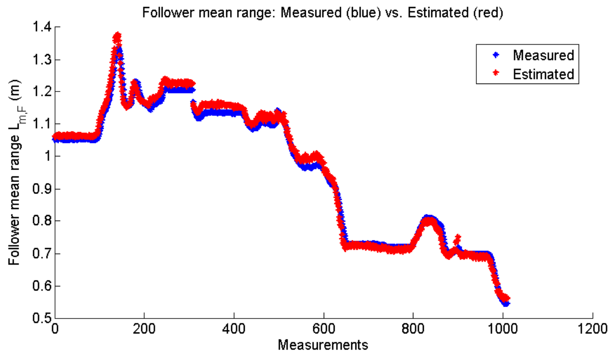


Figure B.9: Mean range of the Follower vehicle based on the estimated intrinsic parameters in comparison to Isotrak measurements.

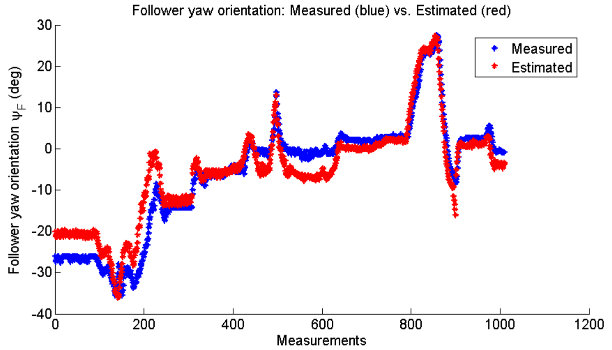


Figure B.10: Yaw orientation of the Follower vehicle based on the estimated intrinsic parameters in comparison to Isotrak measurements.

B.4 Image Processing Basic Concepts

B.4.1 Colour Models & Colour Spaces

A *Colour Model* is an abstract mathematical model that describes the representation of colours as tuples of *colour parameters* or *colour components*. This tuples usually contain three to four elements depending on the specific model used. When there is a mapping function that corresponds a tuple with certain values to a specific colour in the gamut of visible colours (see Fig. B.11), then a *Colour Space* is defined.

In computer vision, a *Colour Space* can also be defined as a three-dimensional space where each point corresponds to a specific colour (see also [9]). The colour space includes all colours

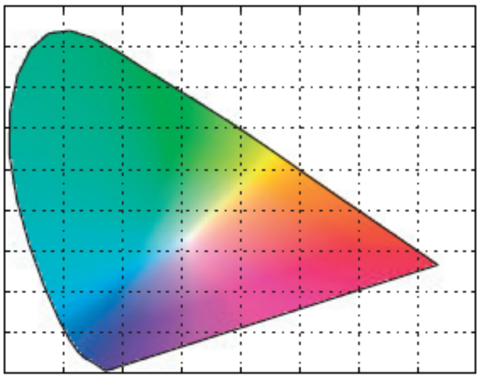


Figure B.11: Horseshoe-shaped gamut of visible colours. *Taken from [9]*

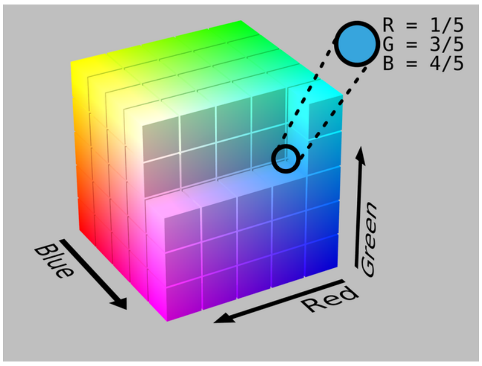


Figure B.12: The RGB cube. *Taken from Wikipedia*

and all levels of brightness. In this sense, there are many colour spaces and each of them describes colour using a unique set of parameters. Thus, the choice of a colour space depends on the particular computer vision application. The most commonly used colour spaces are RGB, HSV, HSL, YUV, YCbCr and CIE. In this section only the RGB and the HSV colour spaces will be presented.

RGB Colour Space

The *RGB colour space* (Fig. B.12) is an additive colour space where each colour is described by the chromaticities of the red, green and blue primaries (**R**ed - **G**reen - **B**lue). This means that each colour is a combination of these primaries and it depends on the amount of each primary in it. In computer vision, the amount of a primary in a colour is usually described by an integer between 0 and 255 (unsigned integer) where 0 corresponds to black (i.e. no colour) and 255 corresponds to white (i.e. full colour). Thus, for example, pure red is defined as (255,0,0), pure green as (0,255,0), pure blue as (0,0,255), yellow as (255,255,0), black as (0,0,0) and white as (255,255,255).

HSV Colour Space

The *HSV colour space* is a cylindrical-coordinate representation of points in the RGB colour space. HSV stands for **H**ue - **S**aturation - **V**alue. If the HSV space can be described by a cylinder (Fig. B.13), the angle around the central vertical axis corresponds to Hue, the distance from the axis corresponds to Saturation, and the distance along the axis corresponds to Value. In general, hue describes the colour itself, saturation describes the intensity of the colour (e.g. pink is red with low saturation) and value describes the brightness. On the HSV cylinder, the hue is described in degrees so that red corresponds to $0°$, green to $120°$ and blue to $240°$, while the cylinder wraps back to red at $360°$. With regard to saturation, pure colours are arranged on the edge of the cylinder and have a saturation value of 1, while at the centre the value is 0.

The main difference between the two colour spaces is that in RGB each component includes information about both colour and brightness, while in HSV brightness is described by a separate component (Value) and colour is described by the Hue and Saturation components. On the other hand, the RGB colour model is closer to the way the human eye interprets the environmental stimuli into visible colour.

The HSV representation is more suitable for applications where colour detection is needed because the colour can be described separately from the brightness and, thus, the corresponding algorithms are simple and straightforward. In this work though, the RGB representation was used instead, because the colours of the laser dots matched to RGB components. Thus, the corresponding algorithm was much simpler.

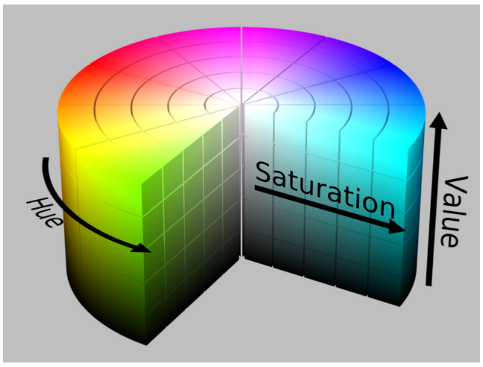


Figure B.13: The HSV cylinder. *Taken from Wikipedia*

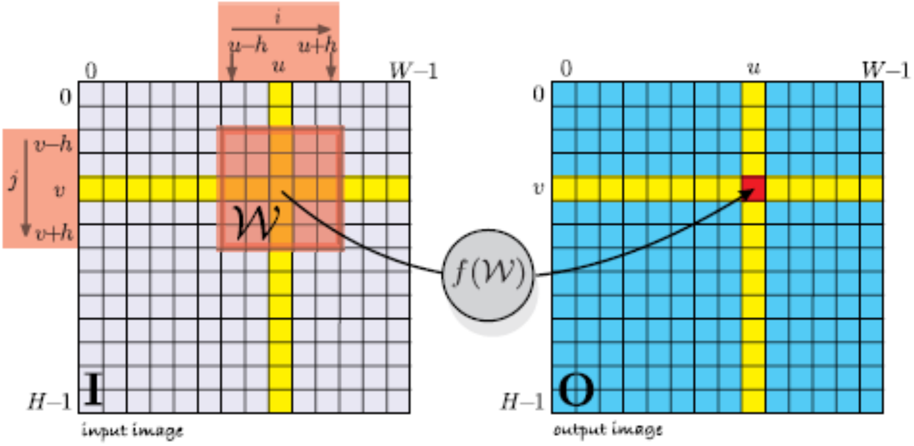


Figure B.14: Spatial operations. *Taken from [9]*

B.4.2 Filtering

The *filtering* operation is a spatial operation that aims to remove noise from an image. In a spatial operation, each pixel in the output image is a function of all pixels in a region surrounding the corresponding pixel in the input image

$$\mathbf{O}[u,v] = f(\mathbf{I}[u+i, v+j]),\ \forall (i,j) \in \mathcal{W},\ \forall (u,v) \in \mathbf{I} \tag{B.11}$$

where $\mathcal{W}$ is known as the window, typically a $w \times w$ square region with odd side length $w = 2h+1$ where $h \in \mathbb{Z}^+$ is the half-width ([9]). Filtering may involve linear spatial operations, in which case it is called *linear spatial filtering* or *spatial convolution*, or it may involve non-linear operations, in which case it is called *non-linear spatial filtering*([17]). Linear filtering involves a convolution

$$\mathbf{O}[u,v] = \sum_{(i,j)\in\mathcal{W}} \mathbf{I}[u+i, v+j]\mathbf{K}[i,j],\ \forall (u,v) \in \mathbf{I} \tag{B.12}$$

where the matrix $\mathbf{K} \in \mathbb{R}^{w\times w}$ is called the *mask* or *kernel* and for every output pixel the corresponding window of pixels from the input image $\mathcal{W}$ is multiplied element-wise with the kernel $\mathbf{K}$. This can be considered as the weighted sum of the pixels inside the region $\mathcal{W}$ where the weights are defined by the kernel $\mathbf{K}$.

The most commonly used filters are the *mean* or *blurring filter*, the *median filter*, the *gaussian filter* and the *bilateral filter*. In mean filtering, each output pixel corresponds to the mean value of the neighborhood of the corresponding input pixel, while in median filtering the median of the neighboring pixels is calculated. In Gaussian filtering, the 2D Gaussian kernel is convolved with the image. The Gaussian kernel can be computed by the expression:

$$\mathbf{G}(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}} \tag{B.13}$$

where σ is the standard deviation parameter of the gaussian distribution. The centre of the kernel is considered to have coordinates $(0,0)$ while $i, j \in [-h, h]$. Finally, the bilateral filter is a filter that does not only smooth out the noise, but it also preserves the edges in the image.

B.4.3 Thresholding

Thresholding is a segmentation procedure where each pixel in the image is classified either as a *foreground pixel* or as a *background pixel*. Foreground pixels belong to objects in the image that are of interest and need to be examined, while background pixels refer to the rest of the image. The classification of pixels into foreground and background pixels is done based on their intensity values. The intensities of all pixels are compared to a reference value of intensity, also called the *threshold value*. Once the foreground pixels are separated from the rest, a specific intensity value is assigned to all of them in order to identify them correctly. In this work, for example, a thresholding operation was used to distinguish laser dots from the rest of the image. The pixels that passed the intensity threshold value were classified as laser dot pixels and the value 255 (white) was assigned to all of them, while a value of 0 (black) was assigned to the rest of the image.

There are two kinds of thresholding procedures, namely the *global thresholding* and the *local thresholding*. When a unique threshold value is used for the entire image, then the thresholding operation is termed as *global*. For example, a well-known thresholding approach is the *Otsu's method* ([29]), where the global threshold value is chosen to be the intensity value that can separate the corresponding histogram into two distinct regions in the most effective way. The method provides optimum results when the image has a bimodal histogram.

In the case of the *local* or *adaptive* thresholding, a separte threhold value is chosen for different regions of the image or even for each pixel separately. A well-known local thresholding method that is used for document binarization is the *Niblack thresholding* ([27]). In that method, a separate threshold value is computed for each pixel in the image based on the mean value and standard deviation of its neighborhood.

B.4.4 Labeling

Labelling is a procedure where a number of pixels is grouped together and is treated as one entity. These pixels share common characteristics usually related to their intensity values, their colour or their proximity to each other. After the labelling procedure, a number of additional attributes can be assigned to this entity. These attributes may refer to size, shape, mean intensity or mean colour. For example, in this work, after the classification of pixels into laser dot pixels and background, a labelling procedure was followed in order to group the pixels into distinct entities based on their proximity. These entities were then treated as the detected laser dots produced by the laser configuration.

Appendix C

Code Documentation

C.1 Matlab Code

► Leader_FollowerControl.m (script)

This is the main script that executes the simulation procedure and plots the simulation results. The structure of the code is given in the algorithm segment 1.

Algorithm 1 Main simulation script

```
Initialization step
Define meander waypoints $WP$
for all $i$ in $WP$ do
    while error check do
        Call state2laserCentresFULL
        Calculate Leader laser dot centre
        Call leaderController
        Call laserCentres2visionState
        Call poseVelocity
        Call simpleFilter
        Call followerController
        Propagate the state of the Leader
        Propagate the state of the Follower
        Increase iteration index
        Refresh previous values
    end while
end for
Plot results
```

► leaderController.m (function)

This function produces the control inputs for the Leader vehicle. It calls the `switching_scheme` function in order to navigate the vehicle in a saw-like manner. When the vehicle must move along its z-axis, the function produces the necessary control inputs to stabilize the vehicle at its last distance x and orientation ψ to the surface and to minimize the error along the vertical direction.

Input variables:

- `state`: the current state of the vehicle

- `init_state`: the position of the vehicle near the lastly visited waypoint
- `dstate`: the desired state of the vehicle, i.e. the position of the next waypoint

Output variables:

- `u`: the surge control input
- `r`: the yaw control input
- `w`: the heave control input

► followerController.m (function)

This function implements the control scheme for the Follower vehicle. It also calculates the vector $\dot{\mathbf{n}}_{\mathbf{L}} = [\dot{x}_L, \dot{y}_L, \dot{z}_L, \dot{\psi}_L]^T$ which corresponds to the linear and yaw velocities of the Leader vehicle with respect to the WCS.

Input variables:

- `GnL`: the state of the Leader vehicle with respect to (w.r.t.) the GCS
- `GnF`: the state of the Follower vehicle w.r.t. the GCS
- `GvL`: the linear and yaw velocities of the Leader vehicle w.r.t. the GCS
- `GvF`: the linear and yaw velocities of the Follower vehicle w.r.t. the GCS
- `u_c`: the abscissa (u-coordinate) of the Leader laser dot centre inside the Follower's image frame
- `v_c`: the ordinate (v-coordinate) of the Leader laser dot centre inside the Follower's image frame
- `y_c`: the y-coordinate of the Leader laser dot centre w.r.t. to the FcCS
- `wF_prev`: the previous value of the Follower's heave velocity

Output variables:

- `u`: the surge control input
- `v`: the sway control input
- `w`: the heave control input
- `r`: the yaw control input
- `xL_dot`: the Leader velocity along the x-axis w.r.t. the WCS ($\dot{x}_L$)
- `yL_dot`: the Leader velocity along the y-axis w.r.t. the WCS ($\dot{y}_L$)
- `zL_dot`: the Leader velocity along the z-axis w.r.t. the WCS ($\dot{z}_L$)
- `yawL_dot`: the Leader velocity about the z-axis (yaw) w.r.t. the WCS ($\dot{\psi}_L$)

► switching_scheme.m (function)

This function implements the saw-like motion of the Leader vehicle. It is called by the `leaderController` function in order to provide the desired values x_d and ψ_d depending on the mode of the controller and the desired direction of motion.

Input variables:

- `state`: the current state of the Leader vehicle
- `y0`: the position of the Leader vehicle along the y-axis of the WCS at the previous waypoint
- `yd`: the desired position of the Leader vehicle along the y-axis of the WCS
- `xbounds`: the x-axis positions x_{min}, x_{med}, x_{max} that the Leader vehicle must reach during the control operation

- psibounds: the angles ψ_c, $-\psi_c$ that the Leader vehicle must reach during the control operation

Output variables:

- xd: the desired x-axis position of the Leader vehicle
- psid: the desired yaw orientation of the Leader vehicle

► state2laserCentresFULL.m (function)

This function takes as input the states of the vehicles with respect to the WCS and calculates the image space coordinates of all laser dots inside the image frame of the Follower. In order to do that, it calls the functions laser2line3D and rotationMatrix.

Input variables:

- nL: the pose of the Leader vehicle w.r.t. the WCS
- nF: the pose of the Follower vehicle w.r.t. the WCS

Output variables:

- uvTL: the image coordinates of the Top Left Leader laser dot
- uvTR: the image coordinates of the Top Right Leader laser dot
- uvBL: the image coordinates of the Bottom Left Leader laser dot
- uvBR: the image coordinates of the Bottom Right Leader laser dot
- uvL: the image coordinates of the Left Follower laser dot
- uvR: the image coordinates of the Right Follower laser dot

► laserCentres2visionState.m (function)

The function calculates the poses of the vehicles with respect to the GCS based on the image space coordinates of the laser dots from the vehicles inside the Follower's image frame. Its C/C++ equivalent is called stateEstimation.

Input variables:

- uvTL: the image coordinates of the Top Left Leader laser dot
- uvTR: the image coordinates of the Top Right Leader laser dot
- uvBL: the image coordinates of the Bottom Left Leader laser dot
- uvBR: the image coordinates of the Bottom Right Leader laser dot
- uvL: the image coordinates of the Left Follower laser dot
- uvR: the image coordinates of the Right Follower laser dot

Output variables:

- nVisL: the pose of the Leader vehicle w.r.t. the GCS
- nVisF: the pose of the Follower vehicle w.r.t. the GCS
- y_c: the y-coordinate of the Leader laser dot centre w.r.t. to the FcCS

► laser2line3D.m (function)

This function expresses the beam of a laser pointer as a line in 3D space. The representation of a line in 3D space can be given as:

$$\mathbf{r} = \mathbf{a}t + \mathbf{d} \tag{C.1}$$

where $\mathbf{r}$ represents a vector that begins at the origin and ends at any point on the line, $\mathbf{a}$ is the direction vector of the line, $\mathbf{d}$ is a vector that corresponds to a known point on the line and t is a parameter (see also Chap. 8 Section 8.1). When the function is used for calibration purposes, the deviations of the laser beams are also taken into consideration by a vector w_{err} which represents roll, pitch and yaw deviations. Otherwise, this vector is considered as a 1×3 vector with zeros.

Input variables:

- `n`: the current state of the vehicle
- `w_conf`: the angle of the laser pointer equipped on the vehicle w.r.t. its coordinate frame
- `w_err`: the deviation angle of the beam emitted from the laser pointer w.r.t. to a coordinate frame that lies on the pointer itself
- `d0`: the position vector of the laser pointer tip where the beam is emitted from w.r.t. the coordinate system of the vehicle

Output variables:

- `a`: the direction vector of the resulting 3D line expression
- `d`: the constant vector of the resulting 3D line expression

► rotationMatrix.m (function)

This function calculates the rotation matrix that corresponds to the orientation of a vehicle in terms of its Euler angles.

Input variables:

- `n`: the current state of the vehicle

Output variables:

- `Rzyx`: the rotation matrix that corresponds to the orientation of the vehicle

► poseVelocity.m (function)

This function calculates the rates of change for the poses of the vehicles with respect to the GCS based on their current and previous states.

Input variables:

- `GnL`: the state of the Leader vehicle w.r.t. the GCS
- `GnF`: the state of the Follower vehicle w.r.t. the GCS
- `GnL_prev`: the previous state of the Leader vehicle w.r.t. the GCS
- `GnF_prev`: the previous state of the Follower vehicle w.r.t. the GCS
- `dt`: the elapsed time between two successive states

Output variables:

- `GvL`: the linear and yaw velocities of the Leader vehicle w.r.t. the GCS
- `GvF`: the linear and yaw velocities of the Follower vehicle w.r.t. the GCS

► simpleFilter.m (function)

This function filters out any noise produced by the differentiation procedure of the `poseVelocity` function using a simple averaging filter.

Input variables:

- `GvL`: the linear and yaw velocities of the Leader vehicle w.r.t. the GCS
- `GvF`: the linear and yaw velocities of the Follower vehicle w.r.t. the GCS
- `GvL_prev`: the previous linear and yaw velocities of the Leader vehicle
- `GvF_prev`: the previous linear and yaw velocities of the Follower vehicle

Output variables:

- `GvL_filt`: the filtered linear and yaw velocities of the Leader vehicle w.r.t. the GCS
- `GvF_filt`: the filtered linear and yaw velocities of the Follower vehicle w.r.t. the GCS

► leaderFollowerCalibration.m (script)

This script performs the calibration procedure for the camera system of the Follower vehicle. Its structure is presented in the algorithm segment 2.

Algorithm 2 Script for the calibration procedure

```
Initialization step
Read Isotrak data
Read computer vision data
Call state2laserCentres3D_FULL
Define the least-squares problem % Define the matrices that form the problem
Solve the least-squares problem using lsqlin
Recalculate image coordinates of laser dots based on estimated parameters
Recalculate Follower mean range and yaw orientation
Plot results
```

► state2laserCentres3D_FULL.m (function)

This function calculates the 3D coordinates of all laser dots with respect to the camera coordinate system of the Follower. It is used during the calibration procedure in order to transform the Isotrak measurement into meaningfull data for the calibration. Although the function calculates the 3D coordinates of all laser dots, only those that correspond to the Follower are used during the calibration procedure.

Input variables:

- `nL`: the pose of the Leader vehicle w.r.t. the WCS
- `nF`: the pose of the Follower vehicle w.r.t. the WCS

Output variables:

- `ccTLwrtF`: the 3D-space coordinates of the Top Left Leader laser dot w.r.t. the FcCS

- ccTRwrtF: the 3D-space coordinates of the Top Right Leader laser dot w.r.t. the FcCS
- ccBLwrtF: the 3D-space coordinates of the Bottom Left Leader laser dot w.r.t. the FcCS
- ccBRwrtF: the 3D-space coordinates of the Bottom Right Leader laser dot w.r.t. the FcCS
- ccLwrtF: the 3D-space coordinates of the Left Follower laser dot w.r.t. the FcCS
- ccRwrtF: the 3D-space coordinates of the Right Follower laser dot w.r.t. the FcCS

C.2 C/C++ Code

C.2.1 Vision Module

► void laserCentresDetect (IplImage* frame, float* rawData)

This function performs the detection of the laser dots inside the image frame of the Follower as described in Chapter 6. It stores the results in a 12×1 vector where the first 8 elements correspond to the image coordinate pairs of the 4 Leader laser dots and the last 4 elements to the 2 coordinate pairs of the Follower laser dots.

Input variables:

- frame: the image frame captured by the camera in IplImage format
- rawData: a 12×1 vector that contains the image space coordinates of the detected laser dots in unsorted order

Output variables: Void.

► void sortLasers (float* rawData, float* sortedData)

This function sorts the image coordinate pairs in the vector rawData and stores the result in the vector sortedData. The 6 coordinate pairs are sorted so that the first 4 pairs correspond to the Top Left, Bottom Left, Top Right and Bottom Right Leader laser dots, while the last 2 correspond to the Left and Right Follower laser dots, respectively.

Input variables:

- rawData: the 12×1 vector that contains the image space coordinates of the detected laser dots in unsorted order
- sortedData: a 12×1 vector that contains the image space coordinates of the detected laser dots in sorted order

Output variables: Void.

► void stateEstimation (float* filterData, float* nVisL, float* LmLead, float* nVisF, float* LmFollow, float* yc)

This function implements the relative localization procedure between the two vehicles performed by the Follower. The variable filterData is used as input to the function and contains the filtered image space coordinates of all laser dots inside Follower's image frame. The rest of the variables are inserted empty and are filled by the function itself.

Input variables:

- filterData: the 12×1 vector that contains the filtered image space coordinates of the sorted laser dots

- nVisL: the pose vector (4×1) of the Leader vehicle w.r.t. the GCS
- LmLead: the mean range of the Leader vehicle
- nVisF: the pose vector (4×1) of the Follower vehicle w.r.t. the GCS
- LmFollow: the mean range of the Follower vehicle
- yc: the y-coordinate of the Leader laser dots centre w.r.t. the FcCS

Output variables: Void.

► void velocityVision (float* GnL, float* GnF, float* GnL_prev, float* GnF_prev, double dt, float* GvL, float* GvF)

This function calculates the rate of change for the states of the vehicles with respect to the GCS based on their current and previous states and the elapsed time.

Input variables:

- GnL: the pose vector (4×1) of the Leader vehicle w.r.t. the GCS
- GnF: the pose vector (4×1) of the Follower vehicle w.r.t. the GCS
- GnL_prev: the previous pose vector of the Leader vehicle w.r.t. the GCS
- GnF_prev: the previous pose vector of the Follower vehicle w.r.t. the GCS
- dt: the elapsed time between two successive measurements
- GvL: the velocity vector (4×1) of the Leader vehicle w.r.t. the GCS
- GvF: the velocity vector (4×1) of the Follower vehicle w.r.t. the GCS

Output variables: Void.

► void state2lasersCentresFULL (float* nLeader, float* nFollower, float* rawData)

This function calculates the image space coordinates of the laser dots inside the Follower image frame based on the states of the vehicles as provided by the Isotrak sensor. The function is used only as a vision substitute during the testing procedure.

Input variables:

- nLeader: the pose vector (4×1) of the Leader as given by the Isotrak system
- nFollower: the pose vector (4×1) of the Follower as given by the Isotrak system
- rawData: the 12×1 vector that contains the image space coordinates of the detected laser dots in unsorted order

Output variables: Void.

► void showLaserDots (IplImage* image, vector<Point2f> laserDots, Scalar color, int cross)

This function shows the detected laser dots on the image provided as input by drawing two concentric circles at their centres. It is used for each group of laser dots (red and green) separately. The user can use the variable cross to draw two lines that pass through the center of each laser dot at right angle to each other and traverse the image along its width and height. This last feature can be used during the calibration procedure.

Input variables:

- image: the image on which the laser dots are drawn

- `laserDots`: structure that holds the image coordinates of the laser dots
- `color`: the color of the laser dots
- `cross`: flag that determines if a cross will be printed on the laser dot or not

Output variables: Void

► int selectLaser (vector<RotatedRect>& lasers, int numlasers)

This function implements the selction procedure of the computer vision algorithm (see Chapter 6).

Input variables:

- `lasers`: structure with information about the laser dots
- `numlasers`: the desired number of laser dots

Output variables: 0

C.2.2 Filtering Module

► void kalman (float *x, float *P, float z, float Q, float R)

This function implements a Kalman filter. It is used on the image space coordinates of the detected laser dots in order to filter any noise produced by the detection procedure.

Input variables:

- `x`: the previously filtered measurement
- `P`: the variance $\mathbf{P}$
- `z`: the current raw measurement
- `Q`: the measurement noise
- `R`: the model noise

Output variables: Void.

► void simpleAverage (float* GvL, float* GvF, float* GvL_prev, float* GvF_prev)

This function implements a simple averaging filter. It is used on the calculated vehicle velocity measurements to filter the noise produced by the differentiation procedure.

Input variables:

- `GvL`: the velocity vector (4×1) of the Leader vehicle w.r.t. the GCS
- `GvF`: the velocity vector (4×1) of the Follower vehicle w.r.t. the GCS
- `GvL_prev`: the previous velocity vector of the Leader vehicle w.r.t. the GCS
- `GvF_prev`: the previous velocity vector of the Follower vehicle w.r.t. the GCS

Output variables: Void.

C.2.3 Socket Communications Module

► int serverSocketInit (int port)

The function initializes the Server part of the socket and prints the appropriate message.

Input variables:

- `port`: the port of the socket

Output variables:

- `listenfd`: the handler of the port from which the server ``listens" for connections

► int clientSocketInit (const char* ip_addr, int port, sockaddr_in *serv_addr)

The function initializes the Client part of the socket and prints the appropriate message.

Input variables:

- `ip_addr`: the IP address of the server
- `port`: the port of the socket
- `serv_addr`: structure that contains all the information for the socket

Output variables:

- `socketfd`: the handler of the socket to which the client is connected

► int socketAccept (int socketfd)

This function accepts a connection to the server and prints the appropriate message.

Input variables:

- `socketfd`: the handler of the socket

Output variables:

- `connectedfd`: the handler of the socket that is connected to the server

► int socketConnect (int socketfd, sockaddr_in serv_addr)

This function attempts a connection of the Client to a socket opened by a Server.

Input variables:

- `socketfd`: the handler of the socket
- `serv_addr`: structure that contains all the information for the socket

Output variables:

- 0, if the connection with the server is successful
- 1, otherwise

► int socketRead (int socketfd, char* recvBuffer)

This function reads data from the socket.

Input variables:

- `socketfd`: the handler of the socket
- `recvBuffer`: a buffer that will contain the information that is received from the socket

Output variables:

- `bytesRead`: the number of bytes read from the socket, if the information is received successfully
- -1, otherwise

► int socketWrite (int socketfd, char* sendBuffer)

This function writes data to the socket.

Input variables:

- `socketfd`: the handler of the socket
- `sendBuffer`: a buffer that contains the information that is sent to the socket

Output variables:

- `bytesSent`: the number of bytes sent to the socket, if the information is written successfully
- -1, otherwise

► void socketEnd (int socketfd)

This function terminates a socket connection.

Input variables:

- `socketfd`: the handler of the socket

Output variables: Void.

C.2.4 Motion Control Module

► LeaderCOOP.cpp

This file implements the main control algorithm for the Leader vehicle. Its structure is presented roughly in the algorithm segment 3.

► FollowerCOOP.cpp

This file implements the main control algorithm for the Follower vehicle. Its structure is presented roughly in the algorithm segment 4.

► void leaderController (float* n, float* nd, float* n0, float* U)

This function implements the motion control scheme of the Leader. The switching scheme is not implemented as a separate module but it is integrated in the Leader control scheme.

Input variables:

Algorithm 3 Main algorithm for the motion control of the Leader

```
Initialization & Memory allocation step
Receive Isotrak measurements for initial position
Define desired waypoints $WP$
for all $i$ in $WP$ do
  while error check do
    Receive Isotrak measurements
    Call leaderController
    Call zPIDcontroller
    Send control message to vehicle
  end while
end for
```

Algorithm 4 Main algorithm for the motion control of the Follower

```
Initialization & Memory allocation step
Read image frame from USB camera
Receive Isotrak measurements
while Experimenting do
  if Simulation Mode ON then
    Call state2lasersCentresFULL to provide vision input
  else
    Call laserCentresDetect to detect laser dots on the image
  end if
  if All laser dots are detected then
    Call sortLasers to sort the detected laser dot coordinates
    Filter the laser dot coordinates using the kalman function
    Call stateEstimation function
    Call velocityVision
    Call simpleAverage
    Calculate the Follower heave velocity w.r.t. WCS ($\dot{z}_F$)
    Calculate the image coordinates of the Leader laser dot centre
    Renew previous values
  else
    Use previous values
  end if
  Call followerController
  Send control message to vehicle
  Show live feed from camera with detected lasers
end while
```

- n: the current state of the Leader vehicle
- nd: the desired position of the Leader vehicle based on the next waypoint to be reached
- n0: the position of the Leader vehicle near the previous waypoint at the beginning of its motion towards the next one
- U: a 3×1 vector that contains the Leader control inputs

Output variables: Void.

► void zPIDcontroller (float z, float zd, float ezprev, float dt, float Sezdt, float* Uz)

This function implements a PID controller dedicated for the motion of the Leader along the z-axis.

Input variables:

- z: the position of the Leader vehicle along the z-axis
- zd: the desired position of the Leader vehicle along the z-axis
- ezprev: the previous error between its current and desired z states
- dt: the elapsed time between two successive measurements
- Sezdt: the sum of the errors that are multiplied by the corresponding time intervals
- Uz: the heave control input

Output variables: Void.

► void followerController (stateVision* stateVisionFollower, float zF_dot, float* U)

This function implements the motion control scheme of the Follower. The scheme uses as input the visual information from the camera, the localization information provided by the state estimation procedure with respect to the GCS and the calculated velocities of the vehicles.

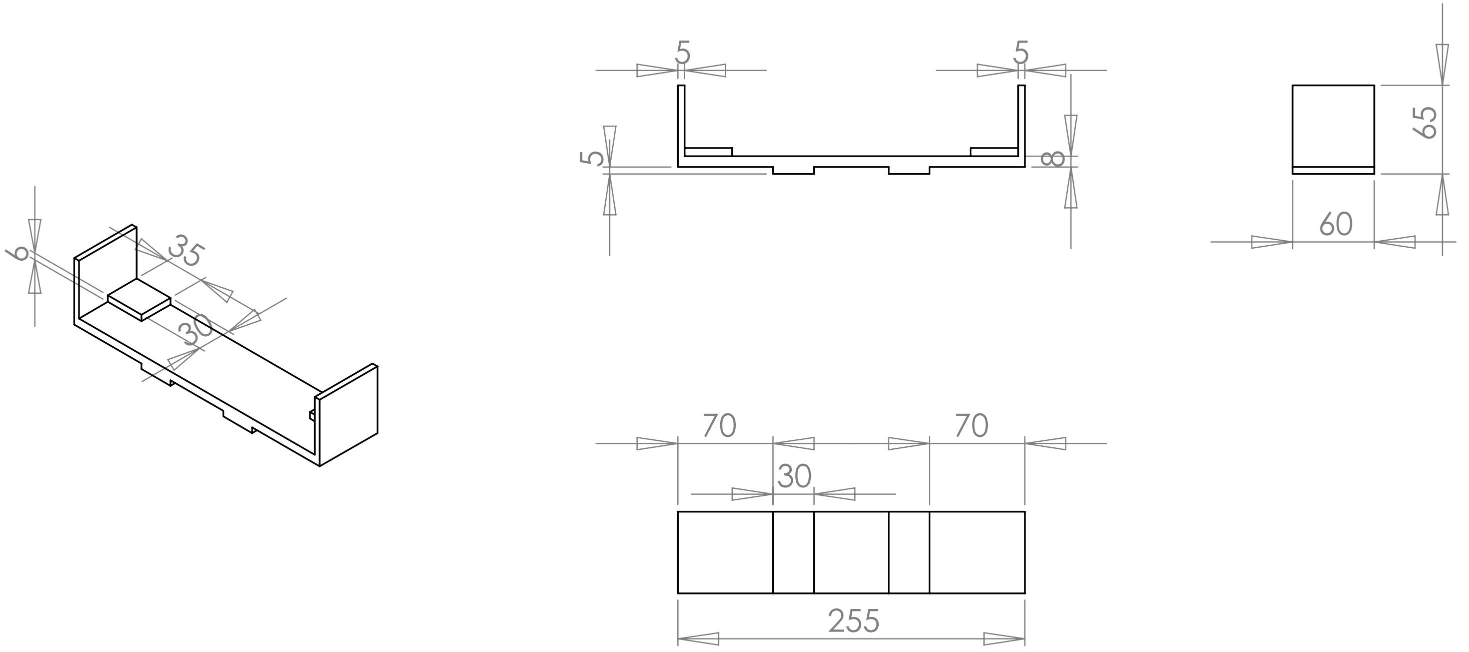
Input variables:

- stateVisionFollower: a structure that contains all the necessary information for the control of the Follower
- zF_dot: the velocity of the Follower w.r.t. the z-axis of the WCS as given by the Isotrak system
- U: a 4×1 vector that contains the Follower control inputs

Output variables: Void.

Appendix D

Laser Setup Schematics



PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

NEXT ASSY	USED ON
APPLICATION	

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

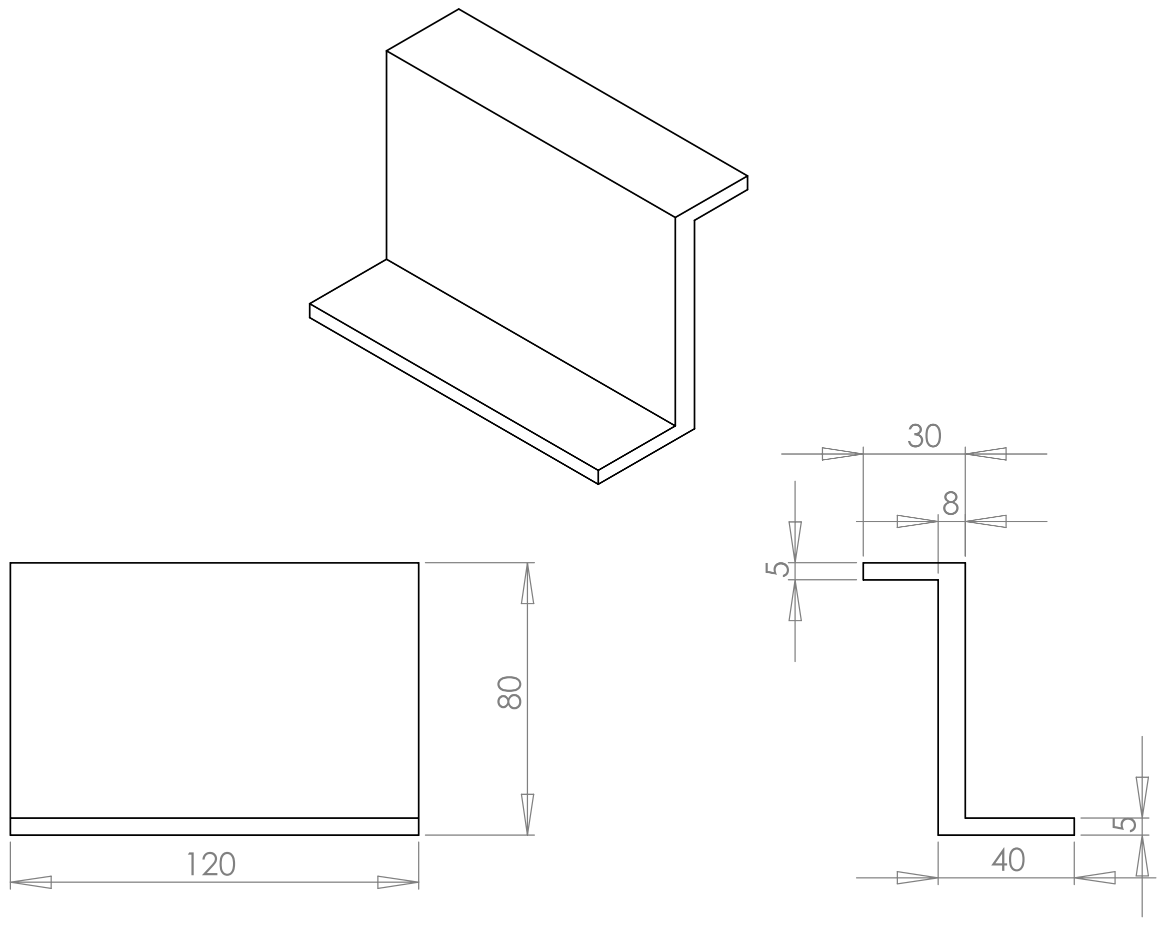
	NAME	DATE
DRAWN		
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

TITLE:

SIZE	DWG. NO.	REV
A	leader_part	
SCALE: 1:5	WEIGHT:	SHEET 1 OF 1

5 4 3 2 1



PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

NEXT ASSY	USED ON
APPLICATION	

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN		
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

TITLE:

SIZE	DWG. NO.	REV
A	follower_part	

SCALE: 1:2 WEIGHT: SHEET 1 OF 1

5 4 3 2 1

Bibliography

[1] Camera calibration with opencv - opencv 2.4.5.0 documentation, April 2013.

[2] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino. Steering of unicycle-like vehicles via lyapunov techniques. In *IEEE Robotics and Automation Magazine*, pages 27--35, 1995.

[3] F. Arrichiello, S. Chiaverini, and T.I. Fossen. Formation control of marine surface vessels using the null-space-based behavioral control. *Lecture Notes in Control and Information Science, Group Coordination and Cooperative Control*, 336:1--19, 2006.

[4] P. Baccou, B. Jouvencel, V. Creuze, and C. Rabaud. Cooperative positioning and navigation for multiple auv operations. In *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, volume 3, pages 1816--1821, November 2001.

[5] R. Bachmayer and N. E. Leonard. Vehicle networks for gradient descent in a sampled environment. In *Proc. 41st IEEE Conference Decision and Control (CDC)*, pages 2968--2973, 2002.

[6] A. Bahr, J. Leonard, and M. F. Fallon. Cooperative localization for autonomous underwater vehicles. *The International Journal of Robotics Research*, 28(6):714--728, June 2009.

[7] Jean-Yves Bouguet. Camera calibration toolbox for matlab, July 2010.

[8] A. Caiti, G. Casalino, A. Munafo, and A. Turetta. Cooperating auv teams: Adaptive area coverage with space-varying communication constraints. In *Proc. of MTS/IEEE Conference and Exhibition OCEANS.*, pages 1--7, May 2009.

[9] P. Corke. *Robotics, Vision and Control. Fundamental Algorithms in MATLAB.* Springer, Berlin, 2011.

[10] R.X. Cui, S. S. Ge, B. Voon Ee How, and Y. S. Choo. Leader-follower formation control of underactuated auvs with leader position measurement. In *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan.

[11] J. V. Diosdado and I. T. Ruiz. Decentralized simultaneous localization and mapping for auvs. In *OCEANS, 2007. MTS/IEEE Conference and Exhibition*, pages 1--6, June 2007.

[12] S. Emrani, A. Dirafzoon, H. A. Talebi, S. K. Yadavar Nikravesh, and M. B. Menhaj. An adaptive leader-follower formation controller for multiple auvs in spatial motions. In *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, pages 59--64, November 2010.

[13] B. Englot and F. Hover. Inspection planning for sensor coverage of 3d marine structures. In *Proc. of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2010*, pages 4412--4417, October 2010.

[14] R. Fierro, P. Song, A. Das, and V. Kumar. *Cooperative Control and Optimization, Applied Optimization*, volume 66. 2002.

[15] T. I. Fossen. *Guidance and control of ocean vehicles.* Wiley, New York, 1994.

[16] R. Ghabcheloo, A. P. Aguiar, A. Pascoal, and C. Silvestre. Coordinated path-following control of multiple auvs in the presence of communication failures and time delays. In *Proc. of the 45th IEEE Conference on Decision and Control.*, pages 4345--4350, December 2006.

[17] R. C. Gonzalez, R. E. Woods, and S. L. Eddins. *Digital Image Processing Using Matlab*. Dorling Kindersley, 2004.

[18] M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball. The WHOI micro-modem: an acoustic communications and navigation system for multiple platforms. In *Proc. of MTS/IEEE Conference and Exhibition OCEANS.*, volume 2, pages 1086--1092, September 2005.

[19] A. Howard, M. J. Matari, and G. S. Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. *Autonomous Robots*, 13(2).

[20] S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. In *IEEE Trans. on Robotics and Automation*, pages 651--670, 1996.

[21] D. Jiang, Y. Pang, and Z. Qin. Application of moos-ivp architecture in multiple autonomous underwater vehicle cooperation. In *Proc. of Chinese Control and Decision Conference (CCDC).*, May 2010.

[22] G. C. Karras, D. J. Panagou, and K. J. Kyriakopoulos. Target-referenced localization of an underwater vehicle using a laser-based vision system. In *OCEANS, 2006. MTS/IEEE Conference and Exhibition*, 2006.

[23] N. E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proc. of the 40th IEEE Conference on Decision and Control (CDC)*, pages 2968--2973, 2001.

[24] M. A. Lewis and K.-H. Tan. High precision formation control of mobile robots using virtual structures. *Autonomous Robots*, 4(1).

[25] M. Mataric. Designing emergent behaviors: From local interactions to collective intelligence. In *From Animals to Animats 2: Int. Conf. on Simulation of Adaptive Behavior*, pages 423--441, 1993.

[26] S. Negahdaripour and P. Firoozfam. An rov stereovision system for ship-hull inspection. *IEEE Journal of Oceanic Engineering*, 31(3):551--564, 2006.

[27] W. Niblack. *An Introduction to Digital Image Processing.* Prentice Hall, 1986.

[28] P. Ogren and N. E. Leonard. Obstacle avoidance in formation. In *Proc. of IEEE 2003 International Conference on Robotics and Automation (ICRA)*, pages 2968--2973, 2002.

[29] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man & Cybernetics*, 9:62--66, January 1979.

[30] E. Pagello, A. D'Angelo, F. Montesello, F. Garelli, and C. Ferrari. Cooperative behaviors in multi-robot systems through implicit communication. *Robotics and Autonomous Systems*, 29(1).

[31] D. Panagou and K. J. Kyriakopoulos. Control of underactuated systems with viability constraints. In *Proc. 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, page 5497–5502, December 2011.

[32] L. E. Parker, B. Kannan, X. Fu, and Y. Tang. Heterogeneous mobile sensor net deployment using robot herding and line-of-sight formations. In *Proc. of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2003*, volume 3, pages 2488--2493, October 2003.

[33] G. A. S. Pereira, B. S. Pimentel, L. Chaimowicz, and Campos M. F. M. Coordination of multiple mobile robots in an object carrying task using implicit communication. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 281--286, 2002.

[34] W. Ren and R. W. Beard. Decentralized scheme for spacecraft formation flying via the virtual structure approach. *Journal of Guidance, Control and Dynamics*, 27:387--403, January 1997.

[35] D. G. Roberson and D. J. Stilwell. Control of an autonomous underwater vehicle platoon with a switched communication network. In *Proc. of the 2005 American Control Conference, ACC 2005.*, volume 6, pages 4333--4338, June 2005.

[36] S. Sariel and T. Balch. A distributed multi-robot cooperation framework for real time task achievement. In *Proc. of DARS 7*, pages 187--196, 2006.

[37] S. Singh, M. Grund, B. Bingham, R. Eustice, H. Singh, and L. Freitag. Underwater acoustic navigation with the WHOI micro-modem. In *Proc. of MTS/IEEE Conference and Exhibition OCEANS.*, September 2006.

[38] J. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, 1991.

[39] The Society of Naval Architects and Marine Engineers. Nomenclature for treating the motion of a submerged body through a fluid (technical and research bulletin no 1–5), 1950.

[40] J. Wang, X. Wu, and Z. Xu. Decentralized formation control and obstacle avoidance based on potential field method. In *Proc. of the 5th Conference on Machine Learning and Cybernetics*, pages 803--808, August 2006.

[41] X. Xiang, B. Jouvencel, and O. Parodi. Coordinated formation control of multiple autonomous underwater vehicles for pipeline inspection. *International Journal of Advanced Robotic Systems*, 10(1):75--84, 2010.

[42] S.-C. Yu and T. Ura. A system of multi-auv interlinked with a smart cable for autonomous inspection of underwater structures. *International Journal of Offshore and Polar Engineering*, 14(4):265--273, December 2004.